Steel-Belted RADIUS Administration Guide Global Enterprise Edition

Release 6.27

Release	**6.27**
Published Date	**October 2019**

Copyright (c) 1995-1998 by Cisco systems, Inc.

Permission to use, copy, modify, and distribute this software for any purpose and without fee is hereby granted, provided that this copyright and permission notice appear on all copies of the software and supporting documentation, the name of Cisco Systems, Inc. not be used in advertising or publicity pertaining to distribution of the program without specific prior permission, and notice be given in supporting documentation that modification, copying and distribution is by permission of Cisco Systems, Inc.

Cisco Systems, Inc. makes no representations about the suitability of this software for any purpose. THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The code written by Marc Huber is distributed under the following license:

Copyright (C) 1999-2015 Marc Huber (<Marc.Huber@web.de>). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions ared met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

  This product includes software developed by Marc Huber (<Marc.Huber@web.de>).

  Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ITS AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

apache/httpclient, that is used in SBR-E software is of license type "Apache License 2.0" and refer the following URL for more information:
https://github.com/apache/httpcomponents-client/blob/4.5.x/LICENSE.txt.

bcgit/bc-java, that is used in SBR-E software is of license type "MIT" and refer the following URL for more information:
https://github.com/bcgit/bc-java/blob/r1rv60/LICENSE.html.

google/gwt, that is used in SBR-E software is of license type "Apache License 2.0" and refer the following URL for more information:
http://www.gwtproject.org/terms.html.

gwtbootstrap3/gwtbootstrap3, that is used in SBR-E software is of license type "Apache License 2.0" and refer the following URL for more information: https://github.com/gwtbootstrap3/gwtbootstrap3/blob/0.9.3/LICENSE.

kohsuke/WinSW, that is used in SBR-E software is of license type "MIT" and refer the following URL for more information:
https://github.com/kohsuke/winsw/blob/winsw-v2.2.0/LICENSE.txt.

laaglu/lib-gwt-file, that is used in SBR-E software is of license type "GNU Lesser General Public License v3" and refer the following URL for more information:
http://www.gnu.org/licenses/lgpl.html .

Revision History

The following table lists the revision history for this document.

Revision	Date	Description
6.27	October 2019	6.27-R1 Updates
6.26	May 2019	6.26-R1 Updates

Contents

About This Guide

The Steel-Belted Radius Administration Guide/Global Enterprise Edition describes how to configure and administer the Steel-Belted Radius software on a server running the Linux operating system or the Windows operating system.

Before You Begin

This manual assumes that you have installed the Steel-Belted Radius software and the SBR Administrator/WebGUI. For more information, refer to the Steel-Belted Radius Installation and Upgrade Guide.

Audience

This manual is intended for network administrators responsible for implementing and maintaining authentication, authorization, and accounting services for an enterprise. This manual assumes that you are familiar with general RADIUS and networking concepts and the specific environment in which you are installing Steel-Belted Radius.

If you use Steel-Belted Radius with third-party products such as Oracle or RSA SecurID, you should be familiar with their installation, configuration, and use.

What's in This Manual

This manual contains the following chapters and appendixes:

- Chapter 1, "About Steel-Belted Radius" presents an overview of Steel-Belted Radius and describes licensing requirements for Steel-Belted Radius.
- Chapter 2, "RADIUS Basics" summarizes important concepts relating to the operation of Steel-Belted Radius.
- Chapter 3, "Using SBR Administrator" describes how to use the SBR Administrator to configure Steel-Belted Radius.
- Chapter 4, "Using Web Graphic User Interface (GUI)" describes how to use the WebGUI to configure Steel Belted Radius Server.
- Chapter 5, "Administering RADIUS Clients via SBR Legacy Administrator" describes how to set up RADIUS clients and client groups via SBR legacy administrator.
- Chapter 6, "Administering RADIUS Clients via WebGUI" describes how to set up RADIUS clients and client groups via WebGUI.
- Chapter 7, "Administering RADIUS Location Groups via Legacy SBR Administrator" describes how to set up a location group via legacy SBR administrator.
- Chapter 8, "Administering RADIUS Location Groups via WebGUI" describes how to set up a location group via WebGUI.
- Chapter 9, "Administering Users via Legacy SBR Administrator" describes how to set up users in the Steel-Belted Radius database via legacy SBR.
- Chapter 10, "Administering Users via WebGUI" describes how to set up users in the Steel-Belted Radius database via WebGUI.

- Chapter 11, "Administering Profiles via Legacy SBR Administrator" describes how to set up user profiles to simplify user administration via legacy SBR.
- Chapter 12, "Administering Profiles via WebGUI" describes how to set up user profiles to simplify user administration via WebGUI.
- Chapter 13, "Administering Proxy RADIUS via Legacy SBR Administrator" describes how to identify proxy RADIUS targets via legacy SBR.
- Chapter 14, "Administering Proxy RADIUS via WebGUI" describes how to identify proxy RADIUS targets via WebGUI.
- Chapter 15, "Administering RADIUS Tunnels via Legacy SBR Administrator" describes how to set up secure RADIUS tunnels via legacy SBR.
- Chapter 16, "Administering RADIUS Tunnels via WebGUI" describes how to set up secure RADIUS tunnels via WebGUI.
- Chapter 17, "Administering Address Pools via Legacy SBR Administrator" describes how to set up IPv4 and IPX address pools via legacy SBR.
- Chapter 18 "Administering Address Pools via WebGUI" describes how to set up IPv4 and IPX address pools via WebGUI.
- Chapter 19, "Setting Up Administrator Accounts via Legacy SBR Administrator" describes how to identify who can administer Steel- Belted Radius via legacy SBR.
- Chapter 20, "Setting Up Administrator Accounts via WebGUI" describes how to identify who can administer Steel- Belted Radius.
- Chapter 21, "Configuring Realm Support" describes how to configure and maintain directed and proxy RADIUS realms.
- Chapter 22, "Setting Up Filters via Legacy SBR Administrator" describes how to configure and maintain attribute filters in Steel- Belted Radius.
- Chapter 23 "Setting Up Filters via WebGUI" describes how to configure and maintain attribute filters in Steel- Belted Radius via WebGUI.
- Chapter 24, "Setting Up EAP Authentication Policies via Legacy SBR Administrator" presents an overview of Extensible Authentication Protocol (EAP) types and describes how to configure and sequence RADIUS authentication methods.
- Chapter 25 "Setting Up EAP Authentication Policies via Web GUI" presents an overview of Extensible Authentication Protocol (EAP) types and describes how to configure and sequence RADIUS authentication methods via WebGUI.
- Chapter 26, "Configuring TACACS+ Server" presents an overview of TACACS+ server and describes how to configure into Steel-Belted Radius.
- Chapter 27, "Configuring SNMP," presents an overview of SNMP components and describes the

statistics available for Steel-Belted Radius servers and clients.

- Chapter 28, "Configuring Replication via Legacy SBR Administrator" describes how to configure and use the centralize configuration management (CCM) feature to coordinate Steel-Belted Radius server settings in a replication environment.
- Chapter 29 "Configuring Replication via WebGUI" describes how to configure and use the centralized configuration management (CCM) feature to coordinate Steel-Belted Radius server settings in a replication environment.
- Chapter 30, "LDAP Configuration Interface" describes how to use public domain LDAP utilities to populate a Steel-Belted Radius server database.
- Chapter 31, "Configuring SQL Authentication" describes how to configure authentication against records stored in an external SQL database.
- Chapter 32, "Configuring SQL Accounting" describes how to configure Steel-Belted Radius to write accounting information to an external SQL database.
- Chapter 33, "Configuring LDAP Authentication" describes how to configure authentication against records stored in an external LDAP database.
- Chapter 34, "Displaying Statistics via Legacy SBR Administrator" describes how to use the monitoring facilities in Steel-Belted Radius.
- Chapter 35 "Displaying Statistics via WebGUI" describes how to use the monitoring facilities in Steel-Belted Radius.
- Chapter 36 "Logging and Reporting via Legacy SBR Administrator" describes how to use the logging and reporting facilities in Steel-Belted Radius.
- Chapter 37, "Logging and Reporting via WebGUI" describes how to use the logging and reporting facilities in Steel- Belted Radius.
- Appendix A, "Glossary" provides brief explanations for RADIUS terminology used in this and other Steel-Belted Radius manuals.
- Appendix B, "When to Restart Steel-Belted Radius" provides a summary of critical operational information.
- Appendix C, "Technical Notes" presents tips for configuring Steel-Belted Radius to interoperate with equipment and facilities from other vendors.
- Appendix D, "Authentication Protocols" provides a matrix of authentication methods and their supported authentication protocols.
- Appendix E, "Importing and Exporting Data" describes how to import and export information in a Steel-Belted Radius database to and from an XML file.
- Appendix F, "Stopping and Starting Steel-Belted Radius" describes how to stop and restart the Steel-Belted Radius service (Windows) or RADIUS daemon (Linux).
- Appendix G, "Stopping and Starting Steel-Belted Radius Jetty Server" describes how to stop and

restart the Steel-Belted Radius Jetty Server (Windows) or (Linux).

- Appendix H, "Use Custom SSL Certificate for Launching SBR-E Web UI" describes how use custom SSL certificates for launching SBR-E Web UI.

TypographicalConventions

Table 1 describes the text conventions used throughout this manual.

Table 1: Typographical Conventions

Convention	Description	Examples
Bold typeface	Indicates buttons, field names, dialog names, and other user interface elements.	Use the **Scheduling** and **Appointment** tabs to schedule a **meeting**.
Plain sans serif typeface	Represents: • Code, commands, and keywords • URLs, file names, and directories	Examples: • Code: certAttr.OU = 'Retail Products Group' • URL: Download the JRE application from: http://java.sun.com/j2se/
Italics	Identifies: • Terms defined in text • Variable elements • Book names	Examples: • Defined term: An RDP client is a Windows component that enables a connection between a Windows server and a user's machine. • Variable element: Use settings in the Users > Roles > Select Role > Terminal Services page to create a terminal emulation session. • Book name: See the Steel-Belted Radius Administration Guide.

Editions/Used In

Steel-Belted Radius is available in multiple editions to meet the requirements of different types of customers. This manual uses the following abbreviations to identify editions of Steel-Belted Radius:

- GEE – Global Enterprise Edition
- EE – Enterprise Edition

Syntax

- radiusdir represents the directory into which Steel-Belted Radius has been installed. By default, this is C:\Program Files (x86)\Pulse Secure\Steel-Belted Radius\Service for Windows systems and /opt/PSsbr/Radius on Linux.
- Brackets [ ] enclose optional items in format and syntax descriptions. In the following example, the first Attribute argument is required; you can include an optional second Attribute argument by entering a comma and the second argument (but not the square brackets) on the same line.

<add | replace> = Attribute [,Attribute]

In configuration files, brackets identify section headers:

the [Processing] section of proxy.ini

In screen prompts, brackets indicate the default value. For example, if you press Enter without entering anything at the following prompt, the system uses the indicated default value (/opt).

Enter install path [/opt]:

- Angle brackets < > enclose a list from which you must choose an item in format and syntax descriptions.
- A vertical bar (|) separates items in a list of choices. In the following example, you must specify add or replace (but not both):

  <add | replace> = Attribute [,Attribute]

Related Documentation

The following documents supplement the information in this manual.

Steel-Belted Radius Documentation

Please review the ReleaseNotes.txt file that accompanies your Steel-Belted Radius software. This file contains the latest information about features, changes, known problems, and resolved problems. If the information the ReleaseNotes.txt file differs from the information found in the Steel-Belted Radius manuals, use the information in the ReleaseNotes.txt file.

In addition to this manual, the Steel-Belted Radius documentation includes the following manuals:

- The Steel-Belted Radius Installation and Upgrade Guide describes how to install the Steel-Belted Radius software on a server running the Linux operating system or the Windows operating system.
- The Steel-Belted Radius Reference Guide describes the configuration options for the Steel-Belted Radius software.

Requests for Comments (RFCs)

The Internet Engineering Task Force (IETF) maintains an online repository of Request for Comments (RFCs) online at http://www.ietf.org/rfc.html. **Table 2** lists the RFCs that apply to this guide.

Table 2: RFCs

RFC Number	Title
RFC 1155	Structure and Identification of Management Information for TCP/IP-based Internets. M. Rose, K. McCloghrie, May 1990.
RFC 1213	Management Information Base for Network Management of TCP/IP-based internets: MIB-II. K. McCloghrie, M. Rose, March 1991.
RFC 2271	An Architecture for Describing SNMP Management Frameworks. D. Harrington, R. Presuhn, B. Wijnen, January 1998.
RFC 2284	PPP Extensible Authentication Protocol (EAP). L. Blunk, J. Volbrecht, March 1998.

RFC Number	Title
RFC 2433	Microsoft PPP CHAP Extensions. G. Zorn, S. Cobb, October 1998.
RFC 2548	Microsoft Vendor-specific RADIUS Attributes. G. Zorn. March 1999.
RFC 2607	Proxy Chaining and Policy Implementation in Roaming. B. Aboba, J. Vollbrecht, June 1999.
RFC 2618	RADIUS Authentication Client MIB. B. Aboba, G. Zorn. June 1999.
RFC 2619	RADIUS Authentication Server MIB. G. Zorn, B. Aboba. June 1999.
RFC 2620	RADIUS Accounting Client MIB. B. Aboba, G. Zorn. June 1999.
RFC 2621	RADIUS Accounting Server MIB. G. Zorn, B. Aboba. June 1999.
RFC 2622	PPP EAP TLS Authentication Protocol. B. Aboba, D. Simon, October 1999.
RFC 2809	Implementation of L2TP Compulsory Tunneling via RADIUS. B. Aboba, G. Zorn. April 2000.
RFC 2865	Remote Authentication Dial In User Service (RADIUS). C. Rigney, S. Willens, A. Rubens, W. Simpson. June 2000.
RFC 2866	RADIUS Accounting. C. Rigney. June 2000.
RFC 2867	RADIUS Accounting Modifications for Tunnel Protocol Support. G. Zorn, B. Aboba, D. Mitton. June 2000.
RFC 2868	RADIUS Attributes for Tunnel Protocol Support. G. Zorn, D. Leifer, A. Rubens, J. Shriver, M. Holdrege, I. Goyret. June 2000.
RFC 2869	RADIUS Extensions. C. Rigney, W. Willats, P. Calhoun. June 2000.
RFC 2882	Network Access Servers Requirements: Extended RADIUS Practices. D. Mitton. July 2000.
RFC 3162	RADIUS and IPv6. B. Aboba, G. Zorn, D. Mitton. August 2001.
RFC 3575	IANA Considerations for RADIUS (Remote Authentication Dial In User Service). B. Aboba, July 2003.
RFC 3579	RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP). B. Aboba, P. Calhoun, September 2003.
RFC 3580	IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines. P. Congdon, B. Aboba, A. Smith, G. Zorn, J. Roese, September 2003.

Third-Party Products

For more information about configuring your access servers and firewalls, consult the manufacturer's documentation provided with each device.

Contacting Technical Support

For technical support, contact Pulse Secure at support@pulsesecure.net,

Check our website (http://www.pulsesecure.net) for additional information and technical notes. When you are running Legacy SBR Administrator, you can choose **Web > Steel-Belted Radius User Page** to access a special home page for Steel-Belted Radius users. When you are running Web GUI, you can choose **Help** > **Home Page** > **Steel-Belted Radius Home Page** to access a special home page for Steel-Belted Radius users.

When you call technical support, please have the following at hand:

- Your Steel-Belted Radius product edition and release number (for example, Global Enterprise Edition version 6.26).
- Information about the server configuration and operating system, including any OS patches that have been applied.
- For licensed products under a current maintenance agreement, your license or support contract number.
- Question or description of the problem, with as much detail as possible.
- Any documentation that might help resolve the problem, such as error messages, memory dumps, compiler listings, and error logs.

Chapter 1
About Steel-Belted Radius

Thank you for selecting Steel-Belted Radius®/Global Enterprise Edition.

Steel-Belted Radius is a complete implementation of the RADIUS (Remote Authentication Dial In User Service) protocol. Steel-Belted Radius interfaces with a wide variety of network access equipment and authenticates remote and wireless LAN (WLAN) users against numerous back-end databases, allowing you to consolidate the administration of all your remote and WLAN users.

Steel-Belted Radius/Global Enterprise Edition delivers a total RADIUS solution on the scale required by large corporations with complex global networks. It provides the power and flexibility you need to manage the delivery of enhanced services to your users, and it integrates with all aspects of your operations, from user authentication and service delivery to cost accounting based on divisions and departments.

Steel-Belted Radius Features

- Centralized management of user access control and security simplifies access administration.
- Flexible, powerful proxy RADIUS features let you easily distribute authentication and accounting requests to the appropriate RADIUS server for processing.
- External authentication features let you authenticate against multiple, redundant SQL or LDAP databases according to configurable load balancing and retry strategies, ensuring the highest level of service delivery to your users.
- Authentication against a local database permits network access by employees.
- Flexible authentication options let you use your existing OS-based authentication database, token systems from RSA Security and other vendors, and external SQL/LDAP databases for remote and WLAN user authentication.
- Support for a wide variety of 802.1X-compliant access points and other network access servers ensures compatibility in your network environment.
- Advanced proxy features let you authenticate users against RADIUS servers at other sites.
  - You have a choice of username format, and you can configure routing based on username decoration, DNIS, or specific attributes.
  - You can selectively modify attributes as proxy packets flow to and from Steel-Belted Radius.
  - You can specify groups of proxy target servers that handle proxy requests according to load-balancing or retry strategies — for the best performance and reliability.
- You can control the time periods during which each user is allowed access. An access request is granted only during a user’s allowed access hours; otherwise it is refused, even if the user presents valid credentials.
- You can configure Steel-Belted Radius by means of a graphical SBR Administrator or by means of LDAP (either programmatically or at the command line prompt).

- Administrative access levels can be defined and applied to user or group accounts on the server machine. Read, write, and read/write access can be applied selectively to various categories of configuration data, including users, RADIUS clients, proxy targets, and statistics.
- Auto-restart enables Steel-Belted Radius to restart itself automatically if it experiences a shutdown.
- Linux: SNMP support lets you centrally monitor Steel-Belted Radius from your SNMP console, in the same manner as you monitor other devices and services on your network. Steel-Belted Radius offers full SNMP support including SNMP traps and alarms.
- IPv4 – IPv6 Dual stack support for RADIUS clients.
- Steel-Belted Radius Server acts as a TACACS+ Server in Linux platform GEE edition alone.

Licensing

If you want to install the Steel-Belted Radius server software for a 30-day evaluation, you do not need a license key.

If you want to install a permanent (non-evaluation) copy of Steel-Belted Radius, you must have a single-seat software license key.

If you have more than one copy of the Steel-Belted Radius software installed, you must have a site license key or you must have a separate license key for each installation.

The SBR Administrator can be installed on as many workstations as you require. The SBR Administrator does not require a license key.

For details about licensing, please refer to the Steel-Belted Radius license agreement or contact Pulse Secure.

ℹ Note: The Steel-Belted Radius license permits you to configure a total of 10 directed authentication and/or directed accounting methods. If you need additional methods, contact Pulse Secure to purchase blocks of additional licenses.

Chapter 2
RADIUS Basics

This chapter presents a conceptual overview of RADIUS (Remote Authentication Dial In User Service) authentication, authorization, and accounting services.

RADIUS Overview

RADIUS is an industry-standard protocol for providing authentication, authorization, and accounting services.

- Authentication is the process of verifying a user's identity and associating additional information (attributes) to the user's login session.
- Authorization is the process of determining whether the user is allowed on the network and controlling network access values based on a defined security policy.
- Accounting is the process of generating log files that record session statistics used for billing, system diagnosis, and usage planning.

A RADIUS-based remote access environment typically involves four types of components:

- An access client is a user who initiates a network connection. An access client might be a user dialing into a service provider network, a router at a small office/home office connecting to an enterprise network to provide network access, or a wireless client connecting to an 802.1X access point.
- A network access device (NAD), also called a RADIUS client, is a device that recognizes and processes connection requests from outside the network edge. A NAD can be a wireless access point, a modem pool, a network firewall, or any other device that needs to authenticate users. When the NAD receives a user's connection request, it might perform an initial access negotiation with the user to obtain identity/password information. The NAD then passes this information to the RADIUS server as part of an authentication/authorization request.

  Note: The terms "network access device" (NAD), "remote access server" (RAS), and "network access server" (NAS) are interchangeable. This manual use "NAD," though some attribute names and parameters retain the older "NAS" in their names.

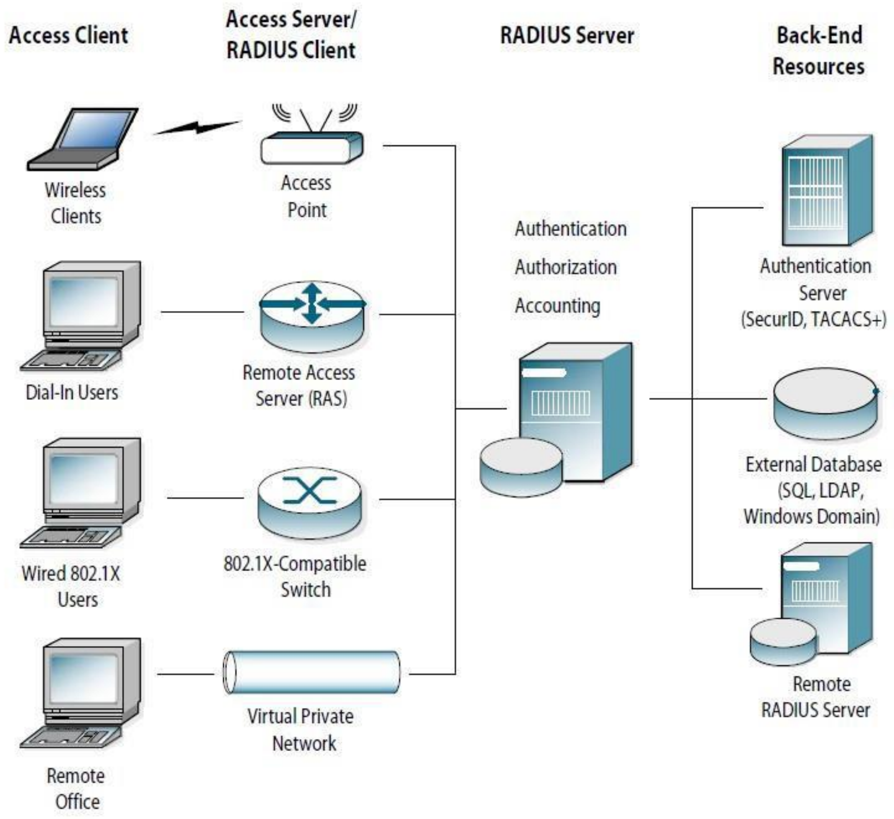
- The RADIUS server matches data from the authentication/authorization request with information in a trusted database, such as the database on the Steel-Belted Radius server or a backend database server. If a match is found and the user's credentials are correct, the RADIUS server sends an Access-Accept message to the NAD; if a match is not found or a problem is found with the user's credentials, the server returns an Access-Reject message. The NAD then establishes or terminates the user's connection. The NAD might then forward accounting information to the RADIUS server to document the transaction; the RADIUS server might store or forward this information as needed to support billing for the services provided.

- In some networks, a backend authentication server, such as RSA SecurID or TACACS+; a SQL or LDAP database; or some other RADIUS server for which this server is a proxy, stores the information against which the authentication request is compared. In some cases, the backend server passes information to the RADIUS server, which determines whether a match exists. In other cases, the matching is performed on the backend server, which then passes an accept/reject result to the

RADIUS server.

Figure 1: Radius Authentication illustrates a simple RADIUS environment.

Figure 1: Radius Authentication



RADIUS Packets

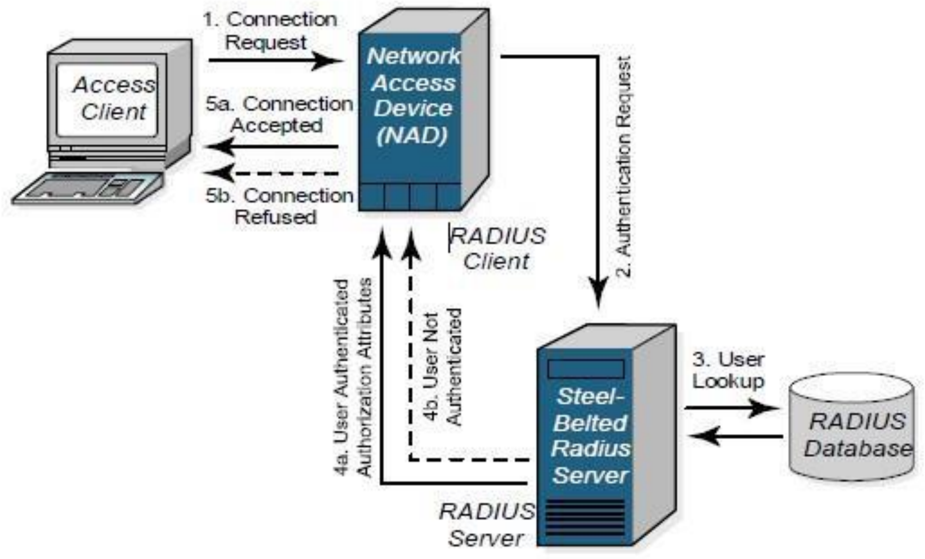
A RADIUS client and RADIUS server communicate by means of RADIUS packets. RADIUS packets carry messages between the RADIUS client and RADIUS server in a series of request/response transactions: the client sends a request and expects a response from the server. If the response does not arrive, the client can retry the request periodically.

Each RADIUS packet supports a specific purpose: authentication or accounting. A packet can contain values called attributes. The specific attributes to be found in each packet depend upon the type of packet (authentication or accounting) and the device that sent it (for example, the specific make and model of the NAD device acting as a RADIUS client).

For information on RADIUS authentication packet structures and attributes, see RFC 2865, Remote Authentication Dial In User Service (RADIUS). For information on RADIUS accounting packet structures and attributes, see RFC 2866, RADIUS Accounting.

Figure 2 illustrates a simple RADIUS authentication/authorization sequence.

Figure 2: Radius Authentication


1. Connection Request
Network Access Device (NAD)
Access Client
5a. Connection Accepted
5b. Connection Refused
RADIUS Client
2. Authentication Request
4a. User Authenticated Authorization Attributes
4b. User Not Authenticated
3. User Lookup
Steel-Belted Radius Server
RADIUS Database
RADIUS Server

1. The RADIUS access client sends an authentication request containing identification and connection information to the network access device (RADIUS client).

2. When the NAD receives a user's connection request, it typically performs an initial access negotiation with the user to establish connection information (username, password, network access device identifier, NAD port number, and so on). The NAD then forwards the user information in an authentication request to the RADIUS server.

3. The RADIUS server looks up the user information in a local or remote RADIUS authentication database. The RADIUS server verifies that the user's name and password are valid. It can also enforce fine-grained security rules by using an access checklist to verify specific attributes in the authentication request.

4. If a match is found, the RADIUS server returns an Access-Accept message (4a). The RADIUS server might also send return list information stored in the database, such as the user's authorization or connection parameters, back to the NAD.

   If a match is not found, the RADIUS server returns an Access-Reject message (4b).

   If third-party software such as RSA SecurID is used, the RADIUS server might prompt the user for more information before accepting or rejecting the authentication request.

5. Based on the information it receives from the RADIUS server, the NAD accepts or refuses the connection request.

After the user is authenticated and the connection established, the NAD might forward accounting data to the RADIUS server to document the transaction; the RADIUS server can store or forward this data to support billing for the services provided.

RADIUS Ports

The RADIUS standard initially used UDP ports 1645 and 1646 for RADIUS authentication and accounting packets.

The RADIUS standards group later changed the port assignments to 1812 and 1813, but many organizations still use the old 1645/1646 port numbers for RADIUS.

Any two devices that exchange RADIUS packets must use compatible UDP port numbers. That is, if you are configuring a NAD to exchange authentication packets with a RADIUS server, you must find out which port the server uses to receive authentication packets from its clients (1812, for example). You must then configure the NAD to send authentication packets on the same port (1812). The same is true for RADIUS accounting.

Steel-Belted Radius can listen on multiple ports. For compatibility, the server listens to the old and new default RADIUS ports: ports 1645 and 1812 for authentication, and ports 1646 and 1813 for accounting. To add, change, or disable the ports on which Steel-Belted Radius listens, modify the radius.ini file or edit the services (Windows) or /etc/services (Linux) file. The radius.ini file and the services file are described in the Steel-Belted Radius Reference Guide.

RADIUS Configuration

You must configure a RADIUS client and RADIUS server before they can communicate. If the client and server are on the same network, one administrator might be able to configure both sides of the RADIUS communication. If the client and server are not administered by the same person, you might have to coordinate RADIUS configuration details with the administrators of other networks.

RADIUS Server Configuration

To configure Steel-Belted Radius to respond to RADIUS clients, run the SBR Administrator, open the RADIUS Clients panel, and enter the following information for each RADIUS client:

- The IPv4/IPv6 addresses of the client device.
- The RADIUS shared secret used by Steel-Belted Radius and the client device. For information on RADIUS shared secrets, see “Shared Secrets”.
- The make and model of the client device, selected from a list of devices that Steel-Belted Radius supports. If a specific make and model is not listed, select **- Standard Radius -.**

Additionally, you must configure the UDP ports the server will use when sending and receiving RADIUS authentication and accounting packets. The UDP ports you configure on the RADIUS server must match the UDP ports that the RADIUS client is using for the same purposes. For more information, see “RADIUS Ports”.

RADIUS Client Configuration

You must tell each RADIUS client how to contact its RADIUS server. To configure a client to work with a Steel-Belted Radius server, log in to the client device, run its administration program, bring up its RADIUS configuration interface, and enter the following information:

- The IP address of the Steel-Belted Radius server.
- The RADIUS shared secret to be used by Steel-Belted Radius and the client device. For information on RADIUS shared secrets, see “Shared Secrets”.
- The UDP ports on which to send and receive RADIUS authentication and accounting packets. These must match the UDP ports that Steel-Belted Radius is using for the same purposes. For more information, see “RADIUS Ports”.

Multiple RADIUS Servers

You can distribute the RADIUS workload among several servers, as follows:

- You can set up separate servers for RADIUS authentication and accounting services. When RADIUS authentication and accounting services are performed by separate servers, each client device must be configured to send its authentication packets to one RADIUS server and its accounting packets to another.
- You can provide redundancy by pairing RADIUS servers to work in tandem. Most NAD configuration interfaces permit you to designate primary and secondary servers for authentication and accounting.

If both measures for distributing the RADIUS workload are implemented, client configuration involves identifying four servers for each client device: a primary RADIUS accounting server, a secondary RADIUS accounting server, a primary RADIUS authentication server, and a secondary RADIUS authentication server.

Shared Secrets

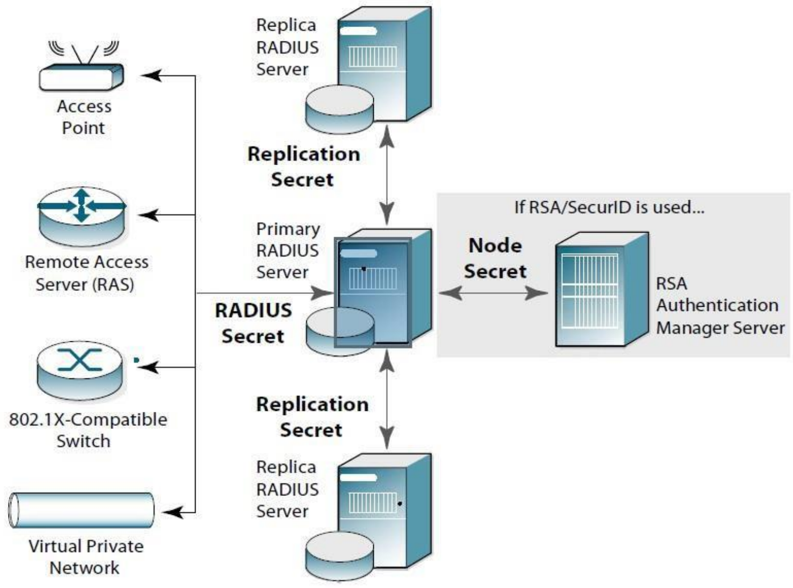
A shared secret is a case-sensitive text string used to validate communications between two RADIUS devices. You should configure shared secrets that are long enough and random enough to resist attack, and you should avoid using the same shared secret throughout your network. To maximize the security of your server's shared secret, consider using Pulse Secure's free Password Amplifier utility, which takes an ordinary shared secret or password (swordfish) and hashes it repeatedly to produce a 16-character amplified secret (g8QvQuRgRsl1AQ1E). You can paste this amplified secret into your server configuration to maximize security.

ⓘ Note: For more information on Pulse Secure's free Password Amplifier utility, see **https://www.pulsesecure.net/support.**

Steel-Belted Radius uses three types of shared secrets:

- RADIUS secret – Used to authenticate communication between a RADIUS server and a RADIUS client
- Replication secret – Used to authenticate communication between a primary server and a replica server
- Node secret – If you use RSA SecurID, Steel-Belted Radius uses a node secret to authenticate communication between a RADIUS server and an RSA Authentication Manager server.

Figure 3: Shared Secrets



RADIUS Secret

A RADIUS shared secret is a case-sensitive password used to validate communications between a RADIUS server, such as Steel-Belted Radius, and a RADIUS client, such as a network access device. Steel-Belted Radius supports shared secrets of up to 127 alphanumeric characters, including spaces and the following special characters:

~!@#$%^&*()_+|\=-‘{}[]:”’;<>?/.,

Identical shared secrets must be configured on both sides of the RADIUS communication link.

Note: Not all network access devices support shared secrets of up to 127 alphanumeric/special characters. You should select shared secrets that are fully supported by RADIUS devices in your network.

Most RADIUS clients allow you to configure different secrets for authentication and accounting. On the server side, the configuration interface allows you to create a list of known RADIUS clients (network access devices). You should be able to identify the authentication shared secret and accounting shared secret that a server uses to communicate with each of the clients on this list.

During an authentication transaction, password information must be transmitted securely between the RADIUS client (network access device) and Steel-Belted Radius. Steel-Belted Radius uses the authentication shared secret to encrypt and decrypt password information.

No encryption is involved in transmitting accounting data between a RADIUS client and RADIUS server. However, the accounting shared secret is used by each device to verify that it can “trust” any RADIUS communications it receives from the other device.

Replication Secret

A replication secret is a text string used to authenticate communications between a primary server and a replica server. You do not need to configure the replication secret for a realm: the primary server generates it automatically, and each replica server in a realm receives the replication secret as part of its configuration package.

See "About Replication for information on primary and replica servers.

Node Secret

If you use Steel-Belted Radius with RSA SecurID, the RSA Authentication Manager software views the Steel-Belted Radius service as a host agent. You must configure a node secret to authenticate communication between Steel-Belted Radius and the RSA Authentication Manager. A node secret is a pseudorandom string known only to the Steel-Belted Radius and RSA Authentication Manager. Before the Steel-Belted Radius sends an authentication request to the RSA Authentication Manager, it encrypts the data using a symmetric node secret key.

Accounting

A NAD can issue an Accounting-Request whenever it chooses, for example upon establishing a successful connection. Each time an Accounting-Request message arrives at the Steel-Belted Radius server, an accounting transaction begins. During this transaction, the server handles the message by examining the Acct-Status-Type and other attributes within the message, and taking the appropriate action.

Attributes

You work with RADIUS attributes while setting up users, profiles, and RADIUS clients in Steel-Belted Radius. The SBR Administrator lets you select RADIUS attributes by name from a predefined list. For each attribute, the SBR Administrator prompts you to enter values using familiar data types such as string, integer, telephone number, or network address.

Dictionaries

Steel-Belted Radius uses dictionary files to store lists of RADIUS attributes. Steel-Belted Radius uses these dictionaries to parse authentication/accounting requests and generate responses.

The main Steel-Belted Radius dictionary file (radius.dct) lists attributes defined by the RADIUS standard. The radius.dct file resides in the same directory as the Steel-Belted Radius service/process (usually C:\Program Files (x86)\Pulse Secure\Steel-Belted Radius\Service on Windows computers or /opt/PSsbr/Radius on Linux computers).

Vendor-Specific Attributes

In addition to the standard attributes, many network access devices use Vendor-Specific Attributes (VSAs) to complete a connection. Steel-Belted Radius supports a large number of specific network access devices by providing vendor-specific, proprietary dictionary files. These files also reside in the server directory and use the filename extension .dct.

Dictionaries and the Make/Model Field

During Steel-Belted Radius configuration, when you make a selection in the RADIUS Client Make/Model field, you are telling the server which dictionary file contains the VSAs for this client device. Thereafter, whenever the server receives a RADIUS packet from this client device, it can consult this dictionary file for any nonstandard attributes that it encounters in the packet. Standard RADIUS attributes are always defined by the radius.dct file. If you are not sure which make/model you should specify for a RADIUS Client, choose - Standard RADIUS -. The selections available in the Make/model field identify devices whose vendors have provided attribute dictionaries for use with Steel-Belted Radius.

Updating Attribute Information

If your NAD vendor announces a new product, a new attribute, or a new value for an attribute, you can add this information to your Steel-Belted Radius configuration. You can edit the dictionary file for that vendor to add new attributes or attribute values, or you can create a new vendor-specific dictionary file that contains new attributes and values. For more information on dictionary files, refer to the Steel-Belted Radius Reference Guide.

User Attribute Lists

Each user entry in the Steel-Belted Radius database provides the information necessary for the server to try to authenticate a connection request using a specific authentication method. When you view a user entry using the SBR Administrator program, this method is identified in the User type field.

You can control authentication at finer levels of detail than simple username/ password checking allow. The checklist, return list, or profile fields in the user entry in the database provide powerful tools for the authentication and authorization of users. These fields tell the server how to handle RADIUS attributes while authenticating a connection request and can be used to configure the authorization of the session.

Checklist Attributes

A checklist is a set of attributes that must accompany the authentication request before the request can be accepted. The NAD must send attributes that match the checklist associated with a user entry; otherwise, Steel-Belted Radius rejects the user even if the user's name and password are valid. By including appropriate attributes in the checklist, a variety of rules can be enforced. For example, only specific users might be permitted to use ISDN or dial-in connections to a particular NAD, or Caller ID might be used to validate a user against a list of acceptable originating telephone numbers. A checklist is created by selecting attributes from a list of all RADIUS attributes known to the Steel-Belted Radius server. This list can include a variety of vendor-specific attributes. During authentication, Steel-Belted Radius filters the checklist based on the dictionary for the RADIUS client that sent the authentication request. The server ignores any checklist attribute that is not valid for this device.

Return List Attributes

A return list is a set of attributes that Steel-Belted Radius must return to the NAD after authentication succeeds. The return list usually provides additional parameters that the NAD needs to complete the connection, typically as part of PPP negotiations. Return list attributes can thus considered to be "authorization configuration parameters."

By including appropriate attributes in the return list, you can create a variety of connection policies. Specific users can be assigned particular IP addresses or IPX network numbers; IP header compression can be turned on or off; or a time limit can be assigned to the connection.

You create a return list by selecting attributes from a list of all RADIUS attributes known to Steel-Belted Radius. This list can include a variety of vendor-specific attributes.

During authentication, Steel-Belted Radius filters the return list based on the dictionary for the RADIUS client that sent the authentication request. The server omits any return list attribute that is not valid for this device.

Attribute Values

The value of each RADIUS attribute has a well-defined data type: numeric, string, IP or IPX address, time, or hexadecimal. For example, Callback-Number is of type string and contains a telephone number, while NAS-Port-Type is an item from a list, and can be Sync, Async, and so forth.

ⓘ Note: Steel-Belted Radius supports signed integers (negative numbers) for attributes received in packets and processing relating to those attributes. However, SBR Administrator does not support signed integers, and treats signed and unsigned integers as unsigned integers.

Single- and Multi-Valued Attributes

Attributes can be single- or multi-valued. Single-valued attributes appear at most once in the checklist or return list; multi-valued attributes might appear several times.

If an attribute appears more than once in the checklist, this means that any one of the values is valid. For example, you can set up a checklist to include multiple telephone numbers for attribute Calling-Station-ID. A user trying to dial into your network would then have to call from one of the designated telephone numbers to be authenticated.

If an attribute appears more than once in the return list, each value of the attribute is sent as part of the response packet. For example, to enable both IP and IPX header compression for a user, the Framed-Compression attribute should appear twice in the return list: once with the value VJ-TCP-IP-header-compression and once with the value IPX-header-compression.

Orderable Multi-Valued Attributes

Certain multi-valued return list attributes are also orderable, which means the attribute can appear more than once in a RADIUS response, and the order in which the attributes appear is important. For example, the Reply-Message attribute allows text messages to be sent back to the user for display. A multi-line message is sent by including this attribute multiple times in the return list, with each line of the message in its proper sequence.

System Assigned Values

Some attributes do not allow the administrator to set a value. Steel-Belted Radius retrieves the appropriate value for this attribute when it is needed.

Echo Property

Using the echo property, you can force an attribute from the RADIUS request to be echoed in the RADIUS response. For example, you might add Callback-Number to the return list and select the echo check box. Steel-Belted Radius takes the value of the Callback-Number it receives in the RADIUS request and echoes it back to the client in the RADIUS response; if it receives no Callback-Number, it echoes nothing.

You enter Callback-Number one or more times into the checklist. This indicates that one of the callback numbers you supplied must be present in the RADIUS request, and that number should be echoed in the RADIUS response.

Default Values

Selecting default for a checklist attribute specifies that, if the RADIUS request does not include this attribute, the request should not be rejected. Instead, the value supplied as the default should be used as if it were received as part of the request. One use for default values is to require that an attribute in a RADIUS request must have one of several values, or must not be present at all. Another use is to provide a default value for an attribute in conjunction with the echo property in the return list.

Steel-Belted Radius can provide alternate values when an attribute appears in the checklist marked as default, and the same attribute appears in the return list marked as echo. The server echoes the actual value of the attribute in the RADIUS response if the attribute appears in the RADIUS request and echoes the default value (from the checklist) in the response if the attribute does not appear in the RADIUS request.

If you add multiple values of the same attribute to the checklist, only one of them can be marked as default.

For example, an administrator adds several Callback-Number values to the checklist and marks one of them as default. The administrator adds Callback-Number to the return list and specifies it as echo.

- If a Callback-Number value is present in the RADIUS request, it must match one of the checklist values or the user is rejected.

- If it does match, the user is accepted and the value supplied is echoed in the RADIUS response.
- If no Callback-Number is supplied in the request, the user is accepted and the default value is echoed in the response.

Other checklist attributes are used to provide configuration for the user, such as time-of-day and concurrent-login-limit information.

Wildcard Support

Steel-Belted Radius supports wildcards (? and *) for string-type attributes in checklist items and for IP addresses using a network number.

To allow backward compatibility with checklist items that treat the string literally, a string containing wildcards must be prefixed with a caret (^). When the caret is present, the remainder of the string is parsed using escape rules.

A ? wildcard matches any character and a * wildcard matches the remainder of the string (but can appear only at the end of a string). Wildcard characters can be treated as literals by using escape codes (for example, \?). **Table 3** lists the non-ASCII characters that can also be present in the wildcard string:

Table 3: Non-ASCII Characters in Wildcard Strings

Code	Meaning
\a	BEL
\b	Backspace
\f	Form feed
\n	Linefeed
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
\\	Backslash
*	Literal '*' (not wildcard)
\?	Literal '?' (not wildcard)
\xnn	Where nn is a hexadecimal value

Code	Meaning
\nnn	Where nnn is a decimal value

A '\' followed by any other character represents that character's value.

The following is a wildcard example for string type attributes:

    Called-Station-ID = ^800*

where Called-Station-ID indicates any 800 number.

The following is a wildcard example for IP Addresses:

    NAS-IP-Address = 199.100.10.0

where NAS-IP-Address indicates any IP address on the 199.100.10.0 network.

Attribute Filtering

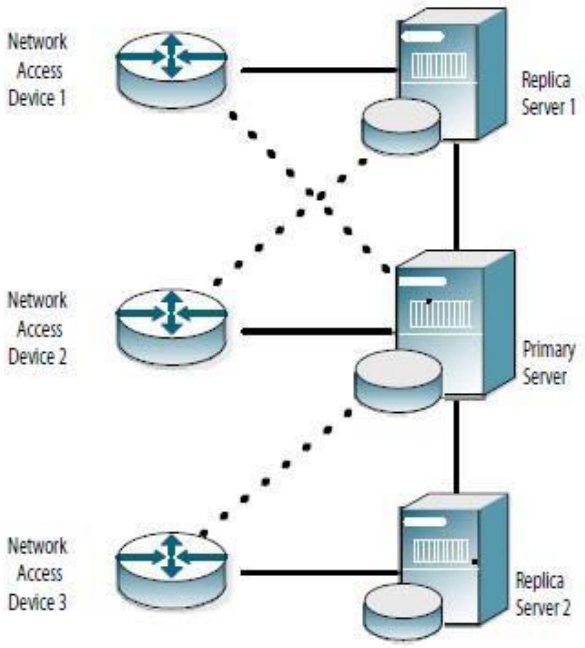
You can filter specific RADIUS attribute/value pairs into and out of RADIUS packets as they travel to and from directed realms and proxy RADIUS realms. Attribute filtering can be useful if there is data in the packets that is needed for routing, but not for authentication or accounting.

Centralized Configuration Management

Steel-Belted Radius supports the replication of RADIUS configuration data from a primary server to a maximum of 10 replica servers within a replication realm. Replica servers help balance the load of authentication requests coming in from RADIUS clients, and ensure that authentication services are not interrupted if the primary or other replica servers stops working.

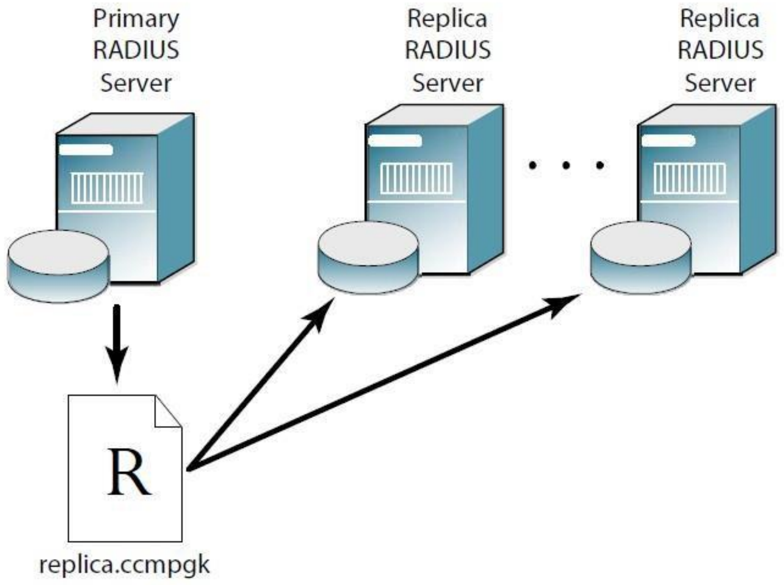
For example, Figure 4: Using Replication for Redundancy and Load Balancing llustrates an environment where RADIUS traffic is load-balanced by configuring each network access device to authenticate users through a different RADIUS server (solid line). If a RADIUS server becomes unavailable, the NAD can fail over to its backup RADIUS server (dotted line).

Figure 4: Using Replication for Redundancy and Load Balancing



All the servers within a realm reflect the current configuration specified by the network administrator: the network administrator modifies the configuration on the primary server, and the primary server propagates the new configuration to its replica servers. For example, after a network administrator configures a new RADIUS client or profile on the primary server, the network administrator tells the primary server to publish a configuration package file (replica.ccmpkg) that contains the updated configuration information. After publication, the primary server notifies each replica server that a new configuration package is ready. Each replica then downloads and installs the configuration package to update its settings.

Figure 5: Configuration Package Publication



The primary server maintains a list of the replica servers that have registered with it. The primary server uses this list to track which servers to notify after it publishes an updated configuration package to resynchronize the configuration of replica servers.

If the primary server needs to be taken out of service, the network administrator promotes one of the replica servers to be the new primary server. Thereafter, the other replica servers copy the configuration package from the promoted primary server.

Proxy RADIUS

The Steel-Belted Radius server can forward a RADIUS request to another server for processing and relay the other server's result back to its client. In such cases, Steel-Belted Radius is acting as a proxy for the target server, and Steel-Belted Radius is proxy-forwarding the request to the target server.

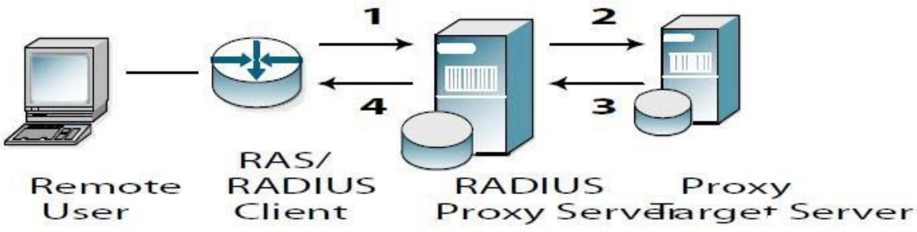
Steel-Belted Radius supports proxy RADIUS; any Steel-Belted Radius server can act as proxy or target for authentication or accounting messages.

Proxy RADIUS Authentication

RADIUS authentication messages are proxy-forwarded as follows:

1. An access client requests authentication from a RADIUS client, which sends an authentication request to a RADIUS proxy server.
2. The proxy RADIUS server forwards the message to a RADIUS target server.
3. The target RADIUS server performs the authentication services indicated by the message, then returns a response message to the proxy RADIUS server.
4. The proxy RADIUS server relays the acknowledgement response message to the RADIUS client.

Figure 6: Radius Proxy Forwarding



Proxy RADIUS Accounting

RADIUS accounting messages are proxy-forwarded as follows:

1. A RADIUS server receives an accounting request.
2. Depending on its configuration, the RADIUS proxy server forwards the accounting message to a target accounting server or records accounting attributes locally (or does both).
3. If the proxy server does not receive an acknowledgement of the forwarded accounting message, it re-sends periodically according to its retry policy.
4. When the target server acknowledges the request, the proxy server forwards an acknowledgement to the RADIUS client.

Authentication

RADIUS uses different types of messages during user authentication. **Table 4** summarizes the conditions under which each type of RADIUS authentication message is issued, and the purpose of any RADIUS attributes the message contains.

Table 4: RADIUS Authentication Messages and Attributes

Message Conditions	Purpose of Message Attributes
When a NAD receives a connection request from a user, the NAD requests authentication by sending an Access-Request to its RADIUS server.	Identify the user. Describe the type of connection the user is trying to establish.
When a RADIUS server is able to authenticate a user, it returns a RADIUS Access-Accept to the NAD.	Allow the NAD to complete access negotiations. Configure connection details such as providing the NAD with an IP address it can assign to the user. Enforce time limits and other "class of service" restrictions on the connection.
When a RADIUS server is unable to authenticate a connection request, it returns an Access-Reject to the NAD.	Terminate access negotiations. Identify the reason for the authorization failure.
If initial authentication conditions are met but additional input is needed from the user, the RADIUS server returns an Access-Challenge to the NAD.	Enable the NAD to prompt the user for more authentication data. Complete the current Access-Request, so the NAD can issue a new one.

Authentication Methods

Each time an Access-Request message reaches the server, an authentication transaction begins. During this transaction, the server attempts to authenticate the request by sequentially trying its configured and enabled authentication methods. The server consults its list of authentication methods to determine which methods to try and in which order to try them.

Native User Authentication

Native user authentication references user accounts stored on the Steel-Belted Radius server. When trying the native user method, Steel-Belted Radius searches its database for an entry whose User-Type is Native User, and whose username matches the User-Name in the Access-Request.

- If the entry cannot be found, or if it is found and the password is invalid, Steel-Belted Radius tries the next enabled method in the authentication methods list.
- If an entry for the user is found but the entry's checklist does not match attributes found in the Access-Request, Steel-Belted Radius returns an Access-Reject message to the NAD.
- If the entry is found and its password and checklist match perfectly, Steel-Belted Radius formats an Access-Accept message using the entry's return list, and returns it to the NAD.

Pass-Through Authentication

Pass-through authentication methods permit Steel-Belted Radius to begin the authentication by asking another entity to validate the username and password found in the Access-Request.

Steel-Belted Radius can pass authentication requests through to a Windows security database, RSA Authentication Manager (RSA SecurID), or TACACS+ server.

Proxy RADIUS Authentication

Steel-Belted Radius can convey an Access-Request to some other RADIUS server, which then (1) attempts to authenticate the connection request according to its own conventions and (2) returns a response to Steel-Belted Radius. Steel-Belted Radius then relays this response to the NAD. The set of conventions for relaying packets between cooperating RADIUS servers is known as proxy RADIUS.

External Authentication

External authentication methods enable Steel-Belted Radius to authenticate users by referring to external SQL or LDAP databases. During external authentication, Steel-Belted Radius queries the database for authentication data, and uses the results to format a response packet. Steel-Belted Radius then relays this response to the NAD.

For information on using Steel-Belted Radius with SQL databases, see "Configuring SQL Authentication". For information on using Steel-Belted Radius with LDAP databases, see "Configuring LDAP Authentication".

Directed Authentication (GEE only)

Every authentication request works its way through the same Authentication Methods list until one of the methods succeeds or the end of the list is reached.

This behavior might not be ideal for every user (GEE). If you want requests from certain users or accounts to bypass the master Authentication Methods list and use an alternate list, you can do so by employing the directed authentication feature. This feature allows you to map the User-Name or DNIS information in an incoming authentication request to a specific list of authentication methods. The list can include any native, pass-through, proxy-as-authentication, or external database authentication method configured on the Steel- Belted Radius server.

You can also direct authentication towards a particular realm using a technique called attribute mapping. This allows you to check for the presence or absence of a particular attribute in an authentication request, or for an attribute containing a specific value. Attribute mapping can be used with both proxy realms and directed realms.

HTTP Digest Access Authentication

HTTP Digest Access authentication provides a simple challenge-response authentication mechanism that an HTTP server can use to challenge an HTTP client request.

Steel-Belted Radius supports two forms of HTTP Digest Access authentication:

- HTTP Digest Access authentication, which is described in draft-ietf-radext-digest-auth-05.txt, uses 13 new RADIUS attributes to authenticate access requests from an HTTP server. When HTTP Digest Access authentication is used:

    a. An HTTP client sends a request without an authorization header to an HTTP server.

    b. The HTTP server sends a challenge containing a random value (nonce) to the HTTP client.

    c. The HTTP client creates an MD5 hash containing the username, password, nonce value, and other information, and returns this MD5 hash to the HTTP server in a request with an authentication header.

d. The HTTP server sends an Access-Request message containing special RADIUS attributes to Steel-Belted Radius.

e. Steel-Belted Radius verifies the HTTP client's credentials and returns a RADIUS Access-Accept or Access-Reject message to the HTTP server.

- The Ericsson ViG version of HTTP Digest Access authentication, which is described in **draft-sterman-aaa-sip-05.txt,** uses two Ericsson vendor-specific attributes (a Digest-Response attribute and one or more Digest-Attributes attributes) to authenticate access requests. When Ericsson ViG HTTP Digest Access authentication is enabled, Steel-Belted Radius looks for the ViG VSAs when it parses incoming packets, and, if it finds them, converts them to AVPs compatible with HTTP Digest Access authentication.

You must edit settings in the radius.ini file to enable HTTP Digest Access authentication.

Authenticate-Only Requests

Steel-Belted Radius supports requests to authenticate a user where the server performs no other processing. The NAD specifies this type of request by setting the Service-Type field to a value of Authenticate-Only (numeric value 8). The server responds with either an Access-Reject or an Access-Accept (without any attributes).

You can disable this feature (so that attributes are always returned in the response packet) by setting the AuthenticateOnly field in the [Configuration] section of the radius.ini file to 0. For more information on radius.ini, refer to the Steel-Belted Radius Reference Guide.

Configuring the Authentication Sequence

After you configure authentication methods for Steel-Belted Radius, the Authentication Policies panel in the SBR Administrator displays them in the order in which the server tries them. Enabled methods are displayed in black text; disabled methods are displayed in gray text. During an authentication transaction, the server works down the list, skipping disabled methods.

You can enable or disable methods or re-order methods in the list by using the controls in the Authentication Methods tab of the Authentication Policies panel. For information on setting up authentication sequences, see "Setting Up EAP Authentication Policies".

Configuring Authentication Methods

Each authentication method in Steel-Belted Radius performs a different type of processing on information in an incoming Access-Request packet. **Table 5** summarizes what you need to do to configure each authentication method.

Table 5: Authentication Method Configuration

Method	How to Configure	See
Native User	Create native user entries in the Steel-Belted Radius database	"**Setting Up Native Users**"
OS Pass-Through Security	This method assumes that you already have users, groups, and passwords defined in your local security database. Create user entries in the Steel-Belted Radius database. Choose User-types as appropriate.	"**Administering Users**"
RSA SecurID	This method assumes that you already have PIN/token code pairs defined on an RSA SecurID server. First, configure Steel-Belted Radius to communicate with	"**Setting Up SecurID Users**"

Method	How to Configure	See
	the RSA SecurID server. Then create user entries in the Steel-Belted Radius database. Choose SecurID User, <ANY>, SecurID Prefix, and SecurID Suffix User-types.	
TACACS+	This method assumes that you have username/ password pairs defined on a TACACS+ server. First, configure Steel-Belted Radius to communicate with the TACACS+ server. Then, create user entries in the Steel-Belted Radius database. Assign TACACS+ User, <ANY>, TACACS+ Prefix, and TACACS+ Suffix user-types.	"**Setting Up TACACS+ Users**"
Proxy RADIUS	Add a single target. You can set up single targets that are not associated with any realm. or: Identify Proxy RADIUS realms, each of which is a pool of proxy RADIUS target servers. Each time a RADIUS request arrives addressed to this realm, Steel-Belted Radius dynamically selects the appropriate target within the realm.	"**Administering Proxy RADIUS**" and "**Setting Up Proxy RADIUS Realms**"
EAP-TTLS	This method provides a means for an authentication request to be sent directly from the client to the server through a TLS connection. The act of establishing the TLS connection authenticates the server to the client and the authentication request sent through the tunnel authenticates the client to the server. Create a Steel-Belted Radius ttlsauth.aut file that specifies options for the TLS connection and the manner in which Steel- Belted Radius routes the inner authentication request. Stop and restart Steel-Belted Radius. Subsequently, the EAP-TTLS authentication method appears in the Authentication Methods tab in the Authentication Policies panel. You can use the Authentication Policies panel to enable, disable, and re-order EAP-TTLS methods.	"**EAP-TLS**"
External SQL Database	This method assumes that you have user records stored in a SQL database. Create a Steel-Belted Radius .aut file that connects to a SQL database and issues a SELECT query based upon the username and password. Give the .aut file a unique InitializationString value. Stop and restart Steel-Belted Radius. Subsequently, the SQL authentication method appears in the Authentication Methods tab of the Authentication Policies panel, using the InitializationString value as its name. You can use the Authentication Policies panel to enable, disable, and re-order the SQL authentication method.	"**Configuring SQL Authentication**"
External LDAP Database	This method assumes that you have user records stored in an LDAP database. Create a Steel-Belted Radius .aut file that validates the username and password based upon Bind and Search requests to an LDAP database. Give the .aut file a unique InitializationString value. Stop and restart Steel-Belted Radius. Subsequently, the LDAP authentication method appears in the Authentication Methods tab of the Authentication Policies panel, using the InitializationString value as its name. You can use the Authentication Policies panel to enable, disable, and re-order the LDAP authentication method as desired.	"**Configuring LDAP Authentication**"
Directed Authentication	For each directed authentication method that you want to configure, add an entry to the [Directed] section of proxy.ini and create a RealmName.dir file that specifies the mapping between the routing information that you expect in the authentication packet, and a list of locally- configured	"**Configuring a Directed Realm**"

Method	How to Configure	See
	authentication methods that you want to use.	

Advanced Options

Steel-Belted Radius provides the following additional authentication control options:

Account Lockout

Account lockout allows you to disable an account after a configurable number of failed login attempts within a configurable period. For example, if a user enters an incorrect password three times within two minutes, Steel-Belted Radius can lock out the user's account temporarily. During the lockout period, the user cannot log in, even with the correct password. When a user account is locked out, the user must wait until the expiration of the lockout period, or a network administrator can clear the lockout status for the account. For information on displaying and administering locked accounts, see "Using the Locked Accounts List".

Note: Do not enable account lockout and account redirection at the same time. If account lockout and account redirection are both enabled, account lockout is used and account redirection settings are ignored.

Note: Account lockout state is not maintained if Steel-Belted Radius is restarted.

Account Redirection

Account redirection allows you to flag an account for special processing after a configurable number of failed login attempts within a configurable period. For example, if a user enters an incorrect password three times within two minutes, Steel-Belted Radius can accept the user (even with an incorrect password) but limit the user's access to specific network resources, such as a secure web page that prompts the user to provide other authentication information. If the user can obtain his or her current password (or can create a new one through such a secure web page), he or she can then reconnect and log in successfully.

When account redirection is enabled and a user repeatedly enters an incorrect password, Steel-Belted Radius places the user in redirect state. When a user is in redirect state:

- If the user does not submit another authentication request within a specified time-out period, the user account is released from redirect state and returned to normal state.
- If the user submits another authentication request within a specified time-out period, the user is accepted without authentication/authorization processing. The accept message for the user includes the attributes and values specified in a redirection profile, and the user is placed into Access-Pending state. The attributes and values in the Access-Accept message are used by an external customer process, which might prompt the user to enter alternate authentication information to receive a password by email.
  When a user is in Accept-Pending state, the next authentication request received determines whether the user is accepted or locked out:
  - If the user enters the appropriate authentication information, the user is returned to normal state and Steel-Belted Radius generates an informational SNMP trap message.
  - If the user does not enter the appropriate authentication information, Steel-Belted Radius issues an Accept-Reject message, locks the user out of the network for a configurable lockout period, and generates an informational SNMP trap message. During this lockout period, authentication

requests for the user are automatically rejected, even if the user enters the correct password.

Optionally, you can identify RADIUS clients that you want to exclude from account redirection processing. Authentication requests from excluded RADIUS clients are processed normally, without use of redirection or account state changes.

Note: Do not enable account lockout and account redirection at the same time. If account lockout and account redirection are both enabled, account lockout is used and account redirection settings are ignored.

Note: Account lockout state is not maintained if Steel-Belted Radius is restarted.

Blacklisting

Blacklisting allows you to automatically reject authentication requests that contain certain values. You configure blacklisting by setting up a profile that identifies checklist attributes that should trigger an automatic authentication failure, and then modifying the blacklist.ini file to use that profile. For example, you could set up a profile to block users calling from a specific area code by configuring a Calling-Station-Id checklist attribute in the blacklist profile.

You can use * and ? wildcards in the blacklist profile. Blacklisting functions on all local authentication requests and can be configured to include proxy-RADIUS requests. For information on setting up profiles, see Administration Profiles.

For information on configuring the blacklist.ini file, refer to the Steel-Belted Radius Reference Guide.

Allowed Access Hours

Steel-Belted Radius provides a vendor-specific attribute called Funk-Allowed-Access-Hours. This attribute can be placed in the checklist for a user or profile entry to control the times during which a user can be allowed access.

Note: See "Allowed Access Hours" for the format of this value (and how to enter it into a user or profile record).

During authentication, the server processes the current time, the Funk-Allowed-Access-Hours value from the checklist, and the Session-Timeout value from the return list.

- If a Funk-Allowed-Access-Hours attribute is present in the checklist, and if the present time does not fall within a valid time period according to Funk-Allowed-Access-Hours, the server rejects the session.
- If a Funk-Allowed-Access-Hours attribute is present in the checklist, and if the current time falls within a valid time range according to Funk-Allowed-Access-Hours, the server accepts the session and calculates a session end time as follows:
    - If a Session-Timeout attribute exists in the user's return list, Steel-Belted Radius adds this number of seconds to the present time to calculate a proposed session end time. If the proposed session end time falls within the current time period (as defined by Funk-Allowed-Access-Hours), then Steel-Belted Radius returns the proposed session end time to the NAD in the Session-Timeout attribute.
      If the proposed end time occurs after the end of the current time period, then Steel-Belted Radius calculates the number of seconds between the present time and the end of the current time period (as defined by Funk-Allowed-Access-Hours, and returns this value to the NAD in the Session-Timeout attribute.

- If a Funk-Allowed-Access-Hours attribute is present in the checklist but a Session-Timeout attribute does not exist in the user's return list, Steel-Belted Radius computes a value for Session-Timeout based on the number of seconds between the present time and the end of the current time period. Steel-Belted Radius returns this value to the NAD in the Session-Timeout attribute.
- If Funk-Allowed-Access-Hours is not present in the checklist, the server returns the Session-Timeout value from the user's return list.
- If neither attribute is present, no Session-Timeout value is returned in the Access-Accept message, and the session is unlimited.

Two-Factor Authentication

Extensible Authentication Protocol-Tunneled Transport Layer Security (EAP-TTLS) provides for certificate-based mutual authentication between a client and a network through an encrypted tunnel. A typical implementation of EAP-TTLS uses certificates on authentication servers to create a network-to-user encryption tunnel, and then uses EAP inside the TLS tunnel for user-to-network authentication.

An enhanced version of EAP-TTLS uses certificates on the client side to provide two-factor authentication: the end user must have both a private key for a valid certificate and the password to an active account to obtain network access.

When client certificate support in EAP-TTLS is enabled on the server, you must provide a list of trusted root certificates from which offered client certificates must derive. These certificates must be provided in DER-encoded form and must be placed in the root subdirectory of the server directory.

Optionally, you can enable certificate revocation list (CRL) checking as part of the EAP-TTLS authentication process. CRL checking verifies that an unexpired certificate has not been revoked by its issuing Certificate Authority (CA) for any reason, such as a suspected security breach. Enabling CRL checking means that, every time the client requests a connection, Steel-Belted Radius checks the CRL to confirm that the client certificate has not been revoked. This improves security but increases processing overhead.

Note that, if client certificate support is not enabled in EAP-TTLS, any trusted root certificates and CRL checking options are ignored.

Password Protocols

During an authentication transaction, password information is transmitted between the NAD and the RADIUS server. This password information originally comes from the user, for example during PPP negotiations between a user and a NAD. Steel-Belted Radius supports four protocols (PAP, CHAP, MS-CHAP, and MS-CHAP-V2) for receiving the password from the NAD. Steel-Belted Radius also supports the Extensible Authentication Protocol.

Table 6 lists supported protocols according to the authentication methods with which each protocol can be used.

Table 6: Authentication Methods and Password Protocols

Method	PAP	CHAP	MS-CHAP	MS-CHAP-V2
LDAP	Yes	Yes, if BindName is used and the password is in clear text form or is encrypted with enc-md5.	Yes, if BindName is used and the password is in clear text form or is encrypted with enc-md5.	Yes, if the LDAP server can return clear-text password or MD4 hash of Unicode form of

Method	PAP	CHAP	MS-CHAP	MS-CHAP-V2
				password.
		No, if Bind is used	No, if Bind is used	No, if Bind is used
Local	Yes	Yes	Yes	Yes
Windows Domain Group	Yes	No	Yes, if the user is in local or trusted domain	Yes, if the user is in local or trusted domain
Windows Domain User	Yes	No	Yes, if the user is in local or trusted domain	Yes, if the user is in local or trusted domain
Proxy RADIUS	Yes	Yes	Yes	Yes
SecurID	Yes	No	No	No
SQL	Yes	Yes, if the password is available in clear text form in the database or is encrypted with enc-md5.	Yes, if the password is available in clear text form in the database or is encrypted with enc-md5.	Yes, can return clear-text password or MD4 hash of Unicode form of password.
TACACS+	Yes	Yes	No	No
UNIX User	Yes	No	No	No
UNIX Group	Yes	No	No	No

Password Authentication Protocol

When the Password Authentication Protocol (PAP) is used, a remote user negotiates with the NAD “in the clear,” and no encryption is used to send the password to the NAD. After the NAD has enough information from the user to create an Access-Request, the NAD encrypts the password (using its RADIUS shared secret) before sending an Access-Request packet to Steel-Belted Radius.

Upon receiving the Access-Request, Steel-Belted Radius looks for attributes within the packet that identify the NAD that sent it. Steel-Belted Radius decrypts the password by using the shared secret configured for the RADIUS client entry associated with the sending NAD.

Ultimately, Steel-Belted Radius has the password in clear text form for authentication.

Challenge Handshake Authentication Protocol

The Challenge Handshake Authentication Protocol (CHAP) avoids sending passwords in clear text over any communication link. Under CHAP, during password negotiations the NAD generates a challenge (a random string) and sends it to the user. The user’s PPP client creates a digest (the password concatenated with the challenge), encrypts the digest using one-way encryption, and sends the digest to the NAD.

The NAD sends this digest as the password in the Access-Request.

Because the encryption is one-way, Steel-Belted Radius cannot recover the password from the digest. Instead, it performs an identical operation, using the NAD’s challenge value (provided in the Access-Request packet) and its own copy of the user’s password to generate its own digest. If the two digests match, the password is the same.

Steel-Belted Radius must be able to perform the digest operation to support CHAP. Therefore, it must have access to its own copy of the user’s password. Native User passwords are stored in the Steel-Belted Radius database. SQL or LDAP BindName authentication retrieves the password by means of a query to the database;

the retrieved password can be used to create a digest if it is in clear text form. A TACACS+ server provides CHAP support and handles the digest operation itself after Steel-Belted Radius sends the username and password through. No other authentication methods support CHAP at this time.

MS-CHAP and MS-CHAP-V2

The two varieties of MS-CHAP (Microsoft Challenge Handshake Authentication Protocol) are Microsoft authentication protocols that, like CHAP, avoid sending passwords in clear text. Steel-Belted Radius supports both 40-bit and 128-bit MS-CHAP methods. Steel-Belted Radius must be able to perform a digest operation similar to CHAP to support MS-CHAP. Therefore, it must have access to its own copy of the user's password. Native User passwords are stored in the Steel-Belted Radius database. SQL or LDAP BindName authentication retrieves the password by means of a query to the database; the retrieved password can be used to create a digest if it is in clear text form.

MS-CHAP and MS-CHAP-V2 communicate users' requests to change their passwords to a RADIUS server. Steel-Belted Radius supports this feature, although it must also be supported by whatever application the user is using to log in.

MS-CHAP and MS-CHAP-V2 operate in the same way, but they use different attributes. An MS-CHAP client won't accept MS-CHAP-V2 attributes, and vice-versa; be careful to use the appropriate set of attributes.

For more information about MS-CHAP and MS-CHAP-V2, see RFC 2433, Microsoft PPP CHAP Extensions; RFC 2548, Microsoft Vendor-specific RADIUS Attributes; and RFC 2759, Microsoft PPP CHAP Extensions, Version 2.

Accounting

To understand the Steel-Belted Radius accounting sequence, you will need an overview of RADIUS accounting messages. **Table 7** explains the conditions under which each type of message is issued, and the purpose of any RADIUS attributes that a message contains.

Table 7: Message Conditions and Attributes

Message Conditions	Purpose of Message Attributes
The RADIUS client sends accounting data to Steel-Belted Radius using an Accounting-Request message. The RADIUS client is responsible for verifying that the server receives accounting requests. Most clients retry periodically until the server responds.	Depending on the value of the Acct-Status-Type attribute, the message type is considered to be Start, Stop, Interim-Acct, Accounting-On, or Accounting-Off.
Upon receipt of an Accounting-Request message, the server sends an Accounting-Response.	Complete the request/response cycle.
After receiving an Access-Accept from the server, the NAD completes its access negotiation with the user. The NAD then sends a Start message to the server.	Record connection data, such as username, NAD identifier, NAD port identifier, port type, and connection start time.
At intervals of approximately every six minutes, the NAD sends an Interim-Acct message to the server.	Record a "snapshot" of statistics regarding the connection. One message contains the current value of every statistic that this NAD is capable of recording about this type of connection.
After a connection is terminated, the NAD sends a Stop message to the server.	Record statistics regarding the connection. One message contains the final value of every statistic that this NAD is capable of recording about this type of connection.
Every time a client device comes online, whether after a crash or after an orderly shutdown, it sends an Accounting-On message to the	Identify the device that is going online and clear all session information.

Message Conditions	Purpose of Message Attributes
server.	
Every time a client device experiences an orderly shutdown, before completing its shutdown sequence it sends an Accounting-Off message to the server.	Identify the device that is going offline and clear all session information.

Accounting Sequence

A NAD can issue an Accounting-Request whenever it chooses, for example upon establishing a successful connection. Each time an Accounting-Request message reaches Steel-Belted Radius, an accounting transaction begins. During this transaction, the server handles the message by examining the Acct-Status-Type and other attributes within the message, and taking the appropriate action.

Comma-Delimited Log Files

When the Steel-Belted Radius accounting log is enabled, all of the RADIUS accounting attributes that the server receives are reformatted and logged to a comma-separated value (CSV) text file, which is easily imported into spreadsheets and database programs for report generation and billing.

Proxy RADIUS Accounting

Steel-Belted Radius can relay an Accounting-Request to some other RADIUS server, which records the data according to its own, locally-configured RADIUS accounting options. (You have the option of specifying that the data also be recorded locally on the Steel-Belted Radius server.) The set of conventions for relaying packets between cooperating RADIUS servers is known as proxy RADIUS, and is defined in the RADIUS standard.

See "Configuring a Proxy RADIUS Realm".

External Accounting

External accounting methods permit Steel-Belted Radius to record accounting data to external databases. Configuration files specify how Steel-Belted Radius communicates with an external database and how to insert accounting data into that database.

SQL is the only external accounting method currently supported by Steel-Belted Radius.

See "About SQL Accounting".

Tunneled Accounting

During authentication, a user is typically identified by attributes such as User-Name (in the authentication request) and Class (in the authentication accept response). Standard RADIUS accounting requests typically include these attributes in messages flagging Start, Interim, and Stop events so that the user's identity can be recorded for accounting and auditing purposes.

When an organization uses a tunneled authentication protocol such as EAP/TTLS or EAP/PEAP, the identity of a user requesting authentication might be concealed from the NAD; the User-Name attribute carried by the outer authentication protocol is typically a non-unique value such as "anonymous." As a result, the outer User-Name value included in accounting requests might not be sufficient to determine a user's identity. Class attributes provided by an authentication server cannot be included in cleartext in an outer Access-Accept message because they might contain clues about the user's identity, thereby defeating the identity-hiding feature of the tunneled protocol.

Tunneled accounting allows Steel-Belted Radius to pass user identity information to accounting processes without exposing user identities to a NAD that should not see them. When tunneled accounting is enabled, RADIUS attributes are encrypted and encapsulated in a Class attribute. If the information for a Class attribute exceeds the attribute payload size (253 octets), Steel-Belted Radius returns more than one Class attribute for a user.

The tunneled accounting transaction sequence is:

1. The Steel-Belted Radius server acting as the tunnel endpoint for EAP/TTLS or EAP/PEAP encrypts a user's inner User-Name and Class attributes when it authenticates the user.

2. The server returns the encrypted information to the NAD encapsulated in a Class attribute in the outer Access-Accept message. The NAD associates this encapsulated identity attribute with the user, and echoes the encapsulated identity attribute whenever it generates an accounting request for the user.

3. When Steel-Belted Radius receives an accounting request from a network access device, it scans the request for an encapsulated identity attribute.

4. If Steel-Belted Radius finds an encapsulated identity attribute, it de-encapsulates and decrypts the attributes to reconstitute the original inner User-Name and Class attributes.

5. Steel-Belted Radius substitutes the decrypted attributes for the ones returned from the NAD.

6. Steel-Belted Radius processes the accounting request locally or forwards the accounting request through the proxy to its intended target.

To implement tunneled accounting, you must configure the classmap.ini file to specify how attributes should be presented, and you must configure the spi.ini file to specify the keys that are used to encrypt and decrypt users' identity information. The classmap.ini file and the spi.ini file are described in the Steel-Belted Radius Reference Guide.

For an overview of how EAP/TTLS and EAP/PEAP work, refer to "About the Extensible Authentication Protocol".

Directed Accounting

The directed accounting feature allows you to map an incoming accounting request to one or more accounting methods, based on routing information found in the request packet. Among the options available with directed accounting is that of establishing an accounting log file that is distinct from the Steel-Belted Radius accounting log file in the server directory, and that contains entries from only those accounting requests that were specifically directed to the realm.

See "Configuring a Directed Realm".

Accounting Spooling

Accounting spooling can improve both proxy accounting performance and reliability. When spooling is enabled for a realm, Steel-Belted Radius immediately acknowledges all accounting requests for that realm to the NAD. Meanwhile, it spools accounting requests to a file while a separate thread unspools requests and sends them to the server responsible for the realm. If the server is unavailable, Steel-Belted Radius retries at regular intervals until the proxy target acknowledges the request. Even if Steel-Belted Radius restarts, all spooled requests are preserved until they are completed.

Account spooling offers the following benefits:

- The NAD always gets an immediate ACK (acknowledgement response) for accounting requests.

- Accounting data is never lost if it is sent to a Steel-Belted Radius server with spooling enabled.

A separate and independent spooler is maintained for each realm for which proxy spooling is configured. When an accounting request is received for a realm implementing proxy spooling, it is written to a file in the target directory and a request is prepared, followed by an acknowledgement returned to the client. The file is then read by the unspooling thread and the prepared request proxied.

Targets, fast-fail, round-robin, and other extended proxy features operate normally, but unspooling continues to retry sending a request until it is successfully acknowledged. Since each spooler is independent, one unresponsive realm does not affect the delivery of spooled requests to other realms.

The Acct-Delay-Time attribute in a request is updated or added as necessary if there is a delay between the spooling and the forwarding of the request.

When the spool file's rollover interval expires or the file size exceeds the rollover size limit, the current spool file is closed for writing and a new one created. Files are named in the format, yyyymmdd_hhmm_ssss.psf, where yyyy is the year, mm is the month, dd is the day, hh is the hour, mm is the minute, and ssss is a sequence number. The configuration of the rollover settings enables spooling to be optimized according to the characteristics of the operating environment.

When Steel-Belted Radius is shut down, unspooling continues for the configured ShutdownDelay time until all spooled packets are sent. If the destination server is down at the time of shutdown, however, unspooling terminates immediately. After startup, unspooling continues from the beginning of the oldest spool file.

Request Routing

When Steel-Belted Radius receives a RADIUS authentication or RADIUS accounting request, it examines the attributes in the request to match the request with the service that can best respond to it:

- RADIUS authentication or accounting
- Proxy RADIUS authentication or accounting
- Tunnel authentication
- Directed authentication or accounting

This matching process (request routing) uses the information in the User-Name attribute (in which routing information is supplied as a prefix or suffix to the user's account name), the Called-Station-Id attribute (DNIS), or other attribute(s) to route the request.

Steel-Belted Radius checks the User-Name and Called-Station-Id attributes, then it checks the attributes you have mapped to realms. It uses the first routing destination it finds.

Match Rules

In simple cases, each entry in the [Realms] or [Directed] sections of the proxy.ini file consists of the name of a domain. For prefix- and suffix-based realm matching, the domain name is taken from the User-Name attribute (for example, other.com in user@other.com). The domain name also supplies the root of the name of the configuration file that specifies the treatment for that realm (for example, other.com.pro or other.com.dir).

Match rules extend the format of lines in the [Realms] or [Directed] sections by allowing leading and trailing wildcard support and by allowing multiple entries to be mapped to the same realm.

Match rules in the [Realms] or [Directed] sections of the proxy.ini file take the form:

RealmName [= match_rule]

- Leading wildcards allow you to match any domain name that ends with the appropriate domain name string and map it to a realm name:

```
realm1 = *.msn.com
```

- Trailing wildcards allow you to match any domain name that begins with the appropriate domain name string and map it to a realm name:

```
realm2 = usa.*
```

- Direct matching allows you to map a domain name to a realm name:

```
realm3 = boston.other.com
```

A standalone wildcard character allows you to match any domain name that does not meet the criteria of a more specific rule:

```
realm4 = *
```

You can use match rules to map more than one entry in the [Realms] or [Directed] section of proxy.ini to a realm. When Steel-Belted Radius performs prefix or suffix processing of the User-Name, it isolates the realm portion of the User-Name attribute and then searches through the match rules to find the one that matches the realm information most closely. The best match is the one that specifies the largest number of non-wildcard characters in the rule.

For example, assume the following entries are present in the [Realms] or [Directed] section of your proxy.ini file:

```
realm1 = *.msn.com
realm2  =  usa.msn.com
realm3 = *.uk.msn.com
realm4 = *.com
realm5 = *
```

Table 8 identifies what realms Steel-Belted Radius would choose for different User-Name values.

Table 8: Realm Mapping Example

User-Name	Mapped Realm
bob@usa.msn.com	realm2
alice@scotland.uk.msn.com	realm3
susie@wales.uk.msn.com	realm3
fred@germany.msn.com	realm1
julie@indiana.usa.msn.com	realm1

User-Name	Mapped Realm
ramon@other.com	realm4
seema@other.edu	realm5

Realm names can appear more than once within the [Realms] or [Directed] section of proxy.ini to point multiple match rules to the same realm. For example, you could enter the following settings in the [Realms] or [Directed] section of your proxy.ini file:

```
realm1 = *.msn.com

realm2 = usa.msn.com

realm2 = uk.msn.com
```

These entries would map any User-Name with a domain of usa.msn.com or uk.msn.com to realm2, and map other domain ending in msn.com to realm1.

You cannot enter duplicate realm lines (with or without match rules) in the [Realms] or [Directed] section of your proxy.ini file, and you cannot enter the same match rule on multiple lines.

You cannot combine leading and trailing wildcards in the same match rule. If you do so, the trailing match rule is ignored.

User-Names with a Single Delimiter

An incoming User-Name string can be “decorated” with a single delimiter separating the user’s name from a destination name. A User-Name decorated in this manner can indicate a proxy RADIUS realm, a directed realm, a tunnel, or a proxy entry that is not a member of any realm.

Note: To prevent unexpected routing results, you must ensure that the name of every realm, tunnel, and proxy entry is unique across your entire Steel-Belted Radius configuration.

User-Names with a Single Tunnel Delimiter

If the delimiter matches the currently configured delimiter for tunnels, and if the current name-parsing convention for tunnels is suffix, the User-Name is understood to be:

```
User<SuffixDelimiter>TunnelName
```

If the current name parsing convention for tunnels is prefix, the User-Name is understood to be:

```
TunnelName<PrefixDelimiter>User
```

where:

- User is the name of the dial-in user
- TunnelName identifies the destination
- <SuffixDelimiter> or <PrefixDelimiter> is a delimiter character such as @, / or !

If a tunnel entry is found that matches the TunnelName, and the request is for authentication, Steel-Belted Radius proceeds with tunnel authentication.

Note: Tunnel delimiters are specified in the Name Parsing tab of the Tunnels panel (described in "Configuring RADIUS Tunnels"). You can use either the prefix or the suffix naming.

convention for tunnels, but not both. You can also choose the tunnel delimiter character ('@', '/', and so forth). The conventions you define in the Tunnels panel apply to all tunnels defined on the server.

User-Names with a Single Realm Delimiter

If the User-Name contains a single delimiter that matches the currently configured suffix delimiter for realm destinations, the User-Name is understood to be:

User<SuffixDelimiter>RealmName

If the User-Name contains a single delimiter that matches the currently configured prefix delimiter for realm destinations, the User-Name is understood to be:

RealmName<PrefixDelimiter>User

where:

- User is the name of the dial-in user
- RealmName identifies the destination
- <SuffixDelimiter> or <PrefixDelimiter> is a delimiter character such as @, / or !

Steel-Belted Radius attempts to find a destination that matches RealmName in one of four places, as summarized in **Table 9**.

Table 9: Realm Name Matching

If a match is found in:	Then Steel-Belted Radius does this:
[Self] section of the radius.ini file.	Steel-Belted Radius services the request locally.
The [Directed] section of the **proxy.ini** file.	Steel-Belted Radius routes the request to a specific authentication or accounting method on the local server, according to the rules in the corresponding **RealmName.dir** file.
The [Realms] section of the **proxy.ini** file.	Steel-Belted Radius routes the request to the proxy RADIUS realm called **RealmName** according to the rules in the corresponding **RealmName.pro** file. See "**Target Selection within a Realm**".
A proxy entry in the Steel-Belted Radius database	Steel-Belted Radius uses the information in the proxy entry (IP address, UDP port, shared secret) to forward the RADIUS request.

Note: Realm delimiters and naming conventions are defined in the proxy.ini file. You can define different delimiters for prefixes and suffixes. The conventions you define in proxy.ini apply to all types of realm defined on the server (both proxy realms and directed realms).

User-Names with Multiple Suffix Delimiters

If the User-Name contains multiple realm delimiters

(User<Delimiter>RealmName<Delimiter>RealmName<Delimiter> RealmName) and the delimiter character matches the current RealmSuffix setting in the [Configuration] section of proxy.ini, the name parsing strategy is as follows:

1. Steel-Belted Radius finds the leftmost RealmName in the User-Name that is also listed in the [Self] section of its radius.ini configuration file.

2. If a matching RealmName was found in Step 1, and there is no other RealmName to the left of it, then Steel-Belted Radius services the request locally, without forwarding.

3. If a matching **RealmName** was found in Step 1, but there is another RealmName to the left of it, then Steel-Belted Radius routes the request to the **RealmName** listed immediately to the left of the matching RealmName. The routing is controlled by the corresponding **RealmName.pro** or **RealmName.dir** file.

4. If no **RealmName** was selected in Steps 1, 2, or 3, then Steel-Belted Radius routes the request to the rightmost RealmName in the User-Name. The routing is controlled by the corresponding **RealmName.pro** or **RealmName.dir** file.

Table 10: Realms in User-Names

A request for...	Would be...
fred@bignet@bigserver	Routed to the realm called bignet.
fred@bignet@bigserver@smallnet	Routed to the realm called bignet.
fred@bignet@smallnet	Routed to the realm called smallnet.
fred@bigserver@bignet	Handled locally on bigserver.

User-Names with Multiple Prefix Delimiters

If the User-Name contains multiple realm delimiters:

RealmName<Delimiter>RealmName<Delimiter>RealmName <Delimiter>User

and the Delimiter character matches the current RealmPrefix setting in the [Configuration] section of proxy.ini, the name parsing strategy is the reverse of the suffix strategy described above. In detail:

1. Steel-Belted Radius finds the rightmost RealmName in the User-Name that is also listed in the [Self] section of its radius.ini configuration file.

2. If a matching RealmName was found in Step 1, and there is no other RealmName to the right of it, then Steel-Belted Radius services the request locally, without forwarding.

3. If a matching RealmName was found in Step 1, but there is another RealmName to the right of it, then Steel-Belted Radius routes the request to the RealmName listed immediately to the right of the matching RealmName. The routing is controlled by the corresponding RealmName.pro or RealmName.dir file.

4. If no RealmName was selected in Steps 1, 2, or 3, then Steel-Belted Radius routes the request to the leftmost RealmName in the User-Name. The routing is controlled by the corresponding RealmName.pro or RealmName.dir file.

According to these rules, if the realm prefix Delimiter character is '!', and the User-Name matches realm prefix naming conventions, and the [Self] section of radius.ini lists one realm called bigserver, then incoming User-Name values would be parsed as described in **Table 11**.

Table 11: Realms and User-Names

A request for...	Would be...
superserver!bignet!fred	Routed to the realm called **bignet.**
smallnet!bigserver!bignet!fred	Routed to the realm called **bignet.**
smallnet!bignet!fred	Routed to the realm called **smallnet.**
bignet!bigserver!fred	Handled locally on **bigserver.**

Undecorated User-Names

An undecorated User-Name is a User-Name that does not include realm identification information. When realm support for undecorated User-Names is enabled, Steel-Belted Radius routes authentication requests that contains undecorated User-Name attributes to the proxy realm or directed realm designated in the [Realms] or [Directed] section of the proxy.ini file.

Undecorated User-Name support allows you to specify a realm to handle any request containing a User-Name that does not contain realm identification information.

Steel-Belted Radius uses the following logic to determine if a User-Name is undecorated:

1. If Suffix is enabled in the [Processing] section of the proxy.ini file and the User-Name contains the specified RealmSuffix character, the User-Name is not undecorated. Note that, by default, @ is the default RealmSuffix character and that Suffix is enabled if the proxy.ini file does not include a [Processing]section.

2. If Prefix is enabled in the [Processing] section of the proxy.ini file and the User-Name contains the specified RealmPrefix character, the User-Name is not undecorated. Note that, by default, / is the default RealmPrefix character and that Prefix is enabled if the proxy.ini file does not include a [Processing]section.

3. If neither of the above statements is true, the User-Name is undecorated.

Configuring Undecorated User-Name Support

To configure undecorated User-Name support in Steel-Belted Radius:

1. Add an Undecorated entry to the [Processing] section of proxy.ini.
   If the proxy.ini file does not include a [Processing] section or if the Undecorated entry is commented out or not present, undecorated User-Name processing is disabled.

2. Associate an <undecorated> marker with a proxy realm (in the [Realms] section of proxy.ini) or a directed realm (in the [Directed] section of proxy.ini).

   Only one realm listed in the [Realms] or [Directed] section of proxy.ini can be configured with the = <undecorated> setting. If more than one realm is associated with the = <undecorated> setting, Steel-Belted Radius enables the first entry it finds and writes an error message identifying the duplicate realms to the server log file.

3. Verify that the private directory for Steel-Belted Radius includes a realm.pro file (for the proxy realm associated with undecorated User-Name processing) or realm.dir file (for the directed realm associated with undecorated User-Name processing) that matches the realm specified in Step 2..

   For example, if you enter other.com = <undecorated> in the [Realms] section of proxy.ini, you must have an other.com.pro file in the Steel-Belted Radius directory. If the applicable realm.pro file is not found, Steel-Belted Radius writes an error message noting the realm has not been enabled to the server log file

Example

The following sections of proxy.ini illustrate how undecorated User-Name support is configured in Steel-Belted Radius.

```
[Processing]
Undecorated
Suffix
Prefix
DNIS
Attribute-Mapping
[Realms]
bigco.com
partner.com
other.com = <undecorated>
```

In this example, any authentication request that contains undecorated User-Name attributes will be handled by the other.com realm. Successful authentication requests that are handled by the selected realm will result in a Class attribute that records the name of the realm so that accounting requests resulting from the session can be handled by the same realm.

Note that this is different from cases where a decorated User-Name (that is, a User-Name that contains a prefix or suffix delimiter) does not match explicit realm mapping rules. For example, since no matching rule for littlecorp.com is configured, a request containing a User-Name value of **user@littlecorp.com** would fall through to local processing and result in a rejection if no matching user is found.

Note: All settings are reloaded on receipt of a HUP signal.

Request Routing by DNIS

If the Called-Station-Id attribute is found in the RADIUS request, the request can be routed based on DNIS (Dialed Number Information Services). Steel-Belted Radius checks its administration database and server configuration files for a DNIS string that matches the value of the incoming Called-Station-Id attribute. If found, the matching string might be in one of three places:

- A tunnel entry's Called Station Id list. If a match is found here, and the request is for authentication, Steel-Belted Radius performs tunnel authentication. See "Tunnel Authentication Sequence".
- The [Called-Station-ID] section of a RealmName.dir file. If a match is found here, Steel-Belted Radius handles the request locally using the authentication and/or accounting methods identified in the RealmName.dir file. See "Configuring a Directed Realm".
- The [Called-Station-ID] section of a RealmName.pro file. If a match is found here, Steel-Belted Radius routes the request to the proxy RADIUS realm called RealmName using the rules defined in the RealmName.pro file. See "Target Selection within a Realm".

Note: We recommend that you use DNIS strings that are unique across all tunnel entries, all RealmName.dir files, and all RealmName.pro files. If you duplicate a DNIS string anywhere in your Steel-Belted Radius configuration, the request routing results might be unexpected.

Request Routing by Any Attribute

incoming packet to a specific realm, by providing an [AuthAttributeMap] or [AcctAttributeMap] section in the proxy.ini configuration file.

You can route all of the packets for a session to a realm based on attributes in the Access-Request (the [AuthAttributeMap] section), or you can route the session's accounting packets to a different realm, based on attributes found in these packets (the [AcctAttributeMap] section).

Attribute mapping can be used for proxy RADIUS realms and for directed realms. You cannot use this feature when forwarding packets to a proxy target that is not a member of a realm.

Local Services

If the RADIUS request does not contain routing information (or at least, it does not contain any routing information that Steel-Belted Radius has been configured to recognize), it is processed locally on the Steel-Belted Radius server. Authentication follows the authentication methods list in the server's Authentication Policies panel. No User-Name parsing is performed; the entire string is understood to be the user's name. Accounting is controlled by the server's main account.ini file and (for external database accounting) .acc file.

Control Over Routing Methods

By default, the rules for determining the destination of a request are applied in the following order by default:

1. Apply Suffix delimiter rules.
2. Apply Prefix delimiter rules
3. Apply DNIS rules
4. Apply Attribute Mapping rules

You can specify which methods you want used for request routing and the sequence in which methods are applied.

RADIUS Client Groups

If your RADIUS clients use the same RADIUS attributes and have contiguous IP addresses, you can configure one or more RADIUS client groups and specify an address range consisting of as many as 500 IP addresses for each client group. When Steel-Belted Radius receives a RADIUS request that includes a source IP address in this range, it uses the RADIUS client group to determine the appropriate shared secret, make/model, and IP address pool.

Please note the following when you set up address ranges for RADIUS client groups:

- Address ranges are for IPv4 networks only. Steel-Belted Radius does not support address ranges for IPv6 or IPX.
- The address range assigned to one RADIUS client group cannot overlap the address ranges assigned to other RADIUS client groups.
- The starting address of the address range assigned to a RADIUS client group cannot match the IP address of an individual RADIUS client.
- If an individual RADIUS client entry has an IP address that falls within an address range assigned to a RADIUS client group, Steel-Belted RADIUS uses the make/model for the individual RADIUS client. For example, if RADIUS client RAS1 is configured with IP address 192.168.21.55 and if RADIUS client group BLDG1RAS is configured with an IP address range 192.168.21.50–192.168.21.60, Steel-Belted Radius uses the client information for RAS1 if it receives a RADIUS request from 192.168.21.55, and it uses the client information for BLDG1RAS if it receives a RADIUS request from 192.168.21.56.
- A RADIUS client group cannot use a Class D, E, or F IP address (that is, an address greater than 223.255.255.0).

See "Adding a RADIUS Client or Client Group" for information on how to configure IPv4 address ranges for RADIUS clients.

IP Address Assignment

Steel-Belted Radius can assign IPv4 addresses to users in several ways:

- Static assignment—The same IP address is assigned to a user each time the user connects. For example, if the user Kevin has a Framed-IP-Address attribute set to 123.11.245.123, then the IP address 123.11.245.123 is assigned each time Kevin connects to the network.
- Assignment from a specific address pool—An address is assigned from a specific pool when the user connects. For example, if user Kevin has a Framed-IP-Address attribute set to the Sales IP address pool, the next available IP address from Sales is assigned when Kevin connects to the network.
- Assignment from the RADIUS client's IP address pool (or set of IP address pools)—An address is assigned from one of the pools associated with the RADIUS client that makes the connection when a user connects. For example, assume that a RADIUS client called RAS1 uses IP address pool A, and a RADIUS client called RAS2 uses IP address pool B. A User entry called Kevin has a Framed-IP-Address attribute value of pool associated with RADIUS Client. When user Kevin gets a port on RAS1, an IP address from pool A is assigned. On the next call, Kevin might connect to RAS2; in this case an address from pool B is assigned.
  Alternatively, if a user has been associated with a particular NAD-specific IP address pool (and suffix), an IP address from that pool is assigned.
- Assignment from DHCP server—An address is assigned from a DHCP server for a user-configurable period of time (DHCP lease) when a user connects. The DHCP lease period is typically significant (for

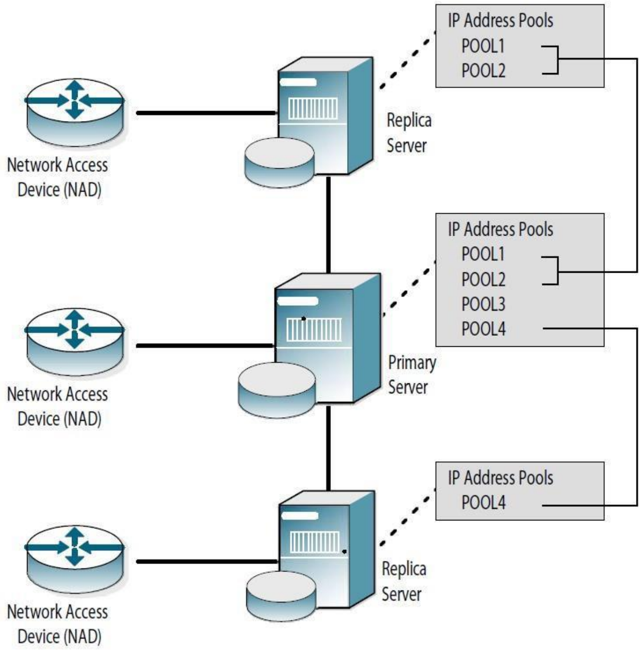
example, twenty-four hours).

Address Pools and Replication

Address pool information is not distributed with other configuration information in a replicated environment. If you are using IPv4 or IPX address pools in a replication environment, you must configure address pools separately on each replica server, and then use the same names to configure a master list of address pools on your primary server.

The master list of address pools configured on the primary server must include the names of all the pools on all of the replica servers. For example, Figure 7: IP Address Pools in a Replication Environment illustrates a simple environment that uses four address pools.
POOL1 and POOL2 are configured on one replica server and POOL4 is configured on a different replica server. As a consequence, the IP address pool list on the primary server must include POOL1, POOL2, POOL3 (the pool used by the primary server), and POOL4.

Figure 7: IP Address Pools in a Replication Environment



The network administrator must configure RADIUS clients (including the address pool associated with a RADIUS client) on the primary server. This RADIUS client/address pool association (but not the address pool information itself) is stored as part of the replication package passed from the primary server to the replica servers.

Hints

Steel-Belted Radius can treat the attribute Framed-IP-Address as a hint. This means that if this attribute appears in the Access-Request and the user return list is configured to allocate Framed-IP-Address from a pool, the IP address in the Access-Request is returned instead of the newly-allocated IP address.

This functionality is defined in the [Configuration] section of radius.ini:

```
[Configuration]

FramedIPAddressHint=<yes/no>
```

When hints are enabled, Steel-Belted Radius uses a hint to determine the value of the Framed-IP-Address attribute in the access response. This means that Framed-IP-Address in the Access-Request is returned in the Access-Accept, regardless of the Framed-IP-Address value stored in the user's account.

The default value is no.

Table 12 details the effect of hints:

Table 12: Effect of Hints

Account Configuration	Framed-IP-Address returned without hints	Framed-IP-Address returned with hints
No Framed-IP-Address	No value	Framed-IP-Address from Access-Request
Static Address	Static address	Static address
Address from Pool	Next address from pool	Framed-IP-Address from Access-Request

Note: By using hints, you can assign the same IP address to multiple active accounts.

Resource Management

This section explains how Steel-Belted Radius manages limited resources, such as network addresses, user or tunnel connections, and UDP ports.

Network Address Assignment

The Steel-Belted Radius address pooling feature allows you to set up one or more pools out of which unique network addresses are assigned dynamically as users require them. Each pool consists of a list of one or more ranges of IP addresses (an IP pool) or IPX network numbers (an IPX pool).

By using this feature, you can avoid allocating specific fixed addresses to individual users. You can make fewer addresses go farther, and you can consolidate address assignment across all your network access devices.

How Address Assignment Works

Proper operation of address assignment from a pool depends crucially on both RADIUS authentication and RADIUS accounting transactions, as follows:

1. During the RADIUS authentication transaction, if the user's attribute settings specify address assignment from a pool, an address is allocated for that user from that pool.

2. The address is reserved for that user until a RADIUS accounting transaction indicates that the user has terminated the connection.

For this reason, the network access device must be configured for RADIUS accounting, and the same Steel-Belted Radius server must be specified for both authentication and accounting. If your NAD is not configured for accounting (or does not support accounting), you cannot use the address pooling feature because addresses would be assigned but never released.

Setting Return List Attributes

The Framed-IP-Address (or Framed-IPX-Address) return list attribute controls how the user's IP (or IPX) address is assigned. The Framed-IP-Address or Framed-IPX-Address attribute can be set for each user in the Steel-Belted Radius database.

Handling Address Leaks

Under optimal conditions, Steel-Belted Radius assigns and releases addresses automatically. In some circumstances, you can get address leakage, where an address remains reserved for a user after the user has terminated the connection.

Address leakage occurs when the address has been assigned during the authentication transaction, but the accounting transaction that would have released the address is never received by Steel-Belted Radius. This can occur for several reasons:

- The Steel-Belted Radius server might have been taken down for a period of time during which accounting transactions occurred.
- The network access device might have failed or been taken down before the user terminated the connection. (In many cases, however, Steel-Belted Radius might be able to prevent address leakage by recovering the addresses when the NAD starts up again.)
- The network access device might have sent the authentication and accounting transactions to different RADIUS servers.
- Despite a successful authentication, the user's PPP negotiation with the NAD might have terminated unsuccessfully for a variety of reasons. In such a case, some network access devices might not initiate a subsequent accounting transaction.
- Routing problems might have prevented the accounting transaction from reaching Steel-Belted Radius.

An address that has "leaked" remains out of circulation until you manually release it by displaying the Sessions list and deleting the corresponding session. See "Deleting Entries from the Sessions List".

Address Leakage Upon Stopping and Starting the Server

Steel-Belted Radius maintains all current address assignments in a persistent database on disk. If you shut down the server and then restart it, all the information about which address is assigned to which user is retained. Note that if you leave Steel-Belted Radius turned off for a substantial period of time after addresses have been assigned, address leakage might occur. After you restart the server, review the Current Sessions list (described in "Displaying the Current Sessions List") and delete entries you know are obsolete.

Overlapping Address Ranges

If you maintain multiple IP or IPX address pools, you can duplicate some of the addresses among the pools. The address tracking mechanism of Steel-Belted Radius, when it is enabled, ensures that, if an IP address appears in more than one pool, after it is assigned out of any pool, it remains unavailable through any of the pools until it is released.

You must disable this type of address tracking if the server is assigning IP addresses from disjoint networks. In that configuration, two numerically identical IP addresses would signal a conflict, even though they actually belong to two different networks.

Order of Address Assignment

IP or IPX addresses are assigned on a first-in-first-out basis; that is, the address that was first released is the first to be reassigned. This ensures that addresses are out of use for as long as possible prior to reuse.

Concurrent Network Connections

The SBR Administrator program allows you to limit the number of active connections, on a per-user, or per-tunnel basis.

Concurrent User Connections

You can set a maximum limit on the number of concurrent connections that a user can have. Subsequently, when the user requests a new connection, Steel-Belted Radius compares the current number of connections to the maximum limit.

If a new connection would exceed the limit, Steel-Belted Radius can reject the additional connection or allow the connection, but log the event in the server log (described in "Using the Server Log File").

ⓘ Note: When counting connections, Steel-Belted Radius does not distinguish between multi-link connections and new user authentication attempts.

For concurrent connection limits to work, each NAD must be configured for RADIUS accounting and the same Steel-Belted Radius server must be responsible for both authentication and accounting. These conventions give the server full access to the data it needs to track connections accurately. The maximum number of concurrent connections can be set individually for any type of user by selecting the Maximum Concurrent Connections check box and entering a number in the accompanying field in the appropriate Add User/Edit User dialog. See "Administering Users" especially "Concurrent Connection Limits".

When a concurrent connection limit is set up for a user, it affects only that user. When a concurrent connection limit is set up for a group, every member of the group receives the same connection limit. For example, if GroupA has a connection limit of two, then each member of the group can have two concurrent connections.

Authentication methods that do not require user entries must provide alternate mechanisms for supporting concurrent connection limits. For example, if you are using external database authentication there is an alias mechanism you can use in the SQL or LDAP configuration file. Concurrent connection limits can be supported under proxy authentication only if the target server supports them.

ⓘ Note: Concurrent user connections can be tracked across multiple Steel-Belted Radius servers by adding the Concurrency Server package.

Concurrent Tunnel Connections

Steel-Belted Radius uses its Current Sessions list to determine the number of active connections for each tunnel. The Sessions list summarizes all of the RADIUS accounting data currently available to the server. Tunnel connections appear in the Sessions list using a special display convention that distinguishes them from user connections.

You can set a maximum limit on the number of concurrent connections that can be open using a specific tunnel. Subsequently, when a user requests a new connection through that tunnel, Steel-Belted Radius compares the current number of connections to the maximum limit. If a new connection would exceed the limit, Steel-Belted Radius rejects the additional connection.

For concurrent connection limits to work, it is essential that each NAD that can open a tunnel be configured for RADIUS accounting and that the same Steel-Belted Radius server be specified for both authentication and accounting. This permits the server's Sessions list to be kept up to date and available to every NAD that needs to authenticate tunnel connections.

Note: Concurrent tunnel connections cannot be tracked across multiple Steel-Belted Radius servers without additional software extensions. Contact Pulse Secure for more information.

Attribute Value Pooling

Attribute value pooling lets you define pools of attribute sets that are assigned dynamically when an Authorization Request is processed. The attribute sets are distributed according to specified weights and the values are returned with the Access-Accept message.

Attribute value pooling allows for a dynamic allocation of attribute values sets, so that attributes needed to configure changeable and complex situations do not have to be assigned in static profiles. This functionality is supported by the use of a vendor-specific attribute called Funk-Round-Robin-Group. The value for this attribute is a string, and should be set to the name of a .rr suffix file that defines an attribute value pool.

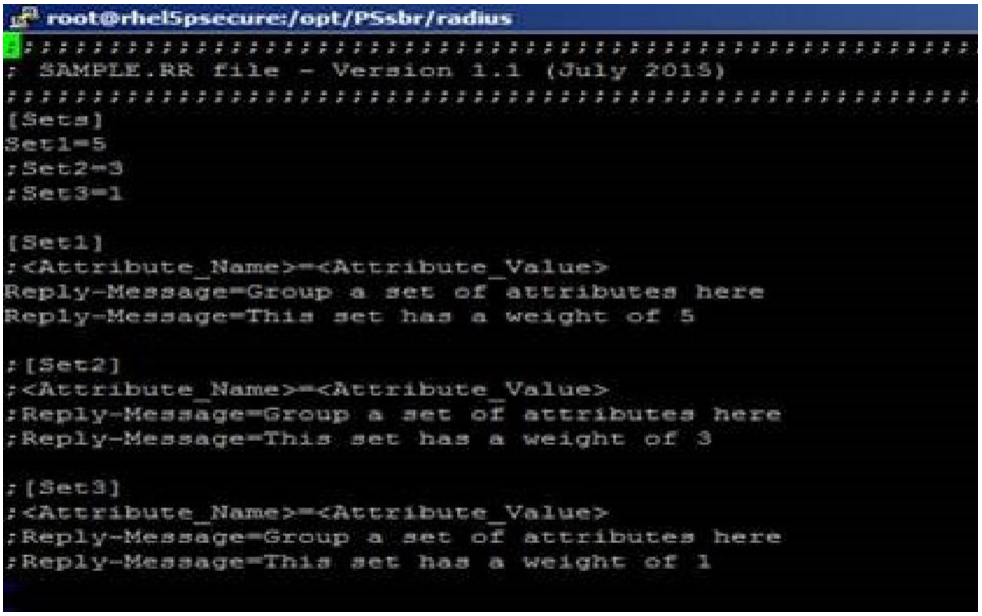
A .rr file is defined as follows:

```
[Sets]
SetName1 = Weight1
SetName2 = Weight2
M
[SetName1]
AttributeName1.1 = AttributeValue1.1
AttributeName1.2 = AttributeValue1.2
M
```

Steel-Belted Radius maintains “round-robin” statistics for each attribute value pool so that weight calculations can be performed properly. When a user who belongs to a profile that has been assigned to a particular attribute value pool logs in, the round-robin values are incremented to determine which Attribute Value set should be assigned to the user. This attribute set is added to the return list of the Access-Accept.

Attribute value pooling can be used in several ways. For example, the Acme Company wants off-site employees to be able to establish tunnels to the company network. The Acme Company maintains three tunnel connection endpoints to which end users can create VPNs into the corporate network, each of these with different capacities. The Acme Company would define an attribute value pool of three attribute sets, each describing how to establish a tunnel with one of these connection points. These attribute sets should be weighted according to the capacity of the three connection points. Figure 8: Simple .rr File illustrates a sample acme.rr file.

Figure 8: Simple .rr File



root@rhel5psecure:/opt/PSsbr/radius

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;  SAMPLE.RR file - Version 1.1 (July 2015)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
[Sets]
Set1=5
;Set2=3
;Set3=1

[Set1]
;<Attribute_Name>=<Attribute_Value>
Reply-Message=Group a set of attributes here
Reply-Message=This set has a weight of 5

;[Set2]
;<Attribute_Name>=<Attribute_Value>
;Reply-Message=Group a set of attributes here
;Reply-Message=This set has a weight of 3

;[Set3]
;<Attribute_Name>=<Attribute_Value>
;Reply-Message=Group a set of attributes here
;Reply-Message=This set has a weight of 1
```

To make this attribute value pool visible, the Acme Company would define a Funk-Round-Robin-Group VSA and assign it to the users (or the profile assigned to these users) and make the value of the VSA point to the acme.rr file shown in Figure 8: Simple .rr File.

```
Funk-Round-Robin-Group=acme.rr
```

Attribute value pools can be combined with other features. For example, by specifying an IP Pool name for a Framed-IP-Address attribute, you could load-balance IP pools.

Note: Attribute merging rules do not apply to attributes in round-robin files. You must follow appropriate attribute usage (single-valued, multi-values, checklist, etc.). No special checks are performed to ensure that the attributes and values specified in round-robin files are consistent with the rest of your system configuration. Check the dictionary file for information on attribute usage.

Attribute value pools can be reconfigured dynamically. Depending on your platform:

- Linux: Issue the HUP signal to the Steel-Belted Radius process.
- Windows: Run RADHUP.EXE from the command shell.

The modified files are re-read and the pool configuration reset appropriately.

Note: You can have only one active Round-Robin-Group attribute at any one time.

Phantom Records

When Steel-Belted Radius allocates resources such as IP addresses, IPX addresses, user connections, and tunnel connections, to its clients, it generates a phantom accounting record for its internal use. Phantom records are not written to the RADIUS accounting database, but they are displayed in the Current Session List as described in Displaying the Current Sessions List. Phantom records resemble accounting start records, except that the session ID for phantom records is displayed as N/A.

After Steel-Belted Radius receives the corresponding accounting start request packet from the client, it discards the phantom record and replaces N/A with the actual Session-ID number returned by the client device in the

Current Sessions list.

In some cases, Steel-Belted Radius can allocate a resource and create a phantom record, but then never receive a corresponding start packet from the client. To avoid committing the resource indefinitely, Steel-Belted Radius waits for a limited period for the start packet to confirm the transaction. By default, Steel-Belted Radius waits 180 seconds, though you can configure a different wait period by editing the radius.ini file (described in the Steel-Belted Radius Reference Guide).

IPv6 Support

IPv6 is the next step in the evolution of the Internet Protocol, currently implemented as Internet Protocol Version 4 (IPv4). IPv6, which has been under development for more than 10 years, provides improvements over IPv4 in addressing, configuration, and security. Although IPv6 is still an evolving standard, many operating system vendors offer production-quality IPv6 implementations for customers interested in using IPv6 networks.

IPv6 and Steel-Belted Radius

With few exceptions, Steel-Belted Radius supports IPv6 addressing wherever IPv4 addressing is supported. You can perform configuration, authentication, and accounting of RADIUS IPv6 attributes per RFC 3162, RADIUS and IPv6. The SBR Administrator (configuration application) and the LDAP configuration interface (LCI) support the configuration of RADIUS IPv6 attributes.

Note: Because many third-party libraries and software development kits (SDKs) do not support IPv6, the Steel-Belted Radius server must support local IPv4 socket connections.

IPv6 Features

Significant changes from IPv4 to IPv6 include the following:

- Expanded routing and addressing capabilities—IPv6 increases the IP address size from 32 bits to 128 bits. As a consequence, IPv6 supports more levels of addressing hierarchy, provides a much greater number of available addresses, and simplifies auto-configuration of addresses. As a consequence, address conservation techniques such as network address translation, are not necessary.
- Improved multicast routing—Multicast routing, which existed in IPv4, has been redefined and improved in IPv6. Multicast addresses now include a Scope field that limits the scope of multicasts, improving scalability. A new Anycast address type allows you to send a message to the nearest single member of a multicast group.
- Header format simplification—The IPv6 packet header format has been designed to be efficient. The IPv6 header has a fixed length of 40 bytes, divided into eight fields.
- Improved support for extensions and options—IPv6 uses extension headers, which are inserted between the IPv6 header and the transport header and packet payload, to handle special packet processing requirements. Extension headers provide a flexible means to support authentication, encryption, fragmentation, source routing, network management, and other functions. An IPv6 packet can carry any number of extension headers.
- Improved datagram sizing and fragmentation—The maximum transmission unit (MTU) describes the maximum size of a datagram that can be transmitted over a network without fragmentation. IPv6 increases the minimum MTU from 576 bytes to 1280 bytes, which makes IPv6 packets more efficient and reduces the need for packet fragmentation. Path MTU discovery enables source routers to

determine the appropriate packet size for a route.

- Quality-of-Service (QoS) functions—Packets belonging to a traffic flow that requires special handling, such as real-time video service, can be labeled by the sender. Because traffic in a particular flow can be identified in the IPv6 header, support for QoS can be implemented even when the payload of a packet is encrypted.
- Improved privacy and security—IPv6 supports extensions for authentication and data integrity to improve security and privacy of network traffic.

IPv6 Addressing

IPv6 addresses are 128 bits in length, which creates a much larger address space than 32-bit IPv4 addresses. IPv6 addresses identify individual interfaces and sets of interfaces. IPv6 addressing architecture is defined in RFC 2373, IP Version 6 Addressing Architecture.

Address Notation

In full form, IPv6 addresses are written as eight 16-bit hexadecimal blocks separated by colons:

FE80:0000:0000:0000:1232:E4BF:FE1A:8324

IPv6 addresses can be interpreted as having two variable-length fields: an IPv6 prefix and an IPv6 interface identifier.

- The IPv6 prefix varies from 0 to 128 bits in length and forms the routable network number portion of the IPv6 address. The trailing CIDR notation that may appear after human-readable IPv6 addresses (for example, /64) indicates the bit length of the IPv6 prefix.
- The IPv6 interface identifier consists of the non-prefix portion of the IPv6 address, if any, and identifies the host interface portion of the address, which identifies an IPv6 interface on the local network. The IPv6 interface identifier is typically generated automatically by the host as a function of a unique hardware identifier, such as an Ethernet MAC address. IPv6 hosts can automatically configure interface addresses by combining the IPv6 prefixes obtained from router advertisements with the IPv6 interface IDs that are determined locally.

For example:

IPv6 Prefix: FEC0:0000:0000:0000:0000:0000:0000:0000/64

IPv6 Interface ID:0260:08FF:FEFF:FFFF

IPv6 Address:FEC0:0000:0000:0000:0260:08FF:FEFF:FFFF

To simplify address notation, IPv6 accepts abbreviations in address notation. For example, leading zeros in a 16-bit block may be omitted:

FE80:0:0:0:0232:E4BF:FE1A:8324

A double colon (::) can replace a series of consecutive zeros within an address:

FE80::232:E4BF:FE1A:8324

Only one double colon can be used to compress an IPv6 address. If more than one double colon was included in an address, networking devices would not know how many zeros to insert for each double colon when expanding a compressed address to its full 128-bit representation.

In networks that support IPv4 and IPv6 nodes, IPv4 addresses can be embedded in the last four bytes of the address. An IPv4 address of 192.168.1.12 can be represented in IPv6 format as ::192.168.1.12, where :: represents a string of zeroes to pad the address to 128 bits.

Address Prefixes Like IPv4 addresses, IPv6 addresses are composed of a routable network number, known as the IPv6 prefix, and a host identifier, known as the IPv6 interface ID. IPv6 does not support address classes; IPv6 uses Classless Inter-Domain Routing with variable length network numbers, or prefixes, meaning that an IPv6 prefix is specified by supplying a bit length in conjunction with the address.

IPv6 prefixes are written as an IPv6 address, followed by a slash and the bit length of the prefix portion of the address. The prefix can be 0–128 bits in length, but the prefix is always written in terms of a 128-bit address. When writing prefixes, the trailing bits of the address comprising the interface ID are sometimes dropped so that the prefix can be abbreviated. The following prefixes are equivalent, assuming that the interface ID portion of the address may be ignored:

Canonical form: 2001:1c44:820d:eea0:0260:08ff:feff:ffff/64

Abbreviated form: 2001:1c44:820d:eea0::/64

In many cases, the interface ID portion of the address contains relevant information. A hierarchy of prefixes may reflect the assignment and reassignment of blocks of addresses to progressively smaller organizations. For example, in a typical hierarchy, the largest service providers are assigned the largest blocks of addresses and hence the shortest prefixes, called Top Level Aggregator Identifiers (TLA IDs). The large service providers reassign blocks of addresses to smaller service providers by adding a few more bits after the TLA ID; these added bits are known as Next Level Aggregator IDs. The smaller service providers again reassign smaller blocks of addresses to end-user organizations by again adding a few more bits after the NLA ID. These added bits are known as Site Level Aggregator IDs. The hierarchy continues in this way until the prefix is exhausted, leaving only the trailing bits that correspond to the non-routable IPv6 interface ID.

Steel-Belted Radius accepts all equivalent forms of IPv6 prefixes and recognizes them as being equivalent. The following prefixes are related by hierarchy, but only the canonical and abbreviated forms are equivalent:

Canonical form: 2001:1c44:820d:eea0:0260:08ff:feff:ffff/64

Abbreviated form: 2001:1c44:820d:eea0::/64

Site level prefix: 2001:1c44:820d::/48

Next level prefix: 2001:1c44:8200::/40

Top level prefix: 2001:1c00::/24

IPv6 prefixes should always be written out in full and unabbreviated form when wild cards are used, as the abbreviations become ambiguous in the presence of wild cards. The following prefixes are equivalent:

Canonical form: 2001:1c44:820d:eea0:0260:08ff:feff:ffff/64

With wild cards: 2001:1c*:??0d:eea0:*:*:*:*/64

Address Interface IDs

Because the overall size of the IPv6 address is fixed, a longer address prefix means a shorter interface ID. For example, a 48-bit prefix implies an 80-bit interface ID:

Canonical prefix: 2001:1c44:820d:eea0:0260:08ff:0000:0000/48

Abbreviated prefix: 2001:1c44:820d::/48

48-bit interface ID: eea0:0260:08ff:0000:0000

Though the interface ID portion of an IPv6 address can be 0–128 bits in length, the RADIUS standard assumes 64-bit interface IDs. Steel-Belted Radius uses a convention that all interface IDs are written as a series of four unabbreviated hexadecimal fields regardless of how they are entered:

64-bit interface ID: 0260:08ff:0000:0000

IPv6 interface IDs should always be written out in full and unabbreviated form when wild cards are used, as the abbreviations become ambiguous in the presence of wild cards. The following interface IDs are equivalent:

64-bit interface ID: 0260:08ff:0000:0000

With wild cards: 02??:08ff:*:*

ⓘ Note: The use of wild cards in IPv6 interface IDs is a Steel Belted Radius feature. Wildcards in IPv6 interface IDs are not known to be documented or prohibited by any IPv6 standards now.

IPv6 Network Numbers

In very specific cases, such as checklist processing, Steel-Belted Radius recognizes both IPv4 and IPv6 addresses as representing entire ranges of addresses. Steel-Belted Radius extends the concept of IPv4 network numbers to IPv6 as a means of representing a range of network addresses. Note that using this concept of network numbers means you cannot specify a valid network address that also happens to be a network number.

Prior to the adoption of Classless Inter-Domain Routing (CIDR), the IPv4 address space is divided into five address classes, as shown in **Table 13**.

Table 13: IPv4 Address Classes

Class	Address Range	Description
A	0.0.0.0 – 127.255.255.255	1-bit class, 7-bit network, 24-bit host
B	128.0.0.0 – 191.255.255.255	2-bit class, 14-bit network, 16-bit host
C	192.0.0.0 – 223.255.255.255	3-bit class, 21-bit network, 8-bit host
D	224.0.0.0 – 239.255.255.255	4-bit class, 28-bit multicast group
E	240.0.0.0 – 247.255.255.255	5-bit class, 27-bit reserved

Within an IPv4 address class, each network is identified by a network number that consists of the leading class bits and the network bits that follow. Network numbers are typically written as IP addresses with trailing zero bits; for example, the network number corresponding to the class C address 199.100.10.24 would typically be written as 199.100.10.0.

Each network represents a potential physical interconnection of a maximum number of hosts determined by the number of host bits. Thus, the physical network identified by the network number 199.100.10.0 might connect up to 256 hosts identified by the addresses 199.100.10.0 through 199.100.10.255 inclusive. (In practice, host addresses such as 199.100.10.0 are avoided to prevent confusion between host addresses and network numbers.) Thus, it is reasonable to interpret a network number as the entire range of addresses sharing the same network portion of the address:

Network number: 199.100.10.0 Start

of address range: 199.100.10.0

End of address range: 199.100.10.255

As a wild carded address: 199.100.10.*

To see how the concept of network numbers can be extended to IPv6 addresses, consider that IPv6 addresses can contain embedded IPv4 addresses. The IPv6 address ::ffff:199.100.10.0 can therefore be interpreted as the range ::ffff:199.100.10.0 through ::ffff:199.100.10.255 inclusive.

The IPv6 address space is not divided into classes, because IPv6 was designed with CIDR in mind. Constructing an arbitrary definition of IPv6 network numbers that both resembles IPv4 and scales well across all possible IPv6 addresses is difficult. However, since large portions of the IPv6 address space have not yet been defined and since the RADIUS specification concerns itself only with 64-bit interface IDs, we can consider arbitrarily assigning special meaning to all IPv6 addresses ending in 64 bits of zero. This represents a fraction (1/264) of the IPv6 address space, where the addresses have arbitrarily been assigned special meaning overriding their true meaning. The cases where this arbitrary definition would cause trouble are expected to be extremely rare, and it should be possible to avoid them.

Steel-Belted Radius artificially defines the concept of IPv6 network numbers as IPv6 addresses ending in 64 bits of zero, where the network number is interpreted as the entire range of IPv6 addresses sharing the same 64-bit prefix as the network number:

Network number: 2001:1c44:820d:eea0:0000:0000:0000:0000 Start

of address range: 2001:1c44:820d:eea0:0000:0000:0000:0000

End of address range: 2001:1c44:820d:eea0:ffff:ffff:ffff:ffff

As a wild carded address: 2001:1c44:820d:eea0:*:*:*:*

IPv6 Support in Steel-Belted Radius

In general, IPv6 support in Steel-Belted Radius parallels IPv4 support, both in terms of IPv6 network transport and in terms of RADIUS IPv6 attributes. When IPv6 networking is not supported by an operating system (either because the operating system cannot support IPv6 or because the IPv6 stack for the operating system has not been configured), Steel-Belted Radius recognizes IPv6 addresses and attributes, but does not use IPv6 transport mechanisms.

Socket interfaces in Steel-Belted Radius are both IPv4- and IPv6-capable. By default, IPv4 support is enabled and IPv6 support is disabled in Steel-Belted Radius. You must explicitly enable IPv6 support (by modifying settings in the [IPv6] section of the radius.ini file) before you can use IPv6 networking. Steel-Belted Radius recognizes IPv6 attributes whether or not IPv6 networking is enabled.

With few exceptions, IPv6 addresses may be configured wherever you can configure IPv4 addresses in configuration files and in the SBR Administrator.

Similarly, IPv6 RADIUS attributes can be configured wherever IPv4 RADIUS attributes can be configured. All IPv6 attributes are defined in the radius.dct file to allow inclusion in all standards-conforming dictionaries. IPv6 attributes are correctly interpreted and fully validated by the LDAP Configuration Interface (LCI) and by the SBR Administrator.

Table 14 presents a summary of IPv6 support in Steel-Belted Radius. Features marked "experimental" have not been tested and should not be deployed in a production network.

Table 14: IPv6 Feature Matrix

Feature	Supported?	Comments
Server networking	Yes	IPv6 networking must be explicitly enabled. IPv6 attributes can be processed even if IPv6 networking is not enabled. IPv6 network traffic is limited to the RADIUS authentication/accouting ports (typically 1645, 1646, 1812, and 1813). Proxying to another RADIUS server is not supported. IPv6 link local and site local addressing is deprecated.
Server DNSv6	Yes	DNSv6 must be explicitly enabled. Both IPv6 and IPv4 network connections are supported with remote DNSv6 servers. Only IPv4 network connections are supported with remote DNS servers.
Server logs	Yes	The diagnostic logging and tracing of IPv6 network connections and IPv6 attributes are fully supported. Only RFC 3162 attributes have been tested.
LCI networking	Experimental	If IPv6 networking is enabled, IPv6 addresses can be configured in the [LDAPAddresses] section of the radius.ini file.
LCI inputs	Experimental	In most cases, IPv6 values can be supplied wherever IPv4 inputs can be specified.
Attributes	Yes	Basic IPv6 attributes defined in RFC 3162 and listed in radius.dct are supported as native types or as regular text strings, as appropriate. Only RFC 3162 attributes have been tested.
Checklists	Checklists	IPv6 attributes can appear in checklists, and IPv6 address values can contain network numbers similar to IPv4 address values. IPv6 prefix values and IPv6 interface values cannot be masked or wildcarded. Only RFC 3162 attributes have been tested.
Return lists	Yes	IPv6 attributes can appear within return lists, and IPv6 values can be assigned. Only RFC 3162 attributes have been tested.
Attribute value pools	Yes	IPv6 attributes can appear in attribute value pools, and IPv6 values can be assigned to implement round-robin return list processing. Only RFC 3162 attributes have been tested.
Attribute filtering	Yes	IPv6 attributes can appear in filter rules, and IPv6 values can be assigned. Only RFC 3162 attributes have been tested.
Service type mapping	Yes	IPv6 attributes can appear in a service type mapping, and IPv6 values can be wildcarded similar to IPv4 values. Reliable string comparison of regular expressions requires all values to be expressed in canonical form. Only RFC 3162 attributes have been tested.

Feature	Supported?	Comments
Attribute mapping	Yes	IPv6 attributes can appear in an attribute mapping, and IPv6 values can be wildcarded similar to IPv4 values. Reliable string comparison of regular expressions requires all values to be expressed in canonical form. Only RFC 3162 attributes have been tested.
Class attribute	Experimental	The RADIUS Class attribute cannot contain any IPv6 attributes. You can configure IPv6 addresses in the [Hosts] section of the spi.ini file to process class attributes originating from IPv6 network connections.
DHCP	No	The use of IPv6 networking to communicate with any DHCP server is not supported. The allocation of IPv6 addresses obtained from any DHCP server is not supported.
IP address pools	No	The allocation of IPv6 addresses from an SBR-managed IP address pool is not supported. However, RFC 3162 provides an attribute, Framed-IPv6-Pool, that allows the RAS to implement an IPv6 address pool.
Networking for authentication	Yes	IPv6 addresses can be configured in the [Addresses] section of the radius.ini file. IPv6 network traffic is limited to the RADIUS authentication/accounting ports (typically 1645, 1646, 1812, and 1813). Proxying to another RADIUS server is not supported. IPv6 link local and site local addressing is deprecated.
Authentication logs	Yes	IPv6 attributes are fully supported. Only RFC 3162 attributes have been tested.
Local User authentication	Yes	IPv6 attributes are fully supported. Only RFC 3162 attributes have been tested.
Authenticate-Only requests	Yes	IPv6 attributes are fully supported. Only RFC 3162 attributes have been tested.
Pass-through authentication	Partial	IPv6 attributes are fully supported. However, because many third-party libraries do not support IPv6, IPv6 networking is notnecessarily supported with external services such as RSA SecurID and TACACS+. Only RFC3162 attributes have been tested. Third-party software may not support IPv6 networking or attributes.
External authentication (for example, LDAP or SQL)	Partial	IPv6 attributes are fully supported. However, because many third-party libraries do not support IPv6, IPv6 networking is not necessarily supported with external services such as SQL. Only RFC 3162 attributes have been tested. Third-party software may not support IPv6 networking or attributes.
EAP-TTLS authentication	Yes	IPv6 attributes are fully supported. Only RFC 3162 attributes have been tested.

Feature	Supported?	Comments
Directed authentication	Yes	IPv6 attributes are fully supported. Only RFC 3162 attributes have been tested.
Networking for accounting	Experimental	IPv6 addresses can be configured in the [Addresses] section of the radius.ini file.
Accounting logs	Yes	IPv6 attributes are fully supported. Only RFC 3162 attributes have been tested.
External accounting (for example, SQL)	Partial	IPv6 attributes are fully supported. However, because many third-party libraries do not support IPv6, IPv6 networking is not necessarily supported with external services such as LDAP and SQL. Only RFC 3162 attributes have been tested. Third-party software may not support IPv6 networking or attributes.
Directed accounting	Yes	IPv6 attributes are fully supported. Only RFC 3162 attributes have been tested.
Spooled accounting	Yes	IPv6 attributes are fully supported. Only RFC 3162 attributes have been tested.
Networking for proxy	Experimental	IPv6 addresses can be configured in the [Interfaces] section of the proxy.ini file. Both classic and extended proxy support IPv6.
Proxy authentication	Experimental	IPv6 attributes are fully supported. Both classic and extended proxy support IPv6.
Proxy accounting	Experimental	IPv6 attributes are fully supported. Both classic and extended proxy support IPv6.
3GPP2	Experimental	RFC standards are not sufficiently evolved to enable full support of IPv6. Although IPv6 attributes can be processed, they are not meaningful in the context of 3GPP2.
Master SNMP agent	No	(Linux only) The use of IPv6 networking to communicate with an IPv6 capable SNMP management station and/or SNMP subagent is not supported.
SNMP subagent	No	(Linux only) IPv6 networking, IPv6 trap variables, and IPv6 MIB objects are not supported. IPv6 addresses are reported as IPv4 MIB objects possessing the value 255.255.255.255.
Windows events	No	(Windows only) Neither IPv6 networking nor IPv6 event variables are supported at this time.

Feature	Supported?	Comments
Networking for plug-Ins	Experimental	Steel-Belted Radius does not control the networking of back end servers with its plug-ins. IPv6 networking is generally a hidden detail of third-party back end server configuration.
Plug-In APIs	Experimental	IPv6 features and parameters are exposed in the new plug-in APIs. The older plug-in APIs are deprecated but still functional. You should upgrade to the new plug-in APIs to gain access to IPv6 features. IPv6 APIs can be invoked even if IPv6 networking is not enabled.
Plug-In attributes	Experimental	Basic IPv6 attributes defined in RFC-3162 and listed in radius. dct are supported as native types or as regular text strings, as appropriate.
ODBC plug-Ins	Experimental	(Windows/Linux only) IPv6 attributes are fully supported, but the required third-party software may not support IPv6.
JDBC plug-Ins	Experimental	(Windows/Linux only) IPv6 attributes are fully supported, but the required third-party software may not support IPv6.
LDAP plug-In	Experimental	IPv6 attributes are fully supported, but the required third-party software may not support IPv6.
RSA SecurID plug-Ins	Experimental	IPv6 attributes are fully supported, but the required third-party software may not support IPv6.
PEAP Plug-In	Experimental	IPv6 attributes are fully supported, but the required third party software either does not currently support IPv6 or we have not tested it.
TLS plug-In	Experimental	IPv6 attributes are fully supported, but the required third-party software may not support IPv6.
TTLS plug-In	Experimental	IPv6 attributes are fully supported, but the required third-party software may not support IPv6.
PAS plug-Ins	Experimental	(SPE only) IPv6 attributes are fully supported.
Concurrency plug-Ins	Experimental	(SPE only) IPv6 attributes are fully supported.
Uniport Plug-In	Experimental	IPv6 attributes are fully supported.
3COM CCA Tunnels (deprecated)	Experimental	The use of IPv6 networking is not supported. IPv6 attributes can be processed even if IPv6 networking is not enabled.

Feature	Supported?	Comments

RADIUS IPv6 Attributes

All RADIUS attributes defined in RFC 3162, RADIUS and IPv6, are supported in Steel-Belted Radius as native types or as regular text strings. All forms of attribute processing, such checklist processing, return list processing, attribute echoing/deleting/merging, are supported. However, IPv6 prefix values and IPv6 interface values cannot be masked or wildcarded in checklist processing.

Third-party plug-ins that have not been upgraded to support IPv6 should be able to process IPv6 attributes as opaque hexadecimal strings.

Table 15 lists the attribute number, and the number of times an attribute can appear in an Access-Request, Access-Accept, Access-Reject, Access-Challenge, and Accounting-Request packets for each type of IPv6 RADIUS attribute.

Table 15: IPv6-Specific RADIUS Attributes

Attribute	Attr Num	Acc-Req	Acc-Accept	Acc-Rej	Acc-Chall	Acct- Req
NAS-IPv6-Address	95	0-1	0	0	0	0-1
Framed-Interface-Id	96	0-1	0-1	0	0	0-1
Framed-IPv6-Prefix	97	0+	0+	0	0	0+
Login-IPv6-Host	98	0+	0+	0	0	0+
Framed-IPv6-Route	99	0	0+	0	0	0+
Framed-IPv6-Pool	100	0	0-1	0	0	0-1

NAS-IPv6-Address

The NAS-IPv6-Address attribute is similar in function to the NAS-IP-Address attribute. Either attribute is sufficient to identify the IP address of the requesting RAS to the RADIUS server. If both attributes appear in the same RADIUS Access-Request packet, Steel-Belted Radius processes the NAS-IPv6-Address attribute for the purpose of identifying the RAS.

The NAS-IPv6-Address attribute may be specified by the RAS in access and accounting request packets. Zero or one NAS-IPv6-Address attributes may be specified. If present, the fixed length NAS-IPv6-Address attribute contains the complete 128-bit IPv6 address of the requesting RAS. Steel-Belted Radius allows zero or one 128-bit IPv6 address to be specified for each RAS.

The server authentication logic validates these addresses on extraction from the database and compares them with NAS-IPv6-Address attributes when they are received in access and accounting request packets. The server accounting logic writes these addresses to the accounting logs in a human readable format.

Example

Human readable:fe80::260:8ff:feff:ffff

RADIUS attribute:5f 12 fe 80 00 00 00 00 00 00 02 60 08 ff fe ff ff ff

Framed-Interface-Id

The Framed-Interface-Id attribute specifies the IPv6 interface ID to be assigned to a user. When combined with a Framed-IPv6-Prefix attribute, a single Framed-Interface-Id attribute forms one or more complete IPv6 addresses to be assigned to the user.

In general, the user is assigned the number of addresses equal to the number of Framed-IPv6-Prefix attributes, where the addresses have the same Framed-Interface-Id value and different Framed-IPv6-Prefix values.

It is possible to assign complete IPv6 addresses using only Framed-IPv6-Prefix attributes (i.e. without specifying any Framed-Interface-Id attribute). For example, in the case of PPP, it can be quite difficult to automatically generate a unique IPv6 interface ID for a given network segment, so it is recommended that the RADIUS server honor the hint if this attribute is suggested by the RAS. This is typically accomplished with echo attributes.

The Framed-Interface-Id attribute may be specified by the RAS in Access- and Accounting-Request packets, and by the RADIUS server in Access-Accept packets. Zero or one Frame-Interface-Id attributes may be specified. If present, the fixed length Framed-Interface-Id attribute contains the 64-bit interface ID to be assigned to the user.

Steel-Belted Radius allows zero or one 64-bit interface ID to be specified for each local user. The server authentication logic validates this interface ID on extraction from the database and includes the Framed-Interface-Id attribute in Access-Accept packets if none is received in the Access-Request packets.

Example

Human readable:260:8ff:feff:ffff

RADIUS attribute:60 0a 02 60 08 ff fe ff ff ff

Framed-IPv6-Prefix

The Framed-IPv6-Prefix attribute specifies the IPv6 networks to be assigned to a user. When combined with a Framed-Interface-ID attribute, a single Framed-IPv6-Prefix attribute forms one or more complete IPv6 addresses to be assigned to the user.

In general, the user is assigned the number of addresses equal to the number of Framed-IPv6-Prefix attributes, where the addresses have the same Framed-Interface-Id value, but different Framed-IPv6-Prefix values.

It is possible to assign complete IPv6 addresses using only Framed-IPv6 Prefix attributes (that is, without specifying any Framed-Interface-Id attribute). For example, the Framed-IPv6-Prefix attributes may be suggested by the RAS and overridden by the RADIUS server. In any case, the RAS is expected to be able to plumb the routes implied by the Framed-IPv6-Prefix attributes and these need not be repeated in separate Framed-IPv6-Route attributes.

The Framed-IPv6-Prefix attribute may be specified by the RAS in Access- and Accounting-Request packets, and by the RADIUS server in Access-Accept packets. Zero or more Framed-IPv6-Prefix attributes may be specified. If present, the variable-length Framed-IPv6-Prefix attribute contains an IPv6 prefix from 0 to 128 bits in length. Extra bits beyond the actual prefix length must be set to 0.

Steel-Belted Radius allows zero or more variable-length IPv6 prefixes to be specified for each local user or attribute profile. The server authentication logic validates these prefixes on extraction from the database and includes the proper number of Framed-IPv6-Prefix attributes in Access-Accept packets if none are received in the access request packets. The server accounting logic writes these prefixes to the accounting logs in a human readable format.

Example

Human readable:fe80::260:8ff:feff:ffff/10

RADIUS attribute:61 14 00 0a fe 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00

(8-bit type) (8-bit length)

(8-bit zero) (8-bit prefix length) (128-bit IPv6 prefix)

Login-IPv6-Host

The Login-IPv6-Host attributes specify the IPv6 addresses of the systems with which the user is connected when the Login-Service attribute is also included. The Login-IPv6-Host attribute may be suggested by the RAS and overridden by the RADIUS server.

The Login-IPv6-Host attribute may be specified by the RAS in access and accounting request packets, and by the RADIUS server in Access-Accept packets. Zero or more Login-IPv6-Host attributes may be specified. If present, the fixed length Login-IPv6-Host attribute contains the complete 128-bit IPv6 address of the login host, or a special value:

- 128-bits set to 0 indicates that the RAS should select the login host for the user.
- 128-bits set to 1 indicates that the RAS should allow the user to select the login host.
- Other values indicate the actual 128-bit IPv6 address of the login host.

Steel-Belted Radius allows zero or more 128-bit IPv6 addresses (including special values) to be specified for each local user or attribute profile. The server authentication logic validates these addresses (including special values) on extraction from the database and includes the proper number of Login-IPv6-Host attributes in Access-Accept packets if none are received in the access request packets. The server accounting logic writes these addresses to the accounting logs in a human readable format.

Example

Human readable:fe80::260:8ff:feff:ffff

RADIUS attribute:62 12 fe 80 00 00 00 00 00 00 02 60 08 ff fe ff ff ff

Framed-IPv6-Pool

The Framed-IPv6-Pool attribute specifies the name of a RAS managed pool (as opposed to a RADIUS server managed pool) from which the RAS should assign an IPv6 prefix to the user. The Framed-IPv6-Pool attribute may not be suggested by the RAS and is always determined by the RADIUS server.

The Framed-IPv6-Pool attribute may be specified by the RAS in Accounting-Request packets, and by the RADIUS server in Access-Accept packets. Zero or one Framed-IPv6-Pool attributes may be specified. If present, the variable-length Framed-IPv6-Pool attribute contains the name of a RAS managed pool in human readable text. The text is not NULL terminated.

Steel-Belted Radius allows zero or one variable length pool name to be specified for each local user. The server authentication logic validates the pool name on extraction from the database and includes the proper number of

Framed-IPv6-Pool attributes in Access-Accept packets. The server accounting logic writes these pool names to the accounting logs in a human readable format.

Example

Human readable:ipv6-pool

RADIUS attribute:64 0b 69 70 76 36 2d 70 6f 6f 6c

Framed-IPv6-Route

The Framed-IPv6-Route attribute specifies the IPv6 routing information to be configured for the user on the RAS. The RAS is expected to be able to plumb the routes specified by the Framed-IPv6-Route attributes in addition to those that may already be implied by separate Framed-IPv6-Prefix attributes. The Framed-IPv6-Route attribute may not be suggested by the RAS and is always determined by the RADIUS server.

The Framed-IPv6-Route attribute may be specified by the RAS in accounting request packets, and by the RADIUS server in Access-Accept packets. Zero or more Framed-IPv6-Route attributes may be specified. If present, the variable-length Framed-IPv6-Route attribute contains IPv6 routing information in human readable text. The text is not NULL terminated. The format of the text (destination prefix, gateway address, metrics) is described in RFC-3162.

Steel-Belted Radius allows zero or more variable-length IPv6 routes to be specified for each local user or attribute profile. The server authentication logic validates these routes on extraction from the database and includes the proper number of Framed-IPv6-Route attributes in Access-Accept packets. The server accounting logic writes these routes to the accounting logs in a human readable format.

Example

Human readable:2000:0:0:106::/64 2000::106:a00:20ff:fe99:a998 1

RADIUS attribute:63 32 32 30 30 30 3a 30 3a 30 ... 39 3a 61 39 39 38 20 31

Enabling IPv6 Networking

To enable IPv6 networking in Steel-Belted Radius, you must modify the radius.ini file and then restart your Steel-Belted Radius server. For information on the settings in the radius.ini file, refer to the Steel-Belted Radius Reference Guide.

Note that Steel-Belted Radius can process IPv6 attributes even if IPv6 networking is not enabled, provided that the IPv6 attributes are described in the RADIUS dictionary files.

Configuring IPv6 Scope IDs

Some types of IPv6 addresses requires an IPv6 scope ID to avoid address ambiguity. In some cases, the Steel-Belted Radius server can select a scope ID automatically.

The [IPv6] section of the radius.ini file can specify how scope IDs are selected for each IPv6 address type that is recognized by the server. If the parameter value is 0, the Steel-Belted Radius server selects a scope ID automatically. If the parameter value is not 0, then the Steel-Belted Radius server uses that value as the scope ID when establishing network connections involving that IPv6 address type.

Note: You can use the output of the **ifconfig -a/ip a** shell command on Linux platforms, and the output of the ipconfig /all shell command on Windows platforms (ipv6 if on Windows XP platforms) to determine the proper host specific scope ID for an address type. The scope ID is identical to the interface index on which the address type is supported and on which the desired destinations are reachable. On Linux platforms, the server accepts traditional interface names, such as hme0, instead of numeric scope IDs.

Configuring IPv6 Addresses for RADIUS Client Connections

You can configure the [Addresses] section of the radius.ini file if you want to specify the local address or addresses on which Steel-Belted Radius listens for RADIUS client connections. By default, Steel-Belted Radius automatically discovers and configures all available IPv4 interfaces on the local host. If IPv6 is enabled, Steel-Belted Radius discovers and configures both IPv4 and IPv6 interfaces.

You can configure Steel-Belted Radius to configure IPv4 automatically by entering **AutoConfigureIPv4** in the [Addresses] section. Similarly, you can configure Steel-Belted Radius to configure IPv6 automatically by entering **AutoConfigureIPv6** in the [Addresses] section. If you configure specific IPv4 or IPv6 addresses, Steel-Belted Radius listens for RADIUS traffic on only those interfaces.

The IPv6 unspecified address :: allows connections on any IPv6 address or IPv4 address, with IPv4 connections represented as IPv6 IPv4 mapped addresses. Because IPv6 IPv4 mapped addresses are not currently supported by the Windows IPv6 protocol stack, you must enter the IPv4 unspecified address 0.0.0.0 with the IPv6 unspecified address :: to approximate the desired behavior on Windows platforms.

Configuring DNSv6 Support

Enabling Domain Name Service Version 6 (DNSv6) support lets Steel-Belted Radius communicate with a DNSv6 server to resolve host names. By default, Steel-Belted Radius uses DNS unless IPv6 is enabled and DNSv6 support is configured by means of the DynamicNameResolution parameter in the [IPv6] section of the radius.ini file.

- If DynamicNameResolution is set to 0, Steel-Belted Radius uses IPv4 DNS, which means it does not query DNSv6 services.
- If DynamicNameResolution is set to 1, Steel-Belted Radius uses IPv6 DNS (DNSv6), which means it does not query IPv4 DNS services and ignores IPv4-specific information returned by DNSv6 services.
- If DynamicNameResolution is set to 2, Steel-Belted Radius queries DNSv6 services for IPv6-specific information, and then queries IPv4 DNS services for IPv4 specific information if DNSv6 fails to resolve a host name.

Chapter 3
Using SBR Administrator

This chapter presents an overview of how to use the SBR Administrator, which is a Java-based application that lets you configure settings for Steel-Belted Radius. In minutes, you can set up new users, alter standard profiles, or configure new network access devices from any computer on the network.

The Chapter 4 "Using Web Graphic User Interface (GUI)" describes an overview of how to use the Web graphical user interface (GUI) to configure the Steel Belted Radius server. The legacy SBR Administrator is included in the SBR Enterprise 6.2.5 release to ease transition to the WebGUI and may not be included in further releases.

Note: Administrators making large-scale changes to the Steel-Belted Radius database might prefer to use the LDAP command line interface. See "LDAP Configuration Interface".

Launching SBR Admin GUI

The SBR Admin GUI can be launched using the Internet Explorer version 8.0 and above or the latest version of Mozilla Firefox.

Before you launch, prepare the system by:

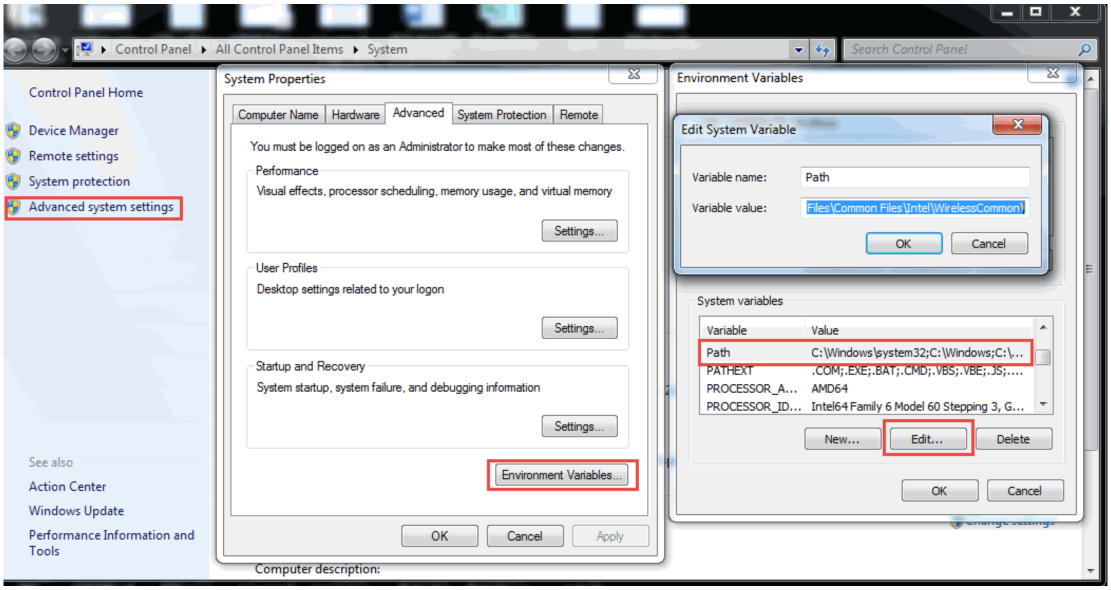
1. Installing Java for SBR Admin
2. Enabling Java Runtime Environment on Internet Explorer Browser (ActiveX deprecated)
3. Enabling Java on Mozilla Firefox Browser for Java Applet

You can launch Admin GUI of SBR 6.1.7 and SBR 6.2/above on the same machine. For detailed procedure, see Launching Admin GUI of SBR 6.1.7 and SBR 6.2/above on the same Machine.

Installing Java for SBR Admin

SBR Admin supports any 32 bit version of JRE (Java 7_u79 or above). For improved security, it is recommended to install Java 8.

Figure 9: Setting Java Environmental Variables



You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your logon

System startup, system failure, and debugging information

Variable name: Path

Variable value: Files\Common Files\Intel\WirelessCommon\

Variable	Value
Path	C:\Windows\system32;C:\Windows;C:\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	AMD64
PROCESSOR_ID...	Intel64 Family 6 Model 60 Stepping 3, G...

Windows maintains a registry which holds Java PATH and updates it when Java is installed. If for any reason the Java path is not set, then set the bin path of JRE to System Environmental Variable "PATH".

If there are more than one version of JRE installed in the system, then set the Java path that should be used for SBR GUI in the System Environmental variables as the first one among all installed Java versions.

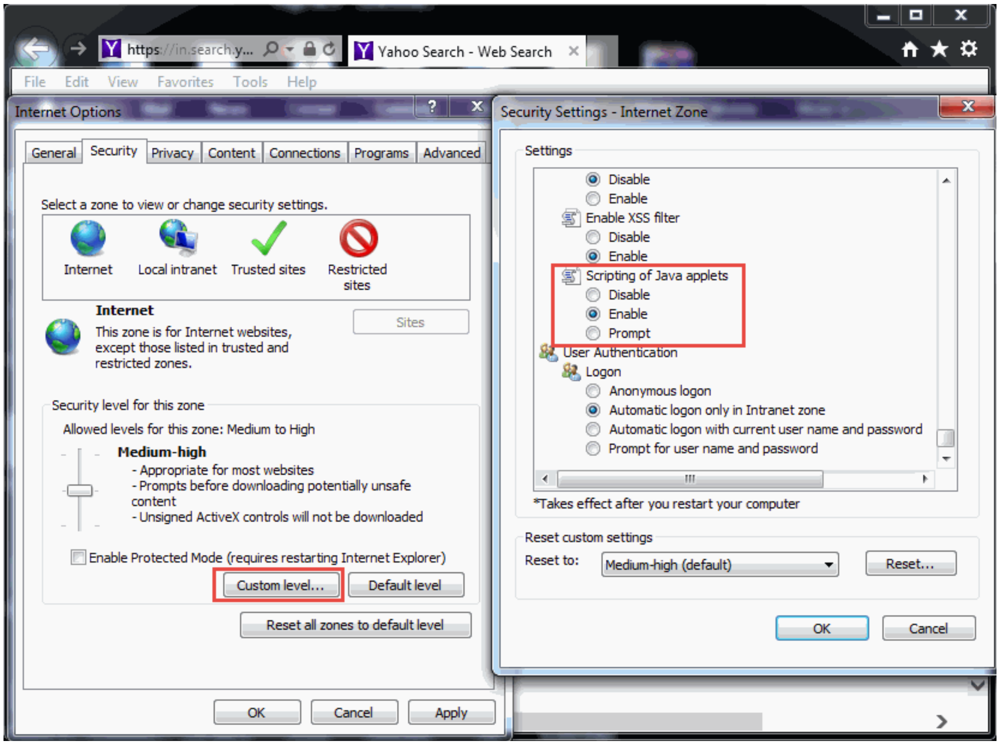
Enabling Java Runtime Environment on Internet Explorer Browser (ActiveX deprecated)

Internet Explorer browser versions 8 and above does not support ActiveX. Though ActiveX is deprecated on IE8, it is not disabled on Windows 7 system. So ActiveX would work for IE8 on Windows7 and older operating systems.

To enable Java on IE8 and above on Windows 8 and later:

1. Open Internet Explorer.
2. On the Tools menu, select Internet Options.
3. In the Internet Options window, select the Security tab, and click the Custom Level button.
4. In the Security Settings window, scroll down to the Scripting of Java applets setting.

Figure 10: Internet Options in IE 8



Internet Options – Security

Select a zone to view or change security settings.

Internet

This zone is for Internet websites, except those listed in trusted and restricted zones.

Security level for this zone

Allowed levels for this zone: Medium to High

Medium-high

- Appropriate for most websites
- Prompts before downloading potentially unsafe content
- Unsigned ActiveX controls will not be downloaded

□ Enable Protected Mode (requires restarting Internet Explorer)

Security Settings - Internet Zone

● Disable
○ Enable

Enable XSS filter
○ Disable
● Enable

Scripting of Java applets
○ Disable
● Enable
○ Prompt

User Authentication
Logon
○ Anonymous logon
● Automatic logon only in Intranet zone
○ Automatic logon with current user name and password
○ Prompt for user name and password

*Takes effect after you restart your computer

Reset custom settings

Reset to: Medium-high (default)

5. Make sure the Enable radio button is selected.
6. Click OK to save the setting.

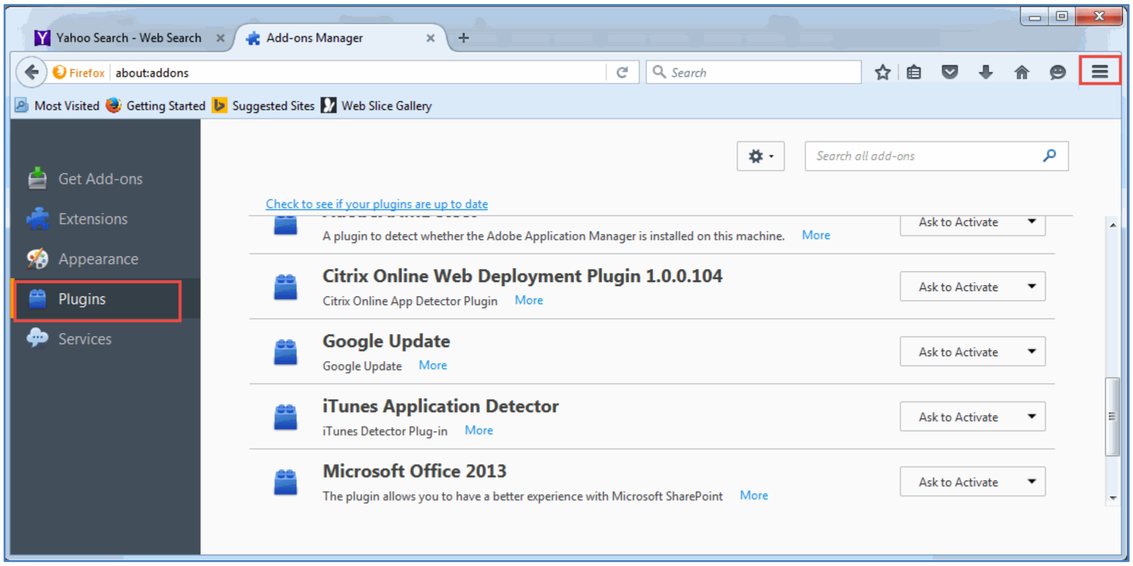
Enabling Java on Mozilla Firefox Browser for Java Applet

Ensure you have installed the latest version of Mozilla Firefox.

1. Install JRE 32 bit (64 bit is not supported).

2. Verify if JRE plugin is installed and enabled for Firefox by selecting Add-ons -> Plugins.

Figure 11: Plugins in Firefox



A plugin to detect whether the Adobe Application Manager is installed on this machine. More

Citrix Online Web Deployment Plugin 1.0.0.104
Citrix Online App Detector Plugin More

Google Update
Google Update More

iTunes Application Detector
iTunes Detector Plug-in More

Microsoft Office 2013
The plugin allows you to have a better experience with Microsoft SharePoint More

3. Select Ask to Activate or Always Activate if it is not enabled.

Launching Admin GUI of SBR 6.1.7 and SBR 6.2/above on the same Machine

Deployment of SBR admin GUI has been changed in SBR 6.2 for supporting latest version of java so that the UI is more interactive and visible.

The following issues are observed when both versions of admin GUI are launched on the same machine:

- Conflict between deployer of SBR admin 6.1.7 and SBR 6.2 on Internet Explorer.
- SBR admin 6.1.7 did not launch on Firefox.

For the first issue on IE, see the work around in section Clearing Cache Directories while Launching New GUI.

For the second issue on Firefox, back merge the fix from SBR 6.2 to SBR 6.1.7.

To launch SBR 1.7 and 6.2 Admin GUI on the same machine, it is also recommended to use IE browser for SBR 6.1.7 and Firefox for SBR 6.2.

Clearing Cache Directories while Launching New GUI

Due to the changes to the deployment files in SBR 6.2, the older files present in the system will lead to conflict of files.

To clear the cache:

1. Close Internet Explorer.
2. Delete following files for ActiveX deployment:

- C:\Windows\Downloaded Program Files\deploy.dll
- C:\Windows\Downloaded Program Files\deploy.inf
- C:\Windows\Downloaded Program Files\deployer.exe

Running the SBR Administrator

You start the SBR Administrator by running a browser and opening a connection to the Steel-Belted Radius server you want to configure.

To log into a Steel-Belted Radius server:

1. Open a browser connection to the Steel-Belted Radius server you want to administer.

   - To administer a Steel-Belted Radius server running on your local host, enter http://localhost:port/, where port is the TCP port on which the server is listening for administration connections. For example, to open a connection on a local host listening on port 1812, use the following URL:

     http://localhost:1812/

   - To administer a Steel-Belted Radius server running on a remote host, enter http://server:port/, where server is the DNS name or IPv4/IPv6 address of the server, and port is the TCP port on which the server is listening for administration connections. For example, to open a connection on a remote host at IP address 192.168.24.15 listening on port 1812, use the following URL, where ipaddress is the IP address or DNS name of the remote server:

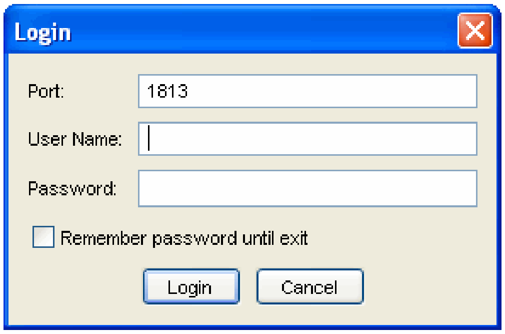
     http://192.168.24.15:1812/

   When the Steel-Belted Radius Administrator page opens, click the Launch link to download and start the SBR Administrator.

2. When the Login dialog (**Figure 12**) opens, specify the TCP port you want to use for communication between SBR Administrator and the Steel-Belted Radius server.

   The value you enter in the Port field must match the value specified in the SecureTcpAdminPort parameter in the [Ports] section of the radius.ini file on the Steel-Belted Radius server. The default value is 1813.

Figure 12: Login Dialog



Port: 1813

User Name:

Password:

□ Remember password until exit

3. Enter your administrator username in the User Name field.

4. Enter your login password in the Password field.

5. Click Login.

When you click Login, SBR Administrator establishes an HTTPS connection with the local or remote server. If it cannot establish a connection in 10 seconds, SBR Administrator times out and displays an error message.

Note: If a timeout occurs, verify that the Steel-Belted Radius service/daemon is running on the target server and that it is listening on the administration port you are specifying in the Login dialog.

SBR Administrator verifies that the username you entered is present in the access.ini file. If the username is found, SBR Administrator validates the password you entered against a local or remote password database.

When you connect to a server, the Status panel lists various features of the running server, such as version, platform on which it is running, IP address, available authentication methods, license information, and any initialization errors that might have occurred.

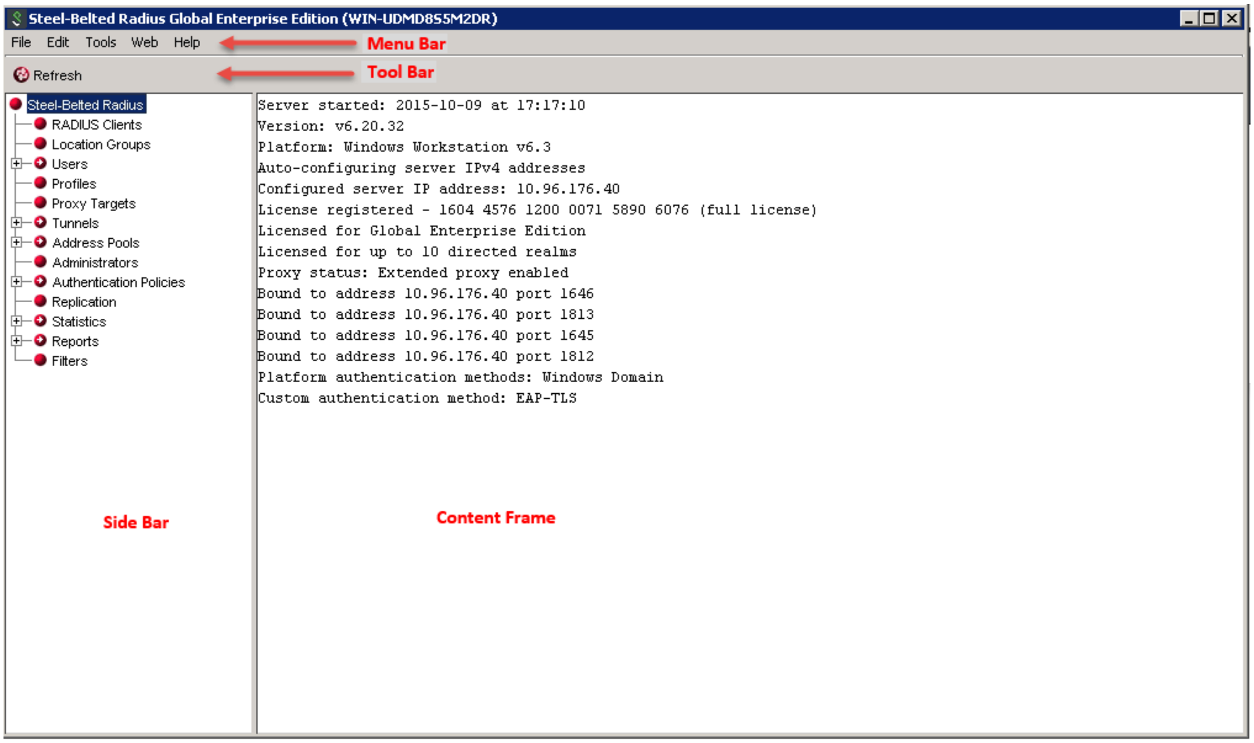
Navigating in SBR Administrator

This section describes how to use the SBR Administrator panels, menus, dialogs, and toolbar.

SBR Administrator Panels

SBR Administrator uses a series of framed windows (panels) to configure server settings and display statistics and logs in SBR Administrator. Figure 13: SBR Administrator Panel Layout illustrates the components of an SBR Administrator panel. To display a panel, you click an entry in the Sidebar. SBR Administrator displays the specified panel in the Content frame.

Figure 13: SBR Administrator Panel Layout



```
Server started: 2015-10-09 at 17:17:10
Version: v6.20.32
Platform: Windows Workstation v6.3
Auto-configuring server IPv4 addresses
Configured server IP address: 10.96.176.40
License registered - 1604 4576 1200 0071 5890 6076 (full license)
Licensed for Global Enterprise Edition
Licensed for up to 10 directed realms
Proxy status: Extended proxy enabled
Bound to address 10.96.176.40 port 1646
Bound to address 10.96.176.40 port 1813
Bound to address 10.96.176.40 port 1645
Bound to address 10.96.176.40 port 1812
Platform authentication methods: Windows Domain
Custom authentication method: EAP-TLS
```

SBR Administrator Sidebar

You use the entries in the sidebar in the SBR Administrator to select which panel you want to display; for example, to display the **RADIUS Clients** panel, you click Radius Clients in the sidebar.

Some entries in the sidebar are expandable. For example, when you click the **Users** entry in the sidebar, the entry displays subentries such as **Local** and **Domain**.

SBR Administrator Menus

The SBR Administrator has four menus: File, Edit, Web, and Help.

File Menu

Table 16 describes the functions of each entry in the File menu in the SBR Administrator.

Table 16: File Menu Options

Menu Entry	Function
License	Opens the Add a License for Server dialog, which lets you add a license string for your Steel-Belted Server software. For more information, see "**Adding License Keys**".
Import	Opens the Import dialog, which lets you import information from an XML file into the Steel-Belted Radius database. The Import dialog is described in Appendix E, "Importing and Exporting Data."
Export	Opens the Export dialog, which lets you export selected information from the Steel-Belted Radius to an XML file. The Export dialog is described in Appendix E, "Importing and Exporting Data."
Backup/Restore	Opens the Backup/Restore dialog, which lets you back up the Steel-Belted Radius database to an archive file or restore Steel-Belted Radius from an archive file. The Backup/Restore dialog is described in xref.
Login	Opens the Login dialog, which lets you log in to a Steel-Belted Radius server. The Login dialog is described in "**Running the SBR Administrator**". The Login menu entry is dimmed if you are logged into a Steel-Belted Radius server.
Logout	Terminates your connection to a Steel-Belted Radius server. The Logout menu entry is dimmed if you are not logged into a Steel-Belted Radius server.
Exit	Exits the Steel-Belted Radius application.

Edit Menu

Table 17 describes the functions of each entry in the Edit menu in the SBR Administrator.

Table 17: Edit Menu Options

Menu Entry	Function
Cut	Deletes an existing object from the Steel-Belted Radius database and copies its information to the Clipboard. Active only when an object is selected in an SBR Admin panel.
Copy	Copies the selected object from the Steel-Belted Radius database to the Clipboard. Active only when an object is selected in an SBR Admin panel.
Paste	Pastes an object from the Clipboard to the Steel-Belted Radius database. Active only after a Cut or Copy command has been used.
Delete	Deletes an object from the Steel-Belted Radius database.

Tools Menu

Table 18 describes the function of the entry in the Tools menu in the SBR Administrator.

Table 18: Tools Menu Options

Menu Entry	Function
Options	Opens the Options dialog, which lets you specify whether you want Steel-Belted Radius to refresh configuration data automatically.

Web Menu

Table 19 describes the functions of each entry in the Web menu in the SBR Administrator.

Table 19: Web Menu Options

Menu Entry	Function
Steel-Belted Radius Home Page	Opens the support page for Steel-Belted Radius in a browser window. This page lets you review product information, download documentation and technical notes, and access other resources.
Pulse Secure Home Page	Opens the home page for Pulse Secure (**https://www.pulsesecure.net/**) in a browser window.

Help Menu

Table 20 describes the functions of each entry in the Help menu in the SBR Administrator.

Table 20: Help Menu Options

Menu Entry	Function
Contents	Opens the online help for the SBR Administrator
About	Displays the About SBR Administrator dialog, which lists version information for the SBR Administrator. For more information, see "**Displaying Version Information**".

SBR Administrator Toolbar

After you log into Steel-Belted Radius, you can use the toolbar to manipulate SBR Administrator objects, such as users or RADIUS clients. The buttons on the SBR Administrator toolbar change when you change panels to provide buttons appropriate for your current context.

Figure 14: SBR Administration Toolbar

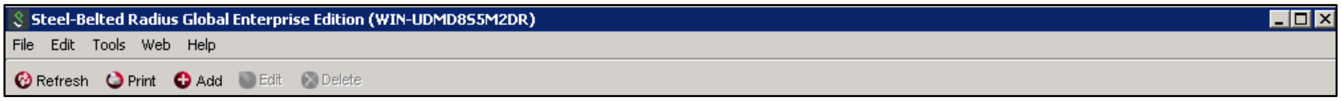

Steel-Belted Radius Global Enterprise Edition (WIN-UDMD8S5M2DR)
File
Edit
Tools
Web
Help
Refresh
Print
Add
Edit
Delete

Table 21: SBR Administrator Toolbar

Toolbar Button	Function
Refresh	Refreshes the displayed list of items in the SBR Administrator dialog.

Toolbar Button	Function
Print	Prints the contents of the active panel.
Add	Adds an object to the Steel-Belted Radius database.
Edit	Edits an existing object in the Steel-Belted Radius database. Active only when an object is selected in the active panel.
Delete	Deletes an existing object from the Steel-Belted Radius database.
Apply	In the Order of Methods dialog, applies any changes you have made to the authentication policy settings.
Reset	Restores the default values for controls in the active panel.
Clear (System Statistics only)	Resets statistics other than Server Up Time to zero.
Search (Filters only)	Opens the Search Filters window, which is described in "**Searching the Filter List**".
EAP Setup (Order of Methods panel only)	Opens the EAP Setup dialog, which lets you specify the active EAP methods that will be used for an authentication method. For more information, see "**Configuring EAP Settings**".

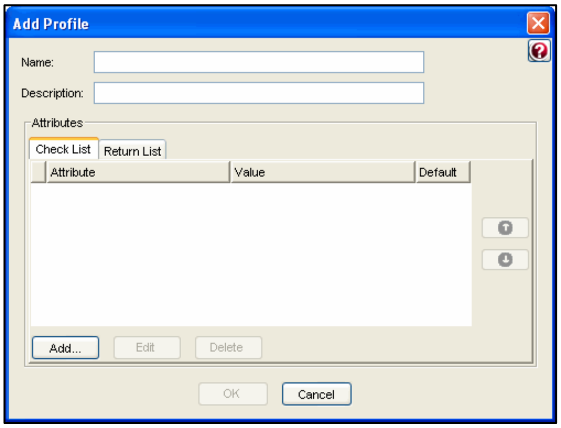
SBR Administrator Dialogs

This section summarizes how to use SBR Administrator dialogs and controls.

Adding an Entry

To add an entry to the Steel-Belted Radius database, open the appropriate panel and click the **Add** button on the SBR Administrator toolbar. The SBR Administrator displays an Add dialog.

Figure 15: Sample Add Dialog



Every object of the same type must have a unique name. If the name you assign to an item is already being used by another item of the same type, the SBR Administrator displays a warning.

Editing an Entry

To edit an existing entry to the Steel-Belted Radius database, open the appropriate panel and double-click the item you want to change (or select the item and click the **Edit** button on the SBR Administrator toolbar). The SBR Administrator displays the settings for the item you selected in an Edit dialog. The **Save** button remains disabled until the contents of a field in the Edit dialog changes.

Note: You cannot change the name associated with an item in an Edit dialog. To change an item's name, you must cut/paste the item and assign it the name you want it to have.

Cutting/Copying/Pasting Records

When you select an item from a panel displaying tables of items, you can choose **Edit > Cut or Edit > Copy** to cut or copy the item to the Clipboard, and then add a new record to the display by pasting it from the Clipboard by choosing **Edit > Paste.**

The Clipboard can contain one item of each type (RADIUS client, user, etc.) If you copy an item to the Clipboard and then copy another item of the same type, the information for the second item overwrites the information for the first item. Clipboard contents are preserved until you exit the SBR Administrator.

Resizing Columns

You can resize columns in an SBR Administrator table by dragging the column header boundary to the left or right.

Changing Column Sequence

You can change the sequence of columns in an SBR Administrator table by dragging the column headers left and right.

Sorting Information

By default, items in SBR Administrator tables are sorted by Name. You can sort items in any order by clicking a column header.

Previously sorted tables retain their order when the table is sorted on another column. If you want to sort a table by more than one column (for example, sort by address pool and subsort by IP address), click the least-significant column (here, IP Address), and then click the more significant columns (here, Address Pool).

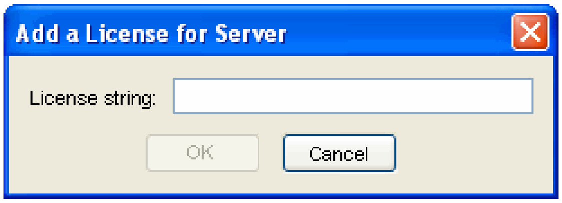
Adding License Keys

Depending upon your purchasing arrangements, your Steel-Belted Radius software might require a new license key at some point after its initial installation.

If you are provided with a new license key by your reseller or by Steel-Belted Radius, you can add the key to an existing Steel-Belted Radius installation as follows:

1. Start the SBR Administrator program and connect to your Steel-Belted Radius server.
2. Select **File > License.**
3. When the Add a License for Server dialog appears, enter the license key and click **OK**.

Figure 16: Add a License for Server Dialog



If the license key you have entered is invalid, the server displays an error message. If this occurs, click **OK** in the message dialog and enter the correct license key.

4. After you have entered a valid license key, the server displays a confirmation message and reminds you that you must restart the server. Click **OK.**

The server does not restart itself automatically after a new license key is added. You must restart Steel-Belted Radius manually to activate the new license key.

Note: The Steel-Belted Radius audit log does not record an entry when you enter license keys through SBR Administrator.

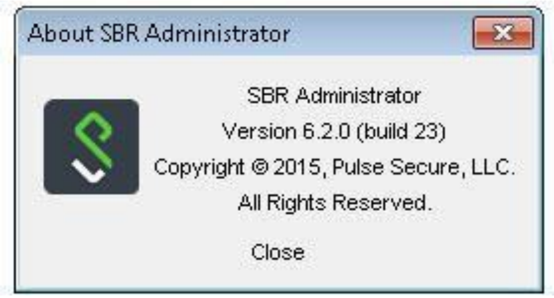
Accessing Online Help

To get help with the SBR Administrator, click the ? (help) button, press F1, or select **Help > Contents.**

Displaying Version Information

To identify the current version of the SBR Administrator, select **Help > About** to open the SBR Administrator dialog.

Figure 17: About SBR Administrator Dialog



About SBR Administrator

SBR Administrator
Version 6.2.0 (build 23)
Copyright © 2015, Pulse Secure, LLC.
All Rights Reserved.

Exiting the SBR Administrator

To exit the SBR Administrator, choose **File > Exit.**

Closing the SBR Administrator has no impact on the Steel-Belted Radius service or daemon.

Chapter 4
Using Web Graphic User Interface (GUI)

WebGUI is an application that enables you to configure settings for SBR Enterprise using a web browser. This chapter presents an overview of how to use the WebGUI to configure Steel Belted Radius Server.

Note: Administrators making large-scale changes to the SBR Enterprise Server database might prefer to use the LDAP command line interface. For more information, see LDAP Configuration Interface.

Running the WebGUI

You launch the WebGUI by running a web browser on your management workstation and opening a connection to the SBR Enterprise server which you want to configure. The WebGUI is designed to support any Web 2.0 browser.

A Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS) connection is established between the web browser in the management workstation and SBR Enterprise Server. This connection is used to communicate configuration data to the server.

Note: Steel Belted Radius server must be running to use WebGUI.

For information on starting the RADIUS server, refer “Starting the Steel-Belted Radius Server” section in the Install and Upgrade Guide.

While using HTTPS, SBR Enterprise provides a Secure Sockets Layer (SSL) certificate to prove its identity and requires credentials when accessing the server. The SSL certificate is self-generated inside the Java Web Server or you can provide a custom SSL certificate by configuring path and password in the application.properties file. For more information, refer to Appendix H - Use Custom SSL Certificate for Launching SBR-E Web UI.

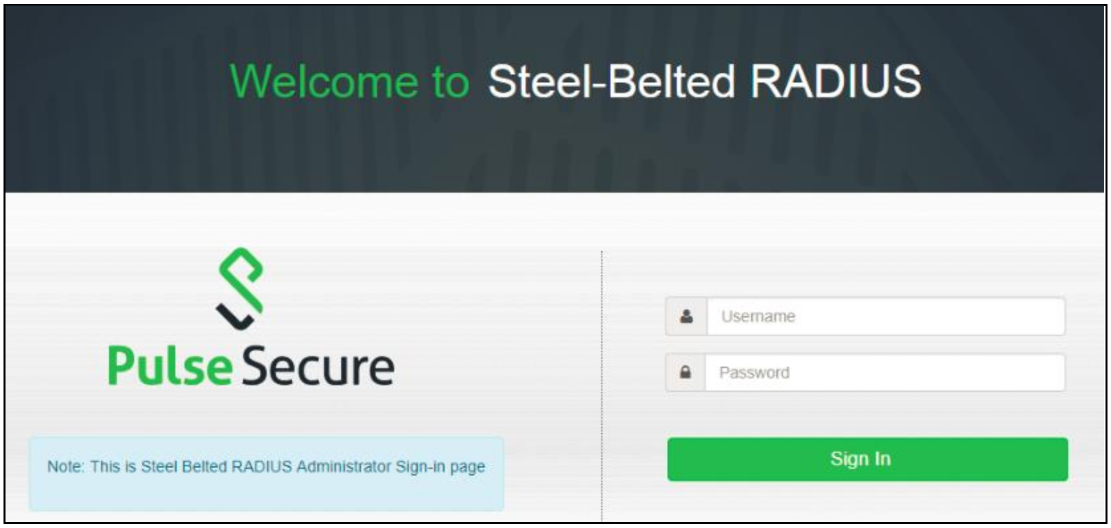
To log in to a SBR Enterprise server:

1. Open a browser connection to the SBR Enterprise server you want to administer:
    a. To administer a SBR Enterprise server running on your local host, enter https://localhost:1810/sbreweb/login.html, where the port assignment of 1812 is the SBR Enterprise’s default TCP port for administration connections.
    b. To administer a SBR Enterprise server running on a remote host, enter https://server:1810/sbreweb/login.html, where the port assignment of 1812 is the SBR Enterprise’s default TCP port for administration connections.
       For example: https://192.168.24.15:1810/sbreweb/login.html

   Note: You must access the WebGUI using HTTPS instead of HTTP.

2. The Login page appears as shown below:

Figure 18: Login Page



Welcome to Steel-Belted RADIUS
Pulse Secure
Note: This is Steel Belted RADIUS Administrator Sign-in page
Username
Password
Sign In

3. Enter your administrator username in the Username.
4. Enter your login password in the Password.
5. Click Sign In.

When you click Sign In, WebGUI establishes an HTTPS connection with the local or remote server. The WebGUI displays an error message if the connection cannot be established.

When you connect to a server, the Home page displays the list of status of the running server, IP addresses, available authentication methods, license information and any initialization errors that might have occurred.

Tested Browsers

WebGUI can be launched in different browsers across different platforms. The following table describes the tested browser versions and the operating systems.

Table 1: WebGUI – Tested Browsers

Browser	Version	Operating System
Google Chrome	73	Windows
Opera	60	Windows
Mozilla Firefox	66	Windows
Edge	42	Windows

Navigating in the WebGUI

This section describes how to use the WebGUI menus and pages.

WebGUI Menus

The WebGUI has following five menus:

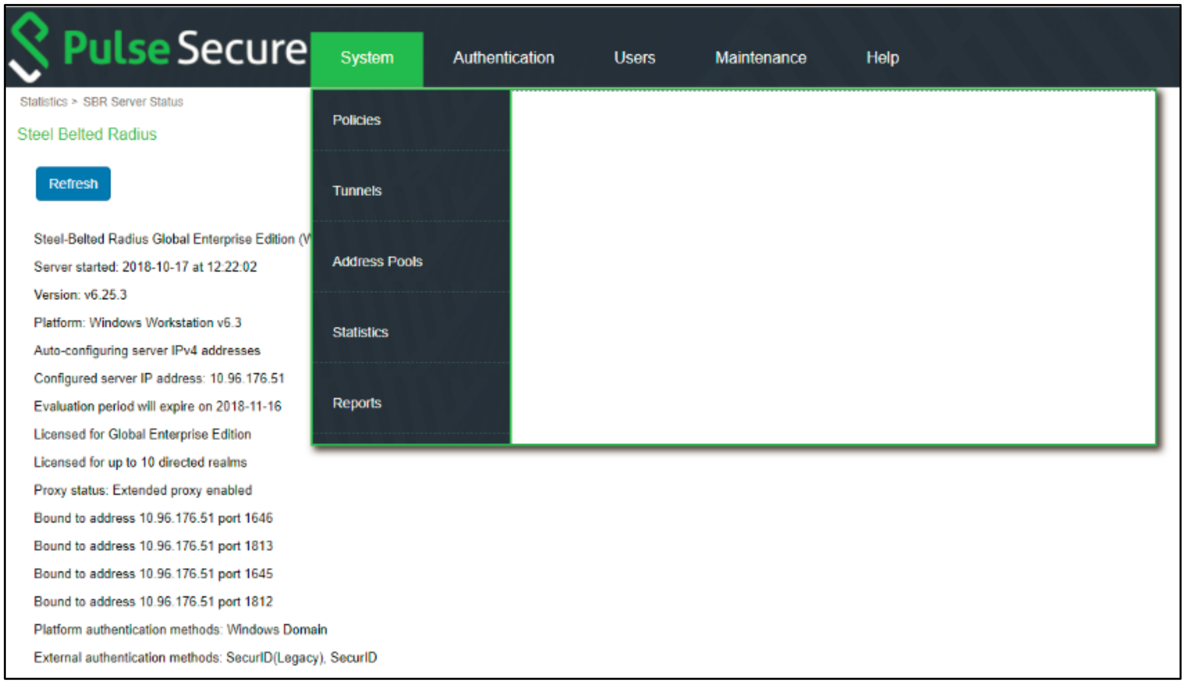
- System
- Authentication
- Users
- Maintenance
- Help

1. System Menu

   The System Menu is logically grouped to contain the following:

   - Policies
   - Tunnels
   - Address Pools
   - Statistics
   - Reports

Figure 19: System Menu



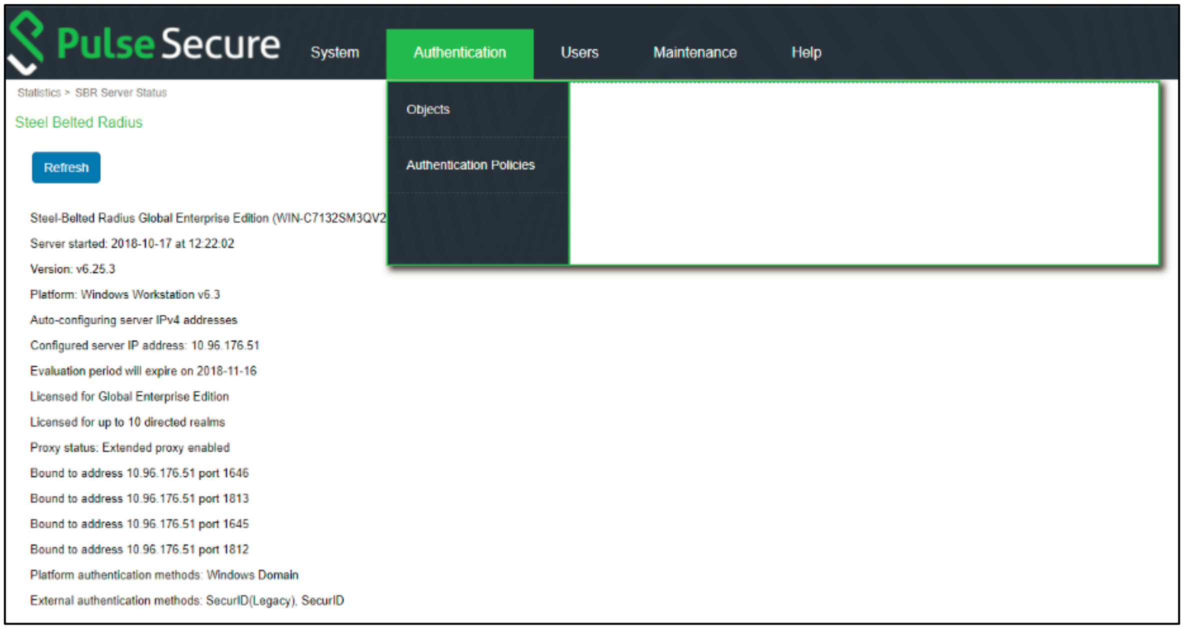
```
Steel-Belted Radius Global Enterprise Edition (W
Server started: 2018-10-17 at 12:22:02
Version: v6.25.3
Platform: Windows Workstation v6.3
Auto-configuring server IPv4 addresses
Configured server IP address: 10.96.176.51
Evaluation period will expire on 2018-11-16
Licensed for Global Enterprise Edition
Licensed for up to 10 directed realms
Proxy status: Extended proxy enabled
Bound to address 10.96.176.51 port 1646
Bound to address 10.96.176.51 port 1813
Bound to address 10.96.176.51 port 1645
Bound to address 10.96.176.51 port 1812
Platform authentication methods: Windows Domain
External authentication methods: SecurID(Legacy), SecurID
```

2. Authentication Menu

   The System Menu is logically grouped to contain the following:

   - Objects
   - Authentication Policies

Figure 20: Authentication Menu



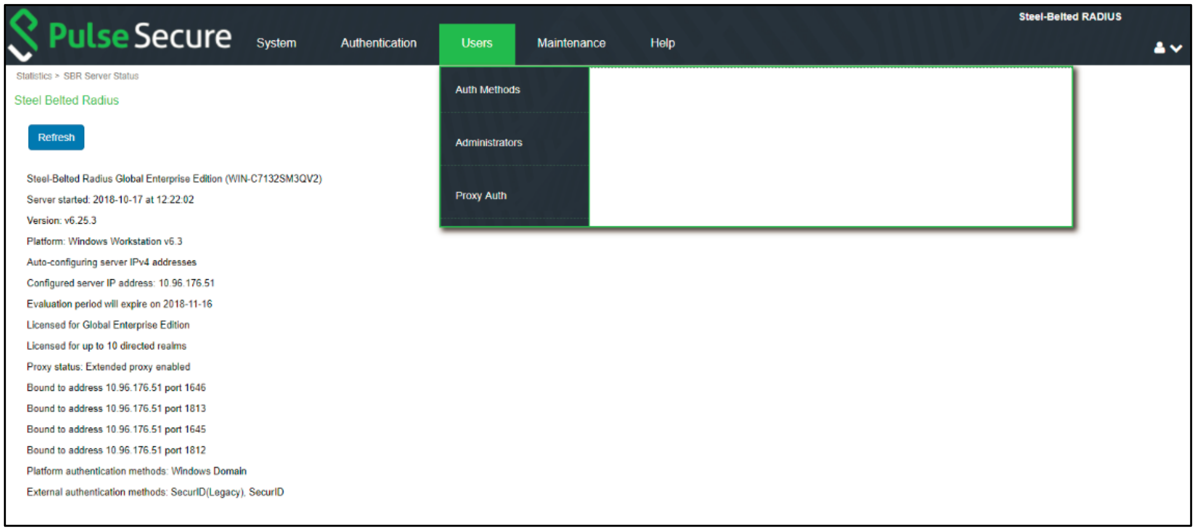
```
Steel-Belted Radius Global Enterprise Edition (WIN-C7132SM3QV2
Server started: 2018-10-17 at 12:22:02
Version: v6.25.3
Platform: Windows Workstation v6.3
Auto-configuring server IPv4 addresses
Configured server IP address: 10.96.176.51
Evaluation period will expire on 2018-11-16
Licensed for Global Enterprise Edition
Licensed for up to 10 directed realms
Proxy status: Extended proxy enabled
Bound to address 10.96.176.51 port 1646
Bound to address 10.96.176.51 port 1813
Bound to address 10.96.176.51 port 1645
Bound to address 10.96.176.51 port 1812
Platform authentication methods: Windows Domain
External authentication methods: SecurID(Legacy), SecurID
```

3. Users Menu

   The System Menu is logically grouped to contain the following:

   - Auth Methods
   - Administrators
   - Proxy Auth

Figure 21: User Menu



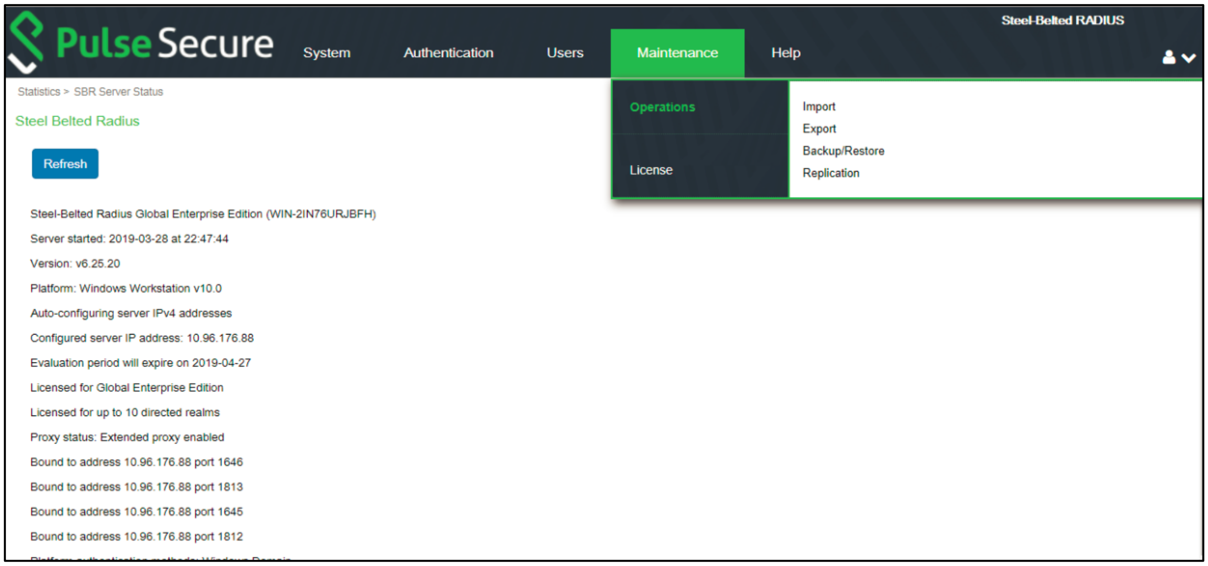
```
Steel-Belted Radius Global Enterprise Edition (WIN-C7132SM3QV2)
Server started: 2018-10-17 at 12:22:02
Version: v6.25.3
Platform: Windows Workstation v6.3
Auto-configuring server IPv4 addresses
Configured server IP address: 10.96.176.51
Evaluation period will expire on 2018-11-16
Licensed for Global Enterprise Edition
Licensed for up to 10 directed realms
Proxy status: Extended proxy enabled
Bound to address 10.96.176.51 port 1646
Bound to address 10.96.176.51 port 1813
Bound to address 10.96.176.51 port 1645
Bound to address 10.96.176.51 port 1812
Platform authentication methods: Windows Domain
External authentication methods: SecurID(Legacy), SecurID
```

4. Maintenance Menu

   The System Menu is logically grouped to contain the following:

   - Operations
   - Licensing

Figure 22: Maintenance Menu



```
Steel-Belted Radius Global Enterprise Edition (WIN-2IN76URJBFH)
Server started: 2019-03-28 at 22:47:44
Version: v6.25.20
Platform: Windows Workstation v10.0
Auto-configuring server IPv4 addresses
Configured server IP address: 10.96.176.88
Evaluation period will expire on 2019-04-27
Licensed for Global Enterprise Edition
Licensed for up to 10 directed realms
Proxy status: Extended proxy enabled
Bound to address 10.96.176.88 port 1646
Bound to address 10.96.176.88 port 1813
Bound to address 10.96.176.88 port 1645
Bound to address 10.96.176.88 port 1812
Platform authentication methods: Windows Domain
```

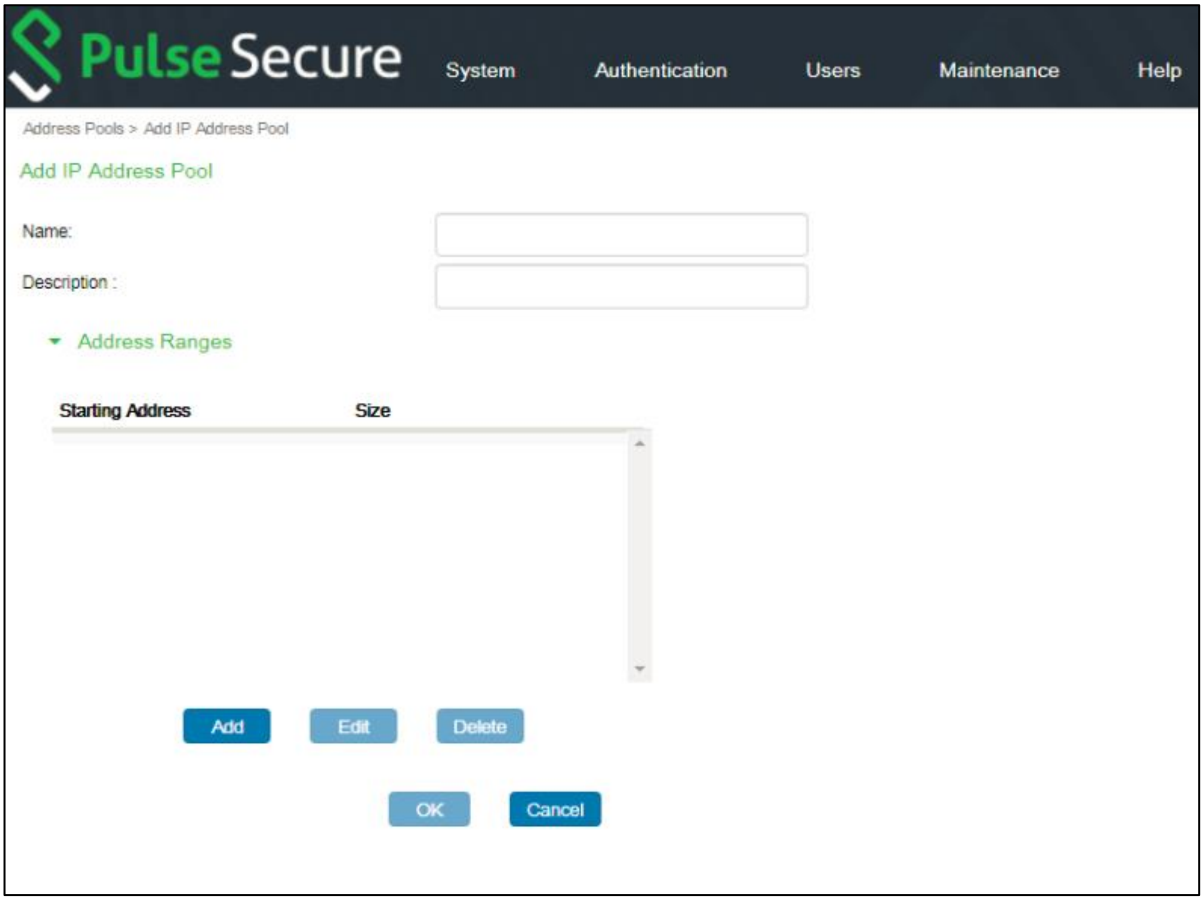
WebGUI Pages

This section summarizes how to use WebGUI pages and controls.

1. Adding an Entry

To add an entry to the SBR Enterprise database, open the appropriate WebGUI page and click **Add**. The WebGUI page displays the Add page.

Figure 23: Adding an Entry



Add IP Address Pool

Name:

Description :

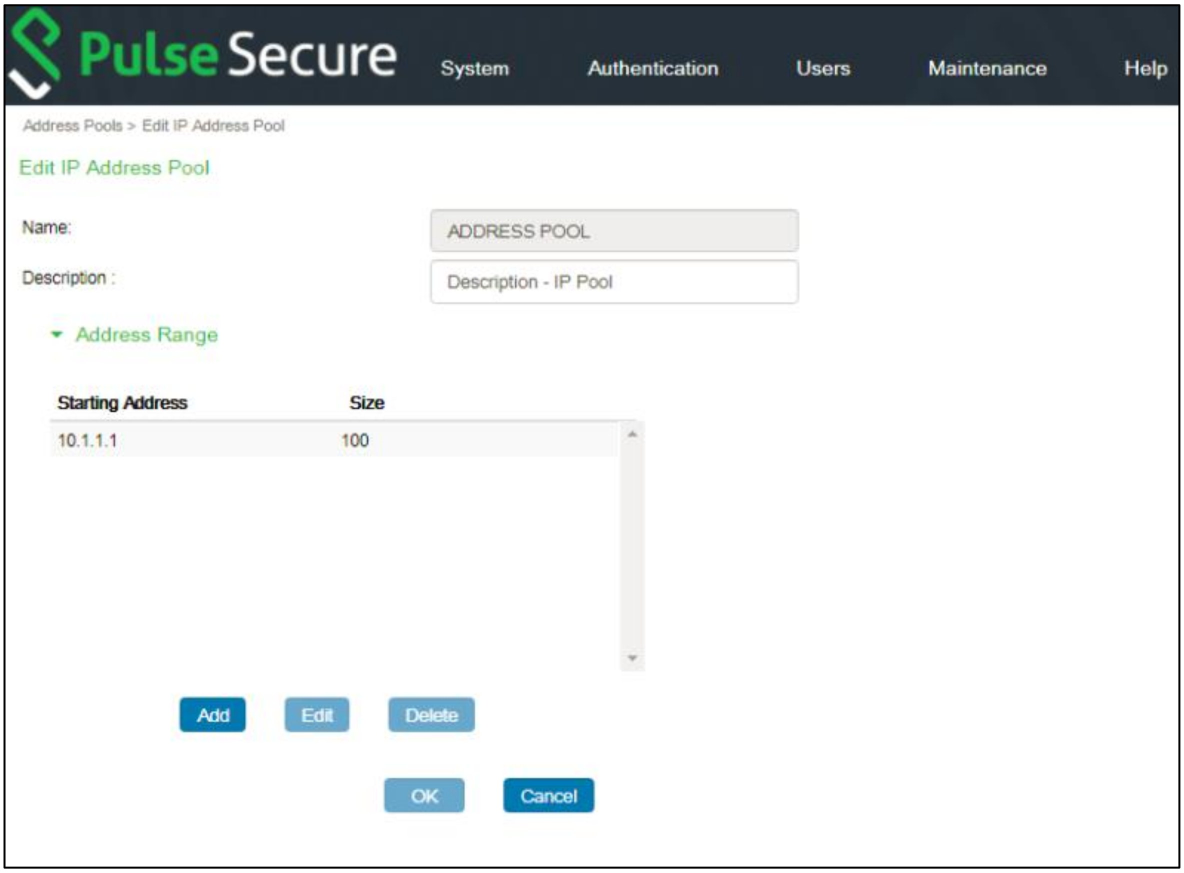
Address Ranges

Starting Address	Size

2. Editing an Entry

   To edit an entry to the SBR Enterprise database, open the appropriate WebGUI page, select the entry using the check box and click **Edit**. The WebGUI page displays the Edit page.

Figure 24: Editing an Entry



Address Pools > Edit IP Address Pool

Edit IP Address Pool

Name: ADDRESS POOL

Description : Description - IP Pool

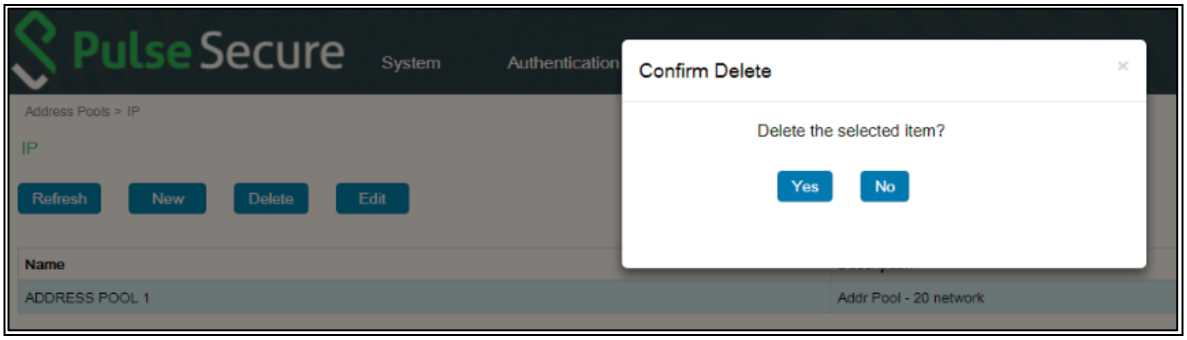
Address Range

Starting Address	Size
10.1.1.1	100

3. Deleting an Entry

   To delete an entry to the SBR Enterprise database, open the appropriate WebGUI page, select the entry using check box and click **Delete**. The WebGUI page displays the Confirm Delete options. To delete multiple configurations, select multiple check boxes and then click **Delete** option.

Figure 25: Deleting an Entry



Address Pools > IP

IP

Confirm Delete

Delete the selected item?

Name	
ADDRESS POOL 1	Addr Pool - 20 network

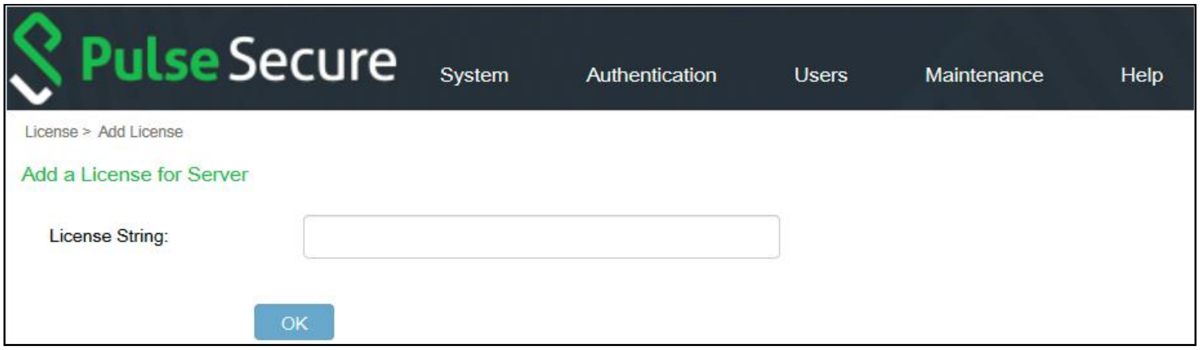
Adding License Keys

Your SBR Enterprise software may require a new license key at some point after its initial installation, Depending upon your purchasing arrangements.

If you are provided with a new license key, you can add the key to an existing SBR Enterprise installation as follows:

1. Start the WebGUI program and connect to your SBR Enterprise server.
2. Select Maintenance > License > Add License.
   The License Registration page appears.

Figure 26: License Registration



3. Enter the license key in the License String.
4. Click OK.
   If the entered license key is invalid, the server displays an error message.
   Click OK in the message dialog box and enter the valid license key.
5. SBR server displays a confirmation message.
6. Click OK.

Note: The server does not restart automatically once a new license key is added.

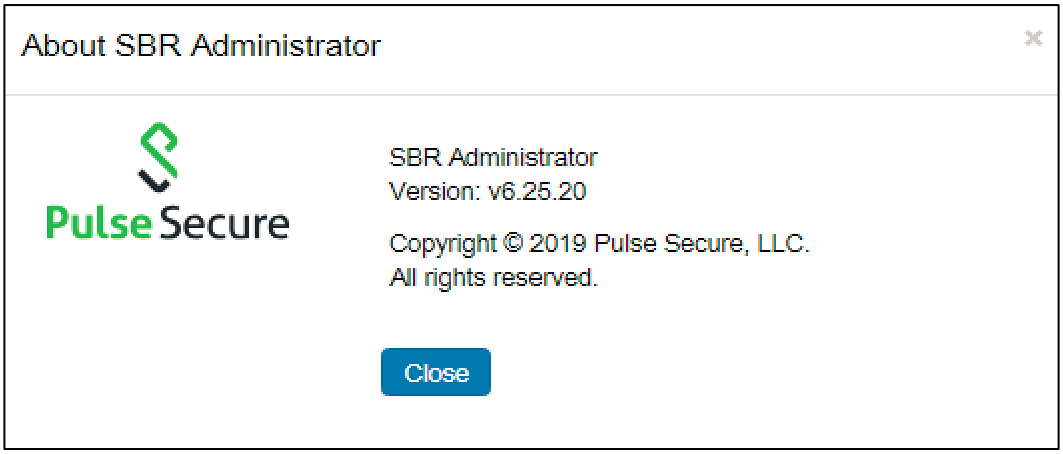
- You must restart Steel Belted Radius Server manually to activate the new license key.
  For more information, refer to [When to Stop and Restart Steel-Belted Radius Enterprise Server.]()
- Refresh and log in again to the WebGUI running on the management workstation to reflect the changes.

Note: The SBR Enterprise audit log does not record an entry when you enter license keys through WebGUI.

Displaying Version Information

To identify the current version of SBR Administrator, select Help > Home Page > About to open the SBR Administrator Dialog. Refer to the following figure:

Figure 27: About SBR Administrator



About SBR Administrator

SBR Administrator
Version: v6.25.20

Copyright © 2019 Pulse Secure, LLC.
All rights reserved.

Exiting the WebGUI

To close the WebGUI,

1. Click Logout.
2. Close the WebGUI browser.

Note: Closing the WebGUI has no impact on the SBR Enterprise daemon.

Chapter 5
Administering RADIUS Clients via Legacy SBR Administrator

This chapter describes how to set up RADIUS clients and client groups via legacy SBR administrator.

A RADIUS client is a network device or software application that contacts Steel-Belted Radius when it needs to authenticate a user or to record accounting information about a network connection.

A RADIUS client group is a collection of network devices or software applications that contacts Steel-Belted Radius to authenticate a user or to record accounting information about a network connection. Members of a RADIUS client group use a contiguous range of IP addresses and use identical settings, such as a shared secret or an IP address pool.

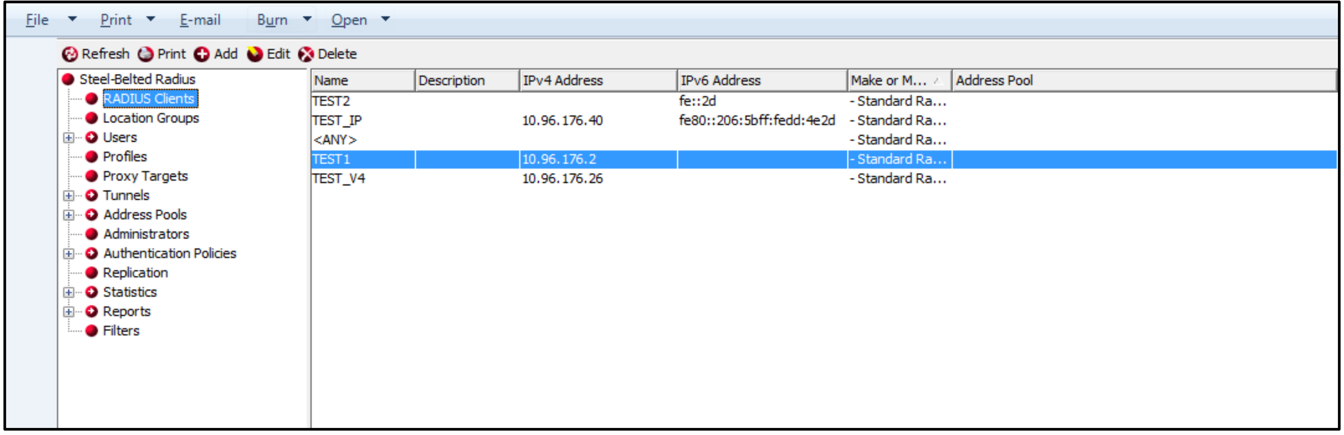
IPv4 – IPv6 Dual stack support for RADIUS clients: two IP addresses can be configured for a RADIUS client; one with IPv4 and another with IPv6. User can add the IP addresses in the “Add RADIUS Client” UI.

Note: If RADIUS client is in IPv6 network, then ensure that IPv6 Networking is enabled in Steel- Belted RADIUS Server. For information on the settings in the radius.ini file, refer to the Steel-Belted RADIUS Reference Guide.

RADIUS Clients Panel

The RADIUS Clients panel lets you identify the devices that you want to define as clients of Steel-Belted RADIUS.

Figure 28: RADIUS Clients Panel



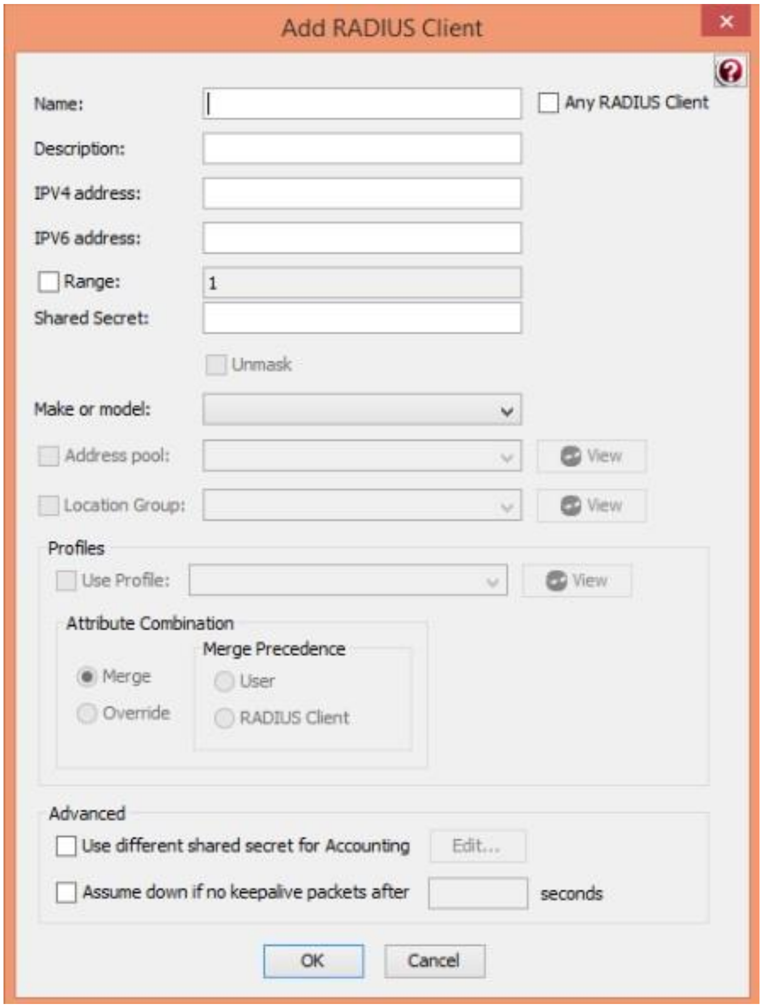
Name	Description	IPv4 Address	IPv6 Address	Make or M...	Address Pool
TEST2			fe::2d	- Standard Ra...	
TEST_IP		10.96.176.40	fe80::206:5bff:fedd:4e2d	- Standard Ra...	
<ANY>				- Standard Ra...	
TEST1		10.96.176.2		- Standard Ra...	
TEST_V4		10.96.176.26		- Standard Ra...	

Adding a RADIUS Client or Client Group

To add a RADIUS client or client group:

1. Choose **Radius Clients** in the sidebar.
2. Click the **Add** button.
   The Add RADIUS Clients page appears.

Figure 29: Add RADIUS Client



Add RADIUS Client

Name: □ Any RADIUS Client

Description:

IPV4 address:

IPV6 address:

□ Range: 1

Shared Secret:

□ Unmask

Make or model:

□ Address pool: View

□ Location Group: View

Profiles

□ Use Profile: View

Attribute Combination

● Merge

○ Override

Merge Precedence

○ User

○ RADIUS Client

Advanced

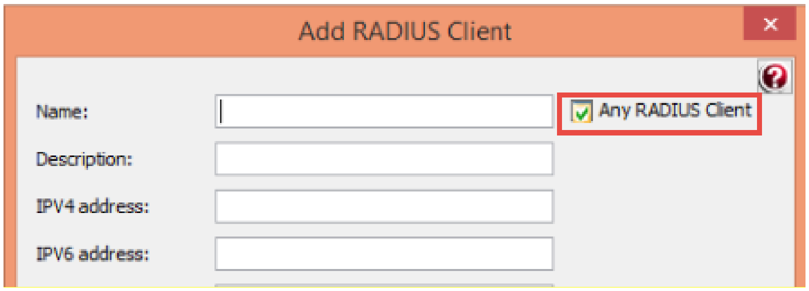
□ Use different shared secret for Accounting Edit...

□ Assume down if no keepalive packets after seconds

OK Cancel

3. Enter the name of the RADIUS client or client group in the Name field.
   Although you can assign any name to a RADIUS client entry, you should use the device’s IP address or DNS hostname to avoid confusion.
   You can create a special RADIUS client entry called **<ANY>** by clicking the Any **RADIUS Client** check box (**Figure 30**). The **<ANY>** RADIUS client allows Steel-Belted Radius to accept requests from any network access device or proxy RADIUS server, as long as the shared secret is correct.

Figure 30: Creating an <ANY> RADIUS Client



Add RADIUS Client

Name: ☑ Any RADIUS Client

Description:

IPV4 address:

IPV6 address:

Note: **IPv4 Address** or **IPv6 Address** fields for an **<ANY>** RADIUS client cannot be edited. **<ANY>** implies that the server accepts requests from any IP address, provided that the shared secret is correct.

4. Optionally, enter a description of the RADIUS client in the **Description** field.
   The description you associate with a RADIUS client is not used during processing.

5. Enter the IPv4 address of the RADIUS client in the **IPV4 Address** field.

6. Enter the IPv6 address of the RADIUS client in the **IPV6 Address** field.
   Alternatively, you can enter the DNS name of the device; the SBR Administrator resolves the name you enter to its corresponding IP address and displays the result in the **IPV4 Address/IPV6 Address** fields.

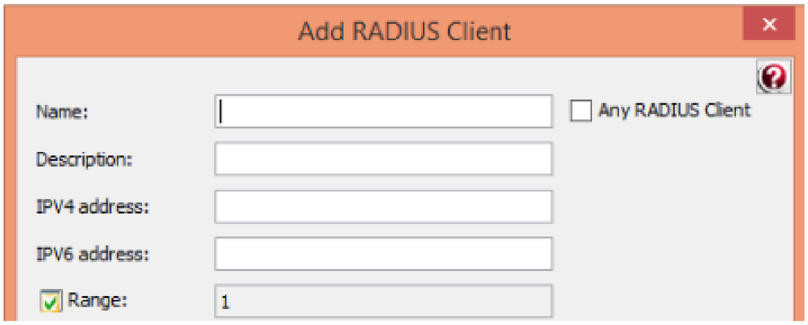
   Note: For a specific RADIUS client entry, it is mandatory to have either IPv4 address or IPv6 address.

   If you want the RADIUS client to use an IPv4 address range, enter the starting address for the range in the **IPV4 Address** field, check the **Range** check box and enter the number of addresses in the range in the **Range** field (**Figure 31**). You can create an address range of as many as 500 addresses in an address range.

   Note: Range is not supported for IPv6 addresses.

   For more information on IPv4 address ranges for RADIUS clients, see "RADIUS Client Groups".

Figure 31: Entering an IPv4 Address Range for a RADIUS Client



7. Enter the authentication shared secret for the RADIUS client in the **Shared secret** field.

   For privacy, asterisks are echoed as you type. You can check **Unmask** to display the characters in the shared secret.
   After you complete configuration of the authentication shared secret on the server side, you must enter the same authentication shared secret when you configure the network access device.

   Note: There will be only one Shared secret field per RADIUS client, and it would be used for both IPv4 and IPv6 addresses.

8. Use the **Make or model** list to select the make and model of your RADIUS client device.

   The **Make or model** selection tells Steel-Belted Radius which dictionary of RADIUS attributes to use when communicating with this client. If you are not sure which make and model you are using or if your device is not in the list, select **- Standard RADIUS -.**

9. If you want the RADIUS client to obtain its IPv4 address from an address pool, check the **Address pool** check box and use the **Address pool** list to specify which address pool to use when authenticating an access request from this RADIUS client.

   Click the **View** button to display details for the address pool you select.

   ⓘ Note: You must configure IP address pools before you set up RADIUS clients if you want the clients to use address pools. For more information, see "Setting Up IP Address Pools".

10. If you want to associate the RADIUS client with a location group, check the **Location Group** check box and use the **Location Group** list to specify the location group to which the RADIUS client belongs.

    Click the **View** button to display details for the RADIUS location group you select.

    ⓘ Note: You must configure RADIUS location groups before you set up RADIUS clients if you want the clients to use location group profiles. For more information, see "Administering RADIUS Location Groups via Legacy SBR Administrator".

11. If you want to associate a profile with the RADIUS client, check the **Use Profile** check box and use the drop-down list to select the profile you want the RADIUS client to use.

    After you select a profile, you can click the **View** button to display the settings configured for the profile.

12. Specify how do you want the profile to interact with the user settings.

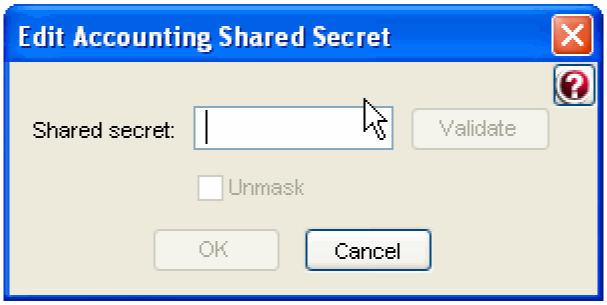
    - If you want attributes in the profile to override identically-named attributes configured for the user, click the **Override User Attributes** check box.

    - If you want attributes in the profile to be merged with identically-named attributes configured for the user, clear the **Override User Attributes** check box and specify whether user or RADIUS client attributes should be used in the event they specify different values for the same attribute.

13. Optionally, specify an accounting secret for the RADIUS client.
    By default, Steel-Belted Radius uses the same shared secret for authentication and accounting. If you want the RADIUS client to use different shared secrets for authentication and accounting:

    a. Check **Use different shared secret for accounting** check box.
    b. Click the **Edit** button.
    c. When the Accounting Shared Secret Page (Figure 32: Accounting Shared Secret Page) opens, enter the shared secret you want the RADIUS client to use for accounting.

Figure 32: Accounting Shared Secret Page



For privacy, asterisks are echoed as you type. You can check the **Unmask** check box to display the characters in the shared secret.

d. Click **OK**.
You must enter the same accounting shared secret when you configure the RADIUS client.

14. Optionally, indicate whether you want to enable keepalive processing and specify how long the server should wait for RADIUS packets from the client before assuming connectivity has been lost.

    If you check the Assume down if no keepalive packets after check box, you can enter a value in the (seconds) field. If the server does not receive any RADIUS packets from this client after the specified number of seconds, the server assumes that the connection to the client is lost or that the client device has failed. When this happens, Steel-Belted Radius gracefully closes any user or tunnel connections it has authenticated for the client. Steel-Belted Radius releases any pooled IP or IPX addresses and adjusts the counts of concurrent user or tunnel connections appropriately.

    Note: If the value you enter in the (seconds) field is too low, valid user or tunnel connections can be lost. For example, during low usage periods, a network access device might send no RADIUS packets to the Steel-Belted Radius server, even though the device is still "up."

15. When you are finished, click OK.

Verifying a Shared Secret

To verify a shared secret on Steel-Belted Radius:

1. Open the RADIUS Clients panel.
2. Select the RADIUS client whose shared secret you want to verify and click the **Edit** button (or double-click the RADIUS client entry).
   The Edit RADIUS Client dialog opens.
3. Enter the shared secret you think is assigned to the RADIUS client in the Shared secret field.
4. Click the **Validate** button.
   The SBR Administrator displays a message indicating whether you entered the correct shared secret.

Deleting a RADIUS Client

To delete a RADIUS client:

1. Open the RADIUS Clients panel.
2. Select the RADIUS client entry you want to delete.
3. Click the **Delete** button on the SBR Administrator toolbar.
4. When you are prompted to confirm the deletion request, click **Yes**.

Chapter 6
Administering RADIUS Clients via WebGUI

This chapter describes how to set up RADIUS clients and client groups via WebGUI.

A RADIUS client is a network device or software application that contacts Steel-Belted Radius when it needs to authenticate a user or to record accounting information about a network connection.

A RADIUS client group is a collection of network devices or software applications that contacts Steel-Belted Radius to authenticate a user or to record accounting information about a network connection. Members of a RADIUS client group use a contiguous range of IP addresses and use identical settings, such as a shared secret or an IP address pool.

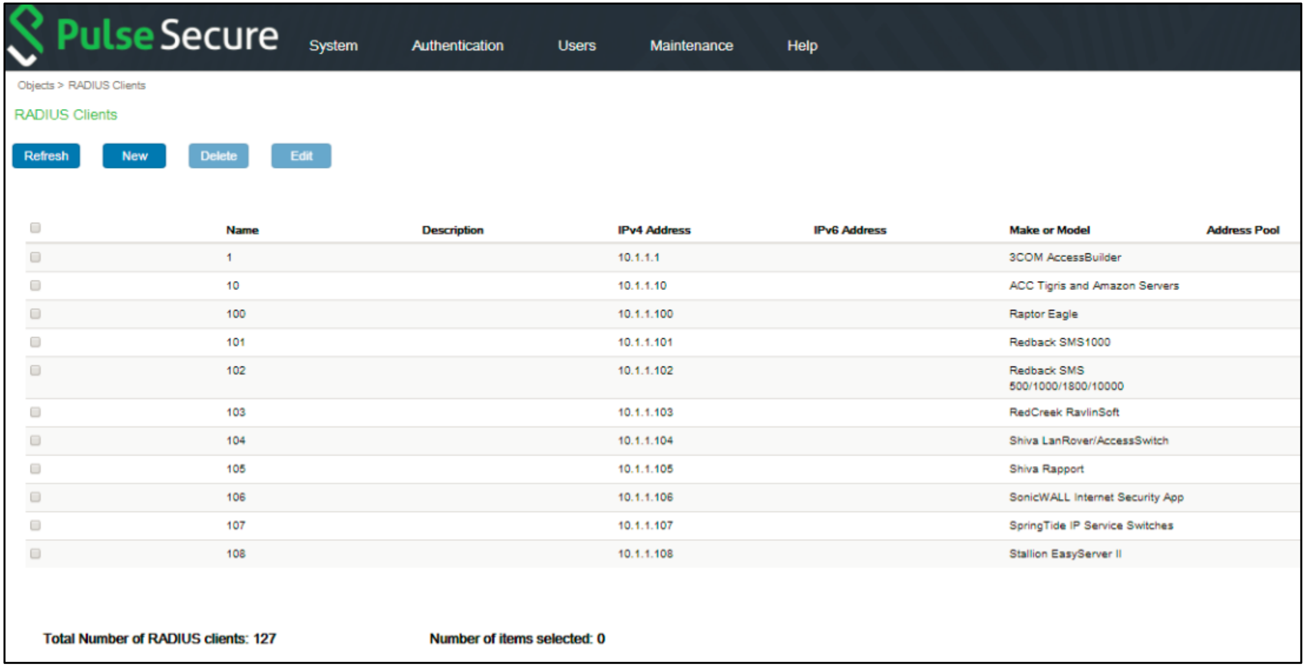
IPv4 – IPv6 Dual stack support for RADIUS clients: two IP addresses can be configured for a RADIUS client; one with IPv4 and another with IPv6. User can add the IP addresses in the “Add RADIUS Client” UI.

Note: If RADIUS client is in IPv6 network, then ensure that IPv6 Networking is enabled in Steel- Belted RADIUS Server. For information on the settings in the radius.ini file, refer to the Steel-Belted RADIUS Reference Guide.

RADIUS Clients Page

The RADIUS Clients page lets you identify the devices that you want to define as clients of Steel-Belted RADIUS.

Figure 33: RADIUS Clients Page



□	Name	Description	IPv4 Address	IPv6 Address	Make or Model	Address Pool
□	1		10.1.1.1		3COM AccessBuilder	
□	10		10.1.1.10		ACC Tigris and Amazon Servers	
□	100		10.1.1.100		Raptor Eagle	
□	101		10.1.1.101		Redback SMS1000	
□	102		10.1.1.102		Redback SMS 500/1000/1800/10000	
□	103		10.1.1.103		RedCreek RavlinSoft	
□	104		10.1.1.104		Shiva LanRover/AccessSwitch	
□	105		10.1.1.105		Shiva Rapport	
□	106		10.1.1.106		SonicWALL Internet Security App	
□	107		10.1.1.107		SpringTide IP Service Switches	
□	108		10.1.1.108		Stallion EasyServer II	

Total Number of RADIUS clients: 127

Number of items selected: 0

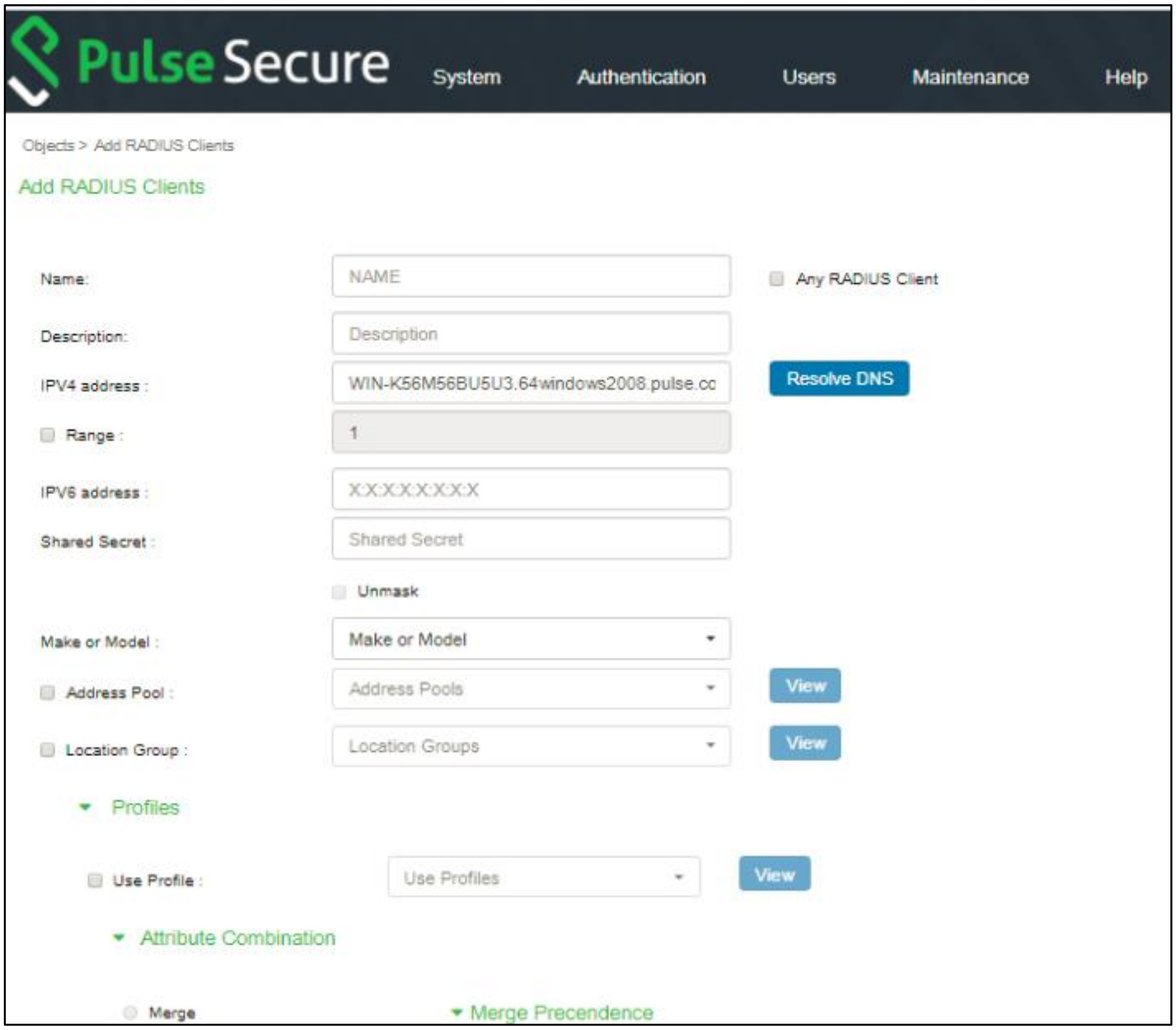
Adding a RADIUS Client or Client Group

To add a RADIUS client or client group:

1. Choose **Radius Clients** from the main menu **Authentication** and sub menu **Objects.**
2. Click the **New** button.

The following Add RADIUS Clients page appears:

Figure 34: Add RADIUS Client



Name: NAME □ Any RADIUS Client

Description: Description

IPV4 address : WIN-K56M56BU5U3.64windows2008.pulse.cc

□ Range : 1

IPV6 address : X:X:X:X:X:X:X:X

Shared Secret : Shared Secret

□ Unmask

Make or Model : Make or Model

□ Address Pool : Address Pools

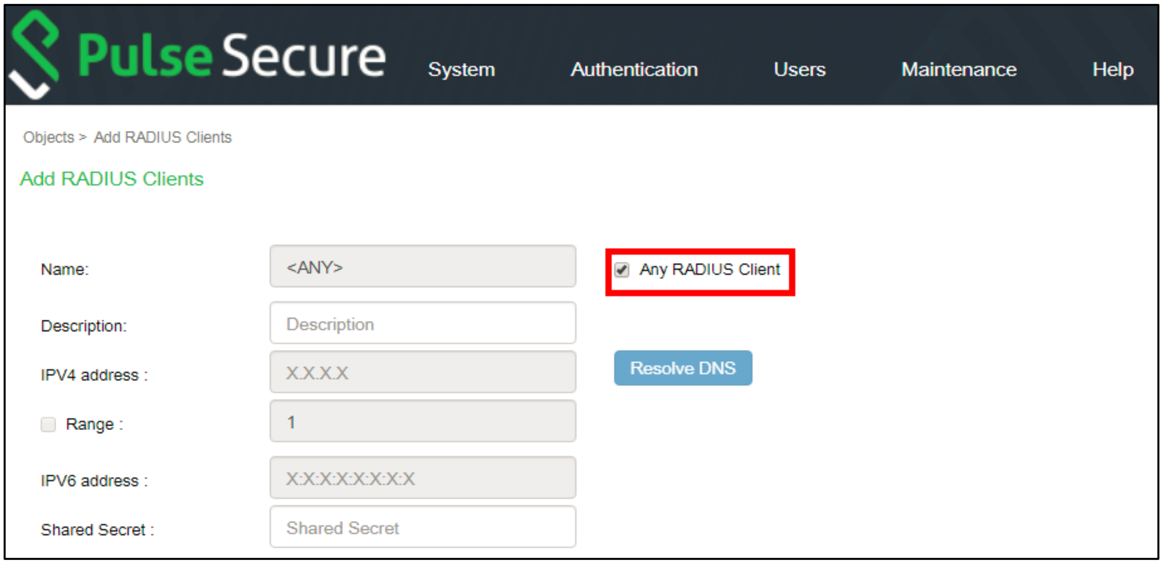
□ Location Group : Location Groups

□ Use Profile : Use Profiles

○ Merge

3. Enter the name of the RADIUS client or client group in the Name field.
   Although you can assign any name to a RADIUS client entry, you should use the device's IP address or DNS hostname to avoid confusion.
   You can create a special RADIUS client entry called **<ANY>** by clicking the **Any RADIUS Client** check box) (Figure 35: Creating an <ANY> RADIUS Client). The **<ANY>** RADIUS client allows Steel-Belted Radius to accept requests from any network access device or proxy RADIUS server, as long as the shared secret is correct.

Figure 35: Creating an <ANY> RADIUS Client



Objects > Add RADIUS Clients

Add RADIUS Clients

Name: <ANY> ☑ Any RADIUS Client

Description: Description

IPV4 address : X.X.X.X

□ Range : 1

IPV6 address : X:X:X:X:X:X:X:X

Shared Secret : Shared Secret

Note: **IPv4 Address** or **IPv6 Address** fields for an **<ANY>** RADIUS client cannot be edited. **<ANY>** implies that the server accepts requests from any IP address, provided that the shared secret is correct.

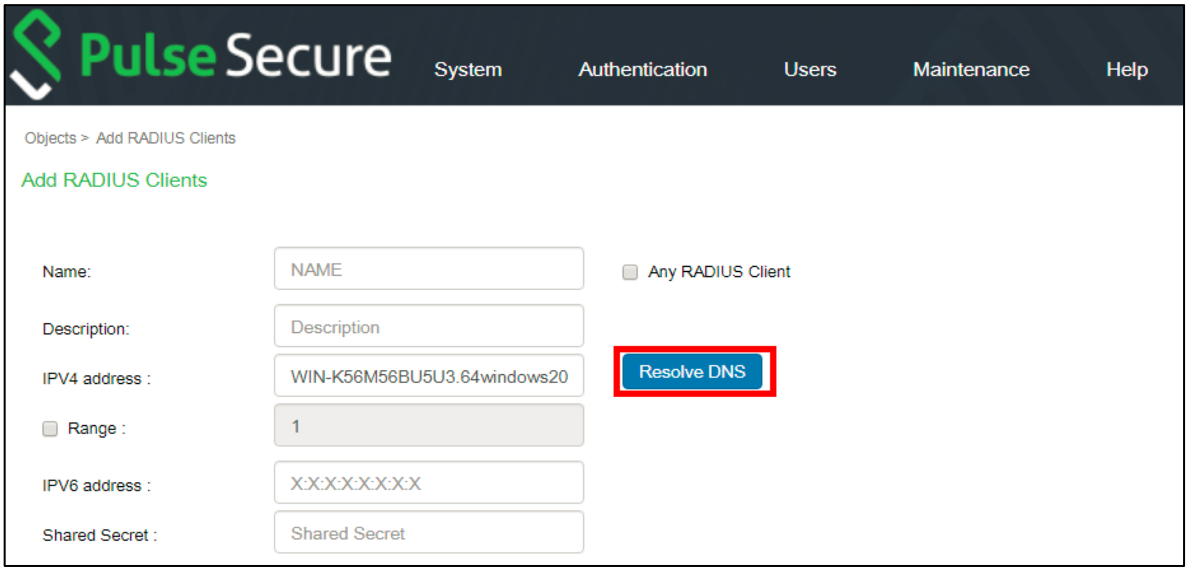
4. Optionally, enter a description of the RADIUS client in the **Description** field.
   The description you associate with a RADIUS client is not used during processing.

5. Enter the IPv4 address of the RADIUS client in the **IPV4 Address** field.

6. Enter the IPv6 address of the RADIUS client in the **IPV6 Address** field.

   Alternatively, you can enter the DNS name of the device and click the Resolve DNS button; the SBR WebGUI resolves the name you enter to its corresponding IP address and displays the result in the **IPV4 Address** fields. Refer to the following figure:

Figure 36: Resolve DNS



Objects > Add RADIUS Clients

Add RADIUS Clients

Name: NAME □ Any RADIUS Client

Description: Description

IPV4 address : WIN-K56M56BU5U3.64windows20

□ Range : 1

IPV6 address : X:X:X:X:X:X:X:X

Shared Secret : Shared Secret

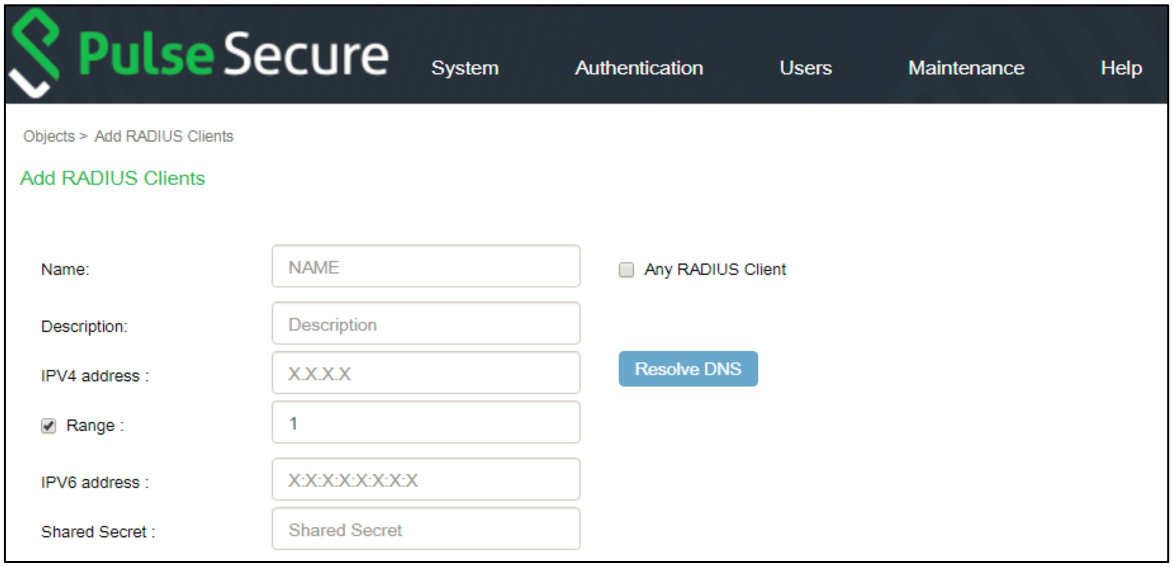
Note: For a specific RADIUS client entry, it is mandatory to have either IPv4 address or IPv6 address.

If you want the RADIUS client to use an IPv4 address range, enter the starting address for the range in the **IPV4 Address** field, check the **Range** check box and enter the number of addresses in the range in the **Range** field (Figure 37: Entering an IPv4 Address Range for a RADIUS Client). You can create an address range of as many as 500 addresses in an address range.

Note: Range is not supported for IPv6 addresses.

For more information on IPv4 address ranges for RADIUS clients, see "RADIUS Client Groups".

Figure 37: Entering an IPv4 Address Range for a RADIUS Client



Name:	NAME	□ Any RADIUS Client
Description:	Description	
IPV4 address :	X.X.X.X	
☑ Range :	1	
IPV6 address :	X:X:X:X:X:X:X:X	
Shared Secret :	Shared Secret	

7. Enter the authentication shared secret for the RADIUS client in the **Shared secret** field.

   For privacy, asterisks are echoed as you type. You can check **Unmask** to display the characters in the shared secret.
   After you complete configuration of the authentication shared secret on the server side, you must enter the same authentication shared secret when you configure the network access device.

   Note: There will be only one Shared secret field per RADIUS client, and it would be used for both IPv4 and IPv6 addresses.

8. Use the **Make or model** list to select the make and model of your RADIUS client device.

   The **Make or model** selection tells Steel-Belted Radius which dictionary of RADIUS attributes to use when communicating with this client. If you are not sure which make and model you are using or if your device is not in the list, select **- Standard RADIUS -.**

9. If you want the RADIUS client to obtain its IPv4 address from an address pool, check the **Address pool** check box and use the **Address pool** list to specify which address pool to use when authenticating an access request from this RADIUS client.

Click the **View** button to display details for the address pool you select.

Note: You must configure IP address pools before you set up RADIUS clients if you want the clients to use address pools. For more information, see "Setting Up IP Address Pools".

10. If you want to associate the RADIUS client with a location group, check the **Location Group** check box and use the **Location Group** list to specify the location group to which the RADIUS client belongs.

    Click the **View** button to display details for the RADIUS location group you select.

    Note: You must configure RADIUS location groups before you set up RADIUS clients if you want the clients to use location group profiles. For more information, see "Administering RADIUS Location Groups via WebGUI".

11. If you want to associate a profile with the RADIUS client, check the **Use Profile** check box and use the drop-down list to select the profile you want the RADIUS client to use.

    After you select a profile, you can click the **View** button to display the settings configured for the profile.

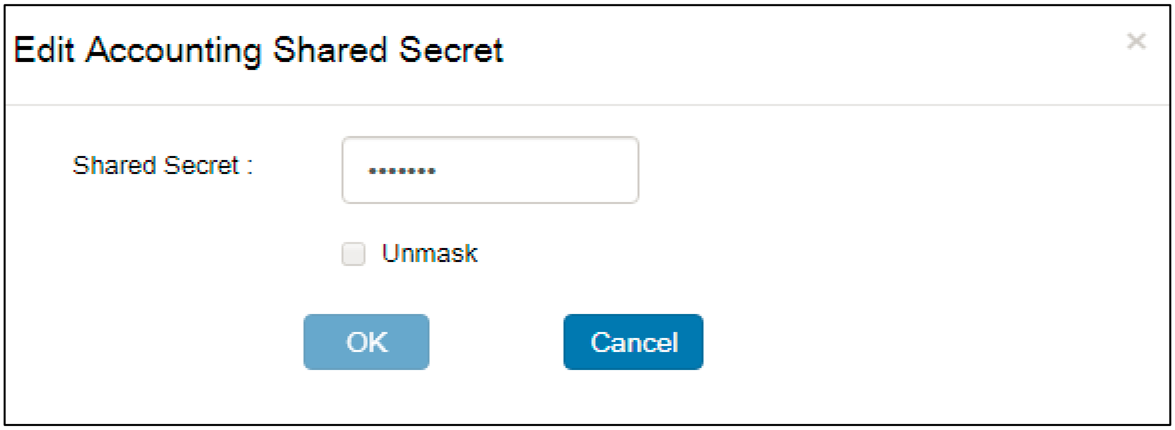
12. Specify how do you want the profile to interact with the user settings.

    - If you want attributes in the profile to override identically-named attributes configured for the user, select **Override** option.
    - If you want attributes in the profile to be merged with identically-named attributes configured for the user, select the Merge option button and then select either **User** or **RADIUS Client** option button to take precedence if the attributes in the profile specify different values for the same single-value or ordered multiple value attribute.

13. Optionally, specify an accounting secret for the RADIUS client.
    By default, Steel-Belted Radius uses the same shared secret for authentication and accounting. If you want the RADIUS client to use different shared secrets for authentication and accounting:

    a. Check **Use different shared secret for accounting** check box.
    b. Click **Edit**.
    c. When the Accounting Shared Secret page (Figure 38: Accounting Shared Secret Page) opens, enter the shared secret you want the RADIUS client to use for accounting.

Figure 38: Accounting Shared Secret Page


Edit Accounting Shared Secret

Shared Secret : *******

☐ Unmask

For privacy, asterisks are echoed as you type. You can check the **Unmask** check box to display the characters in the shared secret.

d. Click **OK**.
You must enter the same accounting shared secret when you configure the RADIUS client.

14. Optionally, indicate whether you want to enable keepalive processing and specify how long the server should wait for RADIUS packets from the client before assuming connectivity has been lost.

    If you check the Assume down if no keepalive packets after check box, you can enter a value in the (seconds) field. If the server does not receive any RADIUS packets from this client after the specified number of seconds, the server assumes that the connection to the client is lost or that the client device has failed. When this happens, Steel-Belted Radius gracefully closes any user or tunnel connections it has authenticated for the client. Steel-Belted Radius releases any pooled IP or IPX addresses and adjusts the counts of concurrent user or tunnel connections appropriately.

    Note: If the value you enter in the (seconds) field is too low, valid user or tunnel connections can be lost. For example, during low usage periods, a network access device might send no RADIUS packets to the Steel-Belted Radius server, even though the device is still "up."

15. When you are finished, click OK.

Verifying a Shared Secret

To verify a shared secret on Steel-Belted Radius:

1. Open the RADIUS Clients page.
2. Select the RADIUS client whose shared secret you want to verify and click **Edit.**
   The Edit RADIUS Client page opens.
3. Enter the shared secret you think is assigned to the RADIUS client in the Shared secret field.
4. Click **Validate**.
   The WebGUI displays a message indicating whether you entered the correct shared secret.

Deleting a RADIUS Client

To delete a RADIUS client:

1. Open the RADIUS Clients page.
2. Select the RADIUS client entry you want to delete.
3. Click **Delete** on RADIUS Clients Page.
4. When you are prompted to confirm the deletion request, click **Yes**.

Chapter 7
Administering RADIUS Location Groups via Legacy SBR Administrator

This chapter describes how to set up RADIUS location groups via legacy SBR administrator.

About Location Groups

RADIUS location groups allow you to assign an attribute profile to a user based on the network access device through which the user is connecting to your network. You can specify that users must use only the attributes specified in the profile associated with the location group, or you can specify that attributes from the NAD profile are merged with attributes from the user's profile.

To simplify administration of RADIUS client profiles, you can associate profiles with location groups and then associate location groups with RADIUS clients.

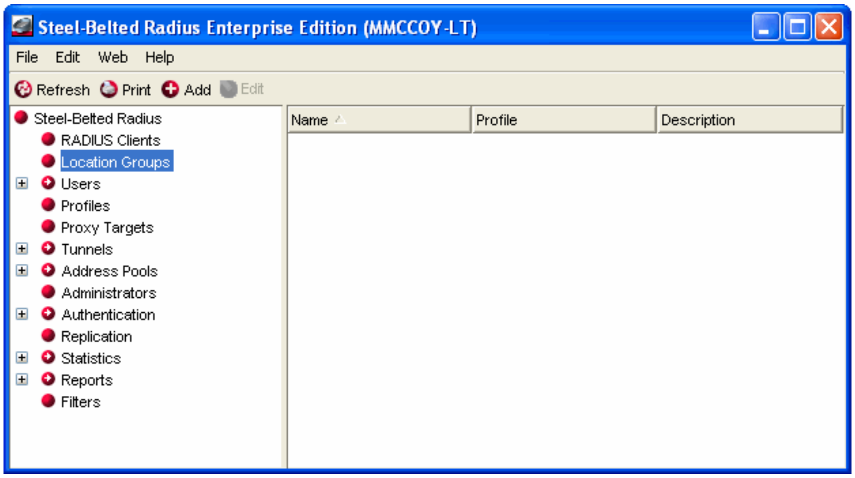
If you set the AddFunkLocationGroupIdToRequest parameter in the [Configuration] section of the radius.ini file to 1 (AddFunkLocationGroupIdToRequest = 1), if an inbound RADIUS authentication or accounting request is matched to a location group, then a Funk-Location-Group-Id attribute with a value set to the name of the location group is added to the RADIUS request. You can then use the location group name associated with the RADIUS client for SQL, LDAP, and checklist attribute processing.

Note: The location group name is case-sensitive. If you create a location group called LOC1 and then assign a user a profile with a checklist attribute called Loc1, the user will be rejected.

Location Groups Panel

The Location Groups panel lets you identify the RADIUS location groups you want to associate with Steel-Belted Radius clients.

Figure 39: Location Groups Panel



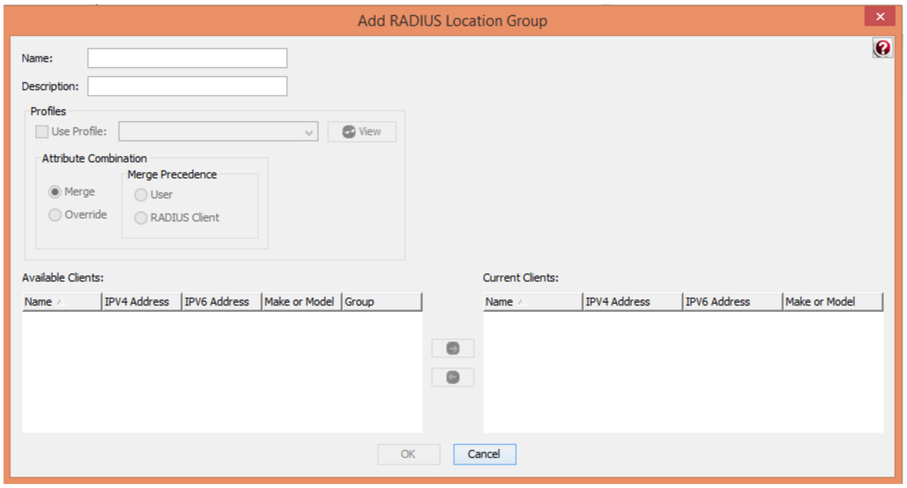
Adding a Location Group

To add a RADIUS location group:

1. Choose **Steel-Belted Radius > Location Groups** in the sidebar.

When the RADIUS Location Groups panel (**Figure 39**) appears, click the **Add** button. The Add RADIUS Location Group dialog (Figure 40: Add RADIUS Client Dialog) appears.

Figure 40: Add RADIUS Client Dialog



2. Enter the name of the RADIUS location group in the **Name** field in upper case.

   Note: The location group name is case-sensitive. If you create a location group called LOC1 and then assign a user a profile with a checklist attribute called Loc1, the user will be rejected.

3. Optionally, enter a description of the RADIUS location group in the **Description** field.
   The description you associate with a RADIUS location group is not used during processing.

4. If you want to associate a profile with the RADIUS location group, click the Use Profile check box and use the drop-down list to select the profile you want the location group to use.

   After you select a profile, you can click the View button to display the settings configured for that profile.

   For more information on profiles, refer to "Administering Profiles via Legacy SBR Administrator".

5. Specify how you want the profile to interact with the user settings.

   - If you want attributes in the specified profile to override identically-named attributes configured for the user, click the **Override User Attributes** check box.

   - If you want attributes in the profile to be merged with identically-named attributes configured for the user, clear the **Override User Attributes** check box and specify whether user or RADIUS client attributes should be used in the event they specify different values for the same single-value or ordered-multiple-value attribute.

6. Identify the RADIUS clients that belong to the location group by selecting one or more client entries in the **Available Clients** list and clicking the right arrow button.
   You can remove a RADIUS client from the location group by selecting the appropriate entry in the **Current Clients** list and clicking the left arrow button.

7. When you are finished, click OK. s

Deleting a RADIUS Location Group

To delete a RADIUS location group:

1. Open the RADIUS Location Groups panel.
2. Select the RADIUS location group you want to delete.
3. Choose **Edit > Delete.**
4. When a dialog asking you to confirm the delete request appears, click **Yes.**

Chapter 8
Administering RADIUS Location Groups via WebGUI

This chapter describes how to set up RADIUS location groups via WebGUI.

About Location Groups

RADIUS location groups allow you to assign an attribute profile to a user based on the network access device through which the user is connecting to your network. You can specify that users must use only the attributes specified in the profile associated with the location group, or you can specify that attributes from the NAD profile are merged with attributes from the user’s profile.

To simplify administration of RADIUS client profiles, you can associate profiles with location groups and then associate location groups with RADIUS clients.

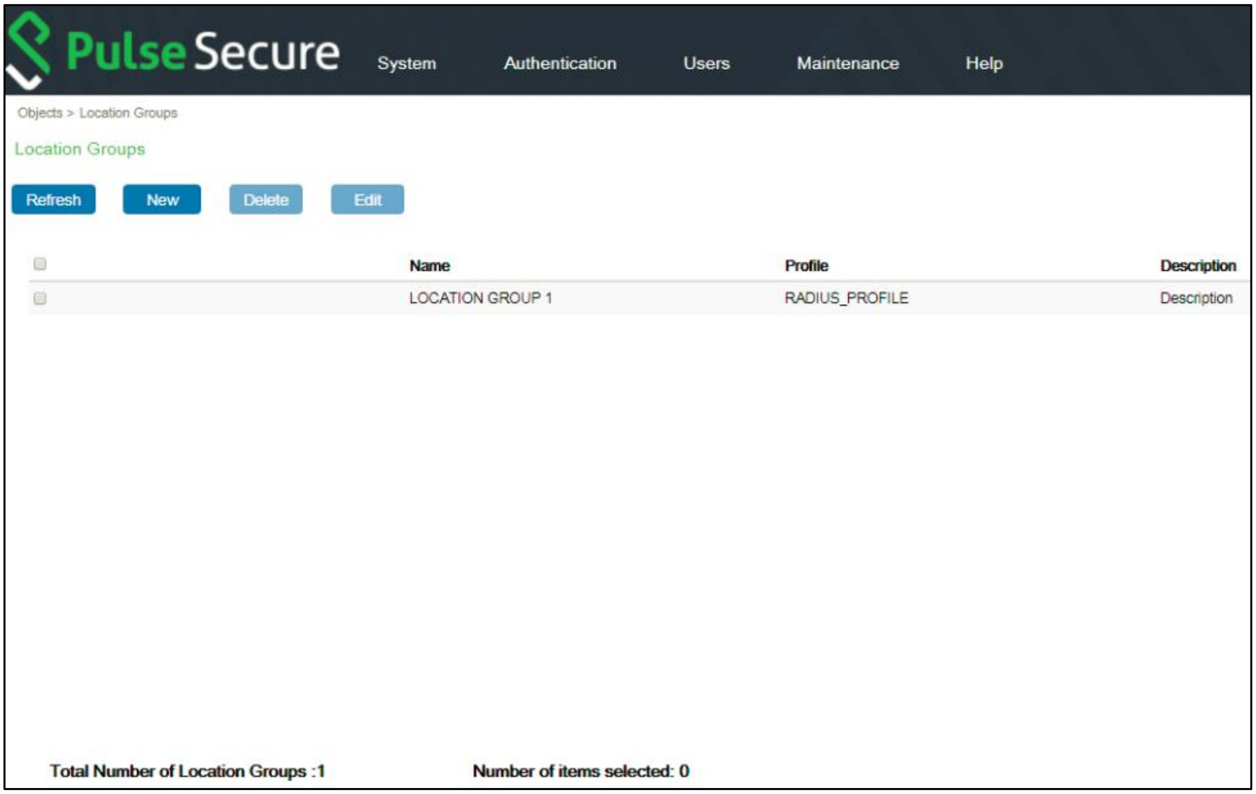
If you set the AddFunkLocationGroupIdToRequest parameter in the [Configuration] section of the radius.ini file to 1 (AddFunkLocationGroupIdToRequest = 1), if an inbound RADIUS authentication or accounting request is matched to a location group, then a Funk-Location-Group-Id attribute with a value set to the name of the location group is added to the RADIUS request. You can then use the location group name associated with the RADIUS client for SQL, LDAP, and checklist attribute processing.

ⓘ Note: The location group name is case-sensitive. If you create a location group called LOC1 and then assign a user a profile with a checklist attribute called Loc1, the user will be rejected.

Location Groups Page

The Location Groups page lets you identify the RADIUS location groups you want to associate with Steel-Belted Radius clients.

Figure 41: Location Groups Page



□	Name	Profile	Description
□	LOCATION GROUP 1	RADIUS_PROFILE	Description

Total Number of Location Groups :1

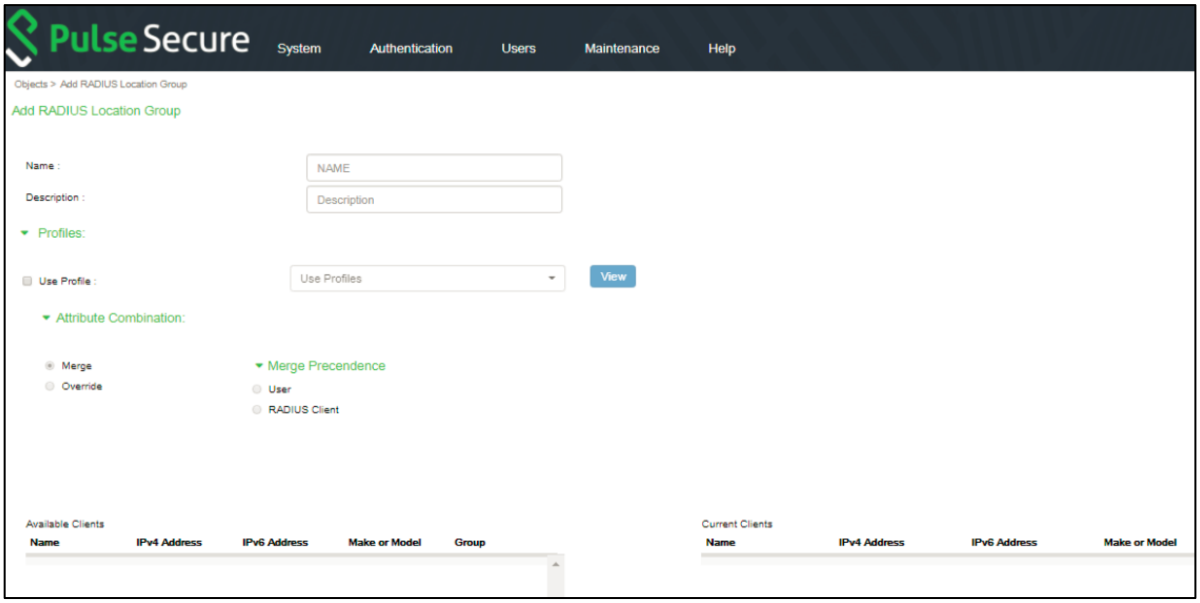
Number of items selected: 0

Adding a Location Group

To add a RADIUS location group:

1. Choose **Location Groups** in from the main menu Authentication and sub menu **Objects**.
2. Click **New**.
   The Add Radius Location Group page appears as below:

Figure 42: Add RADIUS Location Page



Objects > Add RADIUS Location Group

Add RADIUS Location Group

Name : NAME

Description : Description

Profiles:

□ Use Profile : Use Profiles

Attribute Combination:

● Merge
○ Override

Merge Precendence
○ User
○ RADIUS Client

Available Clients

Name	IPv4 Address	IPv6 Address	Make or Model	Group

Current Clients

Name	IPv4 Address	IPv6 Address	Make or Model

3. Enter the name of the RADIUS location group in the **Name** field in upper case.

   Note: The location group name is case-sensitive. If you create a location group called LOC1 and then assign a user a profile with a checklist attribute called Loc1, the user will be rejected.

4. Optionally, enter a description of the RADIUS location group in the **Description** field.
   The description you associate with a RADIUS location group is not used during processing.

5. If you want to associate a profile with the RADIUS location group, click the **Use Profile** check box and use the drop-down list to select the profile you want the location group to use.

   After you select a profile, you can click the **View** button to display the settings configured for that profile.

   For more information on profiles, refer to “Administration Profiles”.

6. Specify how you want the profile to interact with the user settings.

   - If you want attributes in the specified profile to override identically-named attributes configured for the user, click the **Override User Attributes** check box.

   - If you want attributes in the profile to be merged with identically-named attributes configured for the user, clear the **Override User Attributes** check box and specify whether user or RADIUS client attributes should be used in the event they specify different values for the same single-value or ordered-multiple-value attribute.

7. Identify the RADIUS clients that belong to the location group by selecting one or more client entries in the **Available Clients** list and clicking the right arrow button.
   You can remove a RADIUS client from the location group by selecting the appropriate entry in the **Current Clients** list and clicking the left arrow button.

8. When you are finished, click **OK.**

Deleting a RADIUS Location Group

To delete a RADIUS location group:

1. Open the RADIUS Location Groups page.
2. Select the RADIUS location group you want to delete.
3. Click the Delete button on the RADIUS Location Group page.
4. When you are prompted to confirm the deletion request, click **Yes**.

Chapter 9

Administering Users via Legacy SBR Administrator

This chapter describes how to add users to the Steel-Belted Radius database via legacy SBR administrator.

User Files

The following files establish settings for setting up users. For more information about these files, refer to the Steel-Belted Radius Reference Guide.

Table 22: User Account Files

File Name	Function
lockout.ini	Configures settings for when Steel-Belted Radius should lock user accounts after repeated failed login attempts.
redirect.ini	Configures settings for when Steel-Belted Radius should redirect users after repeated failed login attempts.
radius.ini	Specifies (among other things) the settings relating to RSA SecurID support in Steel-Belted Radius.
securid.ini	Specifies the prompt strings returned to SecurID users during login and authentication.
tacplus.ini	Specifies the name of the TACACS+ server and the shared secret used to validate communication between the Steel-Belted Radius server and the TACACS+ server.

Users Panels

The Users entry in the sidebar has as many as five sub-entries, as described in **Table 23**. Each user entry in the Steel-Belted Radius database identifies one method by which the server can authenticate a specific user.

Table 23: User Panels

User Panel	Function	Available On Server Running OS
Local	Lists the native users in the local Steel-Belted Radius database. You must display the Native Usler panel to add, edit and delete native users.	Windows Linux
Domain	Lists the users authenticated using WindowsDomain authentication. You must display the Domain tab to add, edit, and delete Domain users.	Windows
SecurID	Lists the users authenticated using RSA SecurID authentication. You must display the **SecurID** tab to add, edit, and delete SecurID users	Windows Linux

User Panel	Function	Available On Server Running OS
TACACS+	Lists the users authenticated using TACACS+. You must display the **TACACS+** tab to add, edit, and delete TACACS+ users.	Windows Linux
UNIX	Lists the users and groups authenticated using UNIX authentication. You must display the **UNIX** tab to add, edit, and delete UNIX users.	Linux

Note: You can populate the user database for Steel-Belted Radius by entering information in the Users panel or by importing data from other servers. For more information on importing user information, see Appendix E."

Setting Up Native Users

Native user entries require you to enter the user's name and password into the Steel-Belted Radius database. For all other types of user entry, the server relies on another database to confirm the user's password.

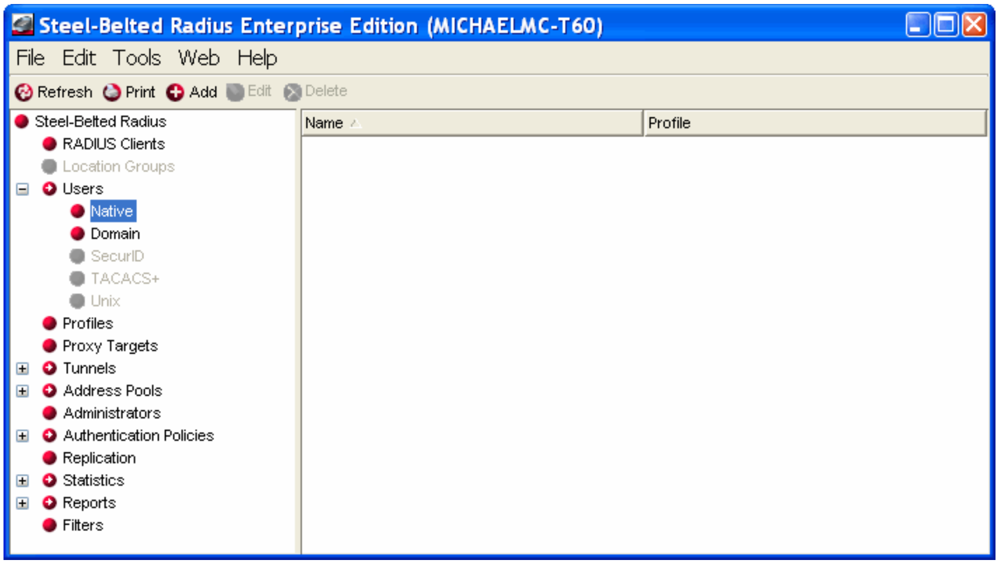
Note: You must define a native user entry for every user who requires remote access to a Windows network. For example, you can accommodate Linux or Macintosh users by adding them as native users.

Adding a Native User

To add a native user to the Steel-Belted Radius database:

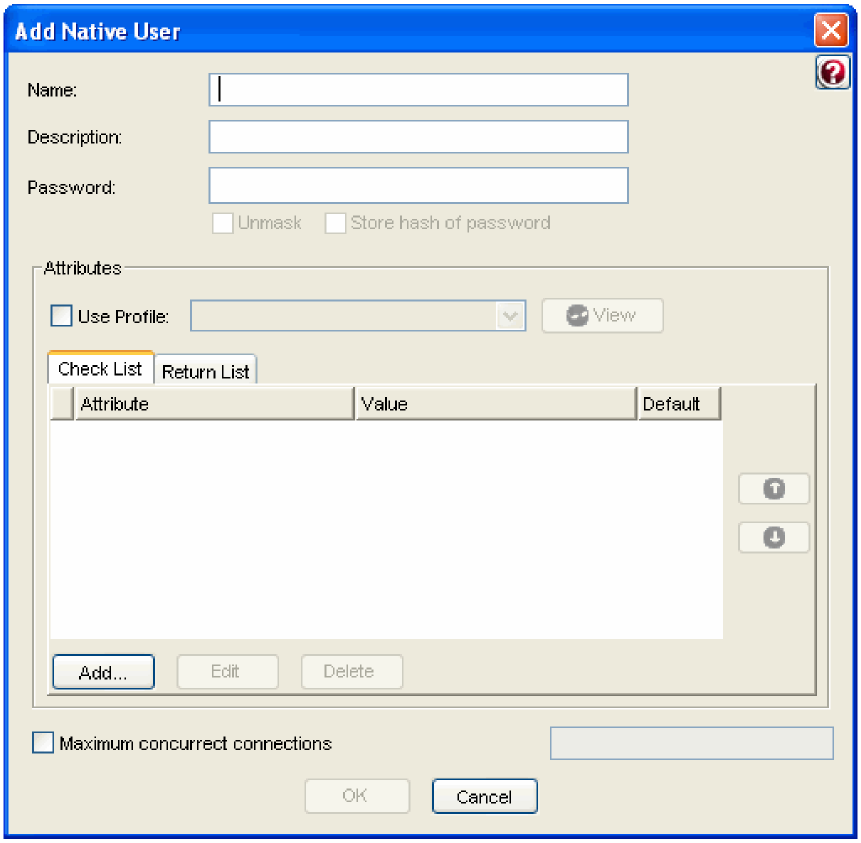
1. Choose **Users > Native** in the sidebar. The Native Users panel (Figure 43: Native Users Panel) appears.

Figure 43: Native Users Panel



2. Click **Add**.
   The Add Native User dialog appears.

Figure 44: Add Native User Dialog



3. Enter the user's login name in the **Name** field.

   Native user entries in the Steel-Belted Radius database have all-uppercase names; names are converted to all-uppercase letters when the native user entry is created, and they remain all-uppercase for the life of the entry. For example, a native username entered as **realLife1** is stored as **REALLIFE1** in the Steel-Belted Radius database.

4. Optionally, enter a description of the user in the **Description** field.

   The description you associate with a native user is not used during processing.

5. Enter the user's login password in the **Password** field.
   If you want the characters in the password (rather than asterisks) to appear as you type, click the **Unmask** check box. Note that passwords are case-sensitive: swordfish, SwordFish, and SWORDFISH are three different passwords.

6. Specify whether you want the user's password to be encrypted before it is stored.

   - If this user requires only PAP authentication and you want to **store the hash of the password** in the Steel-Belted Radius database, click the Store hash of password check box. This option allows the user to authenticate using only PAP.

   - If this user requires CHAP authentication, do not click the **Store hash of password** check box.

7. If you want to use a profile to assign checklist and return list attributes to the user, click the **Use profile** check box and use the **Profile** list to select the profile you want.

After you select a profile, you can click the **View** button to display the checklist and return list attributes in that profile.

For more information on profiles, refer to “Administering Profiles via Legacy SBR Administrator”.

8. If you want to specify checklist attributes or return list attributes for the user, click the **Checklist** tab or the **Return list** tab, and then click the **Add** button.

   Refer to “Adding a Checklist or Return List Attribute for a User” for information on how to add checklist and return list attributes.

9. After you have added the appropriate checklist and return list attributes for a user, select an attribute and use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.

10. If you want to specify the maximum number of concurrent connections this user can maintain, click the **Maximum concurrent connections** check box and enter a number in the accompanying field.

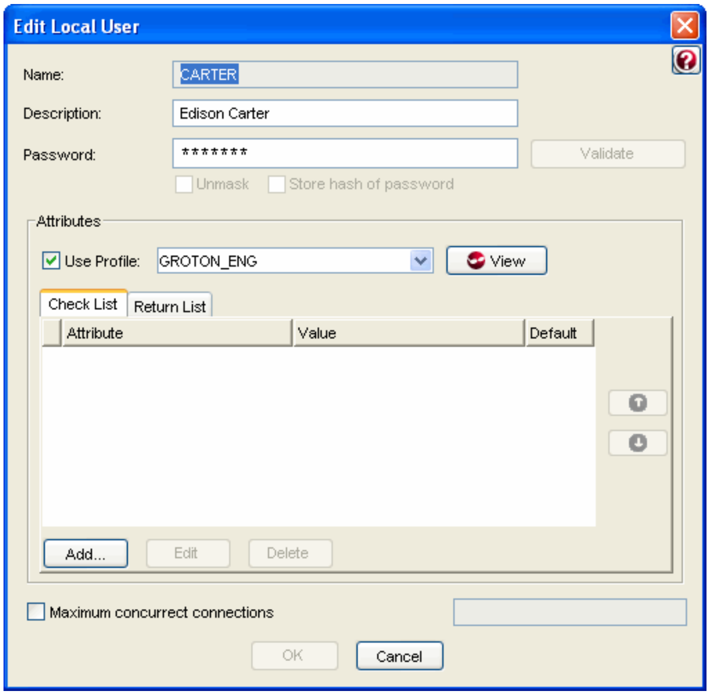
11. Click **OK**.

Editing a Native User

After you have added a user, you can modify any setting for that user except the username. To edit a native user who already exists in the Steel-Belted Radius database:

1. Choose **Users > Native** in the sidebar. The Native Users panel (Figure 45: Edit Native User Dialog) appears.
2. Select the user entry you want to edit and click the **Edit** button (or right-click an entry and choose **Edit** from the context menu).
   The Edit Native User dialog opens.

Figure 45: Edit Native User Dialog



Edit Local User

Name: CARTER

Description: Edison Carter

Password: *******

□ Unmask □ Store hash of password

Attributes

☑ Use Profile: GROTON_ENG

Check List | Return List

Attribute	Value	Default

□ Maximum concurrect connections

3. Edit the settings for the user as appropriate.

   Refer to "Adding a Native User" for information on the fields in the native user dialogs.

   You can modify any setting except the user's name. To edit a user's name, you must copy the user record to a new user entry.

4. Click **Save.**

Deleting a Native User

To delete a native user:

1. Choose **Users > Native** in the sidebar.
   The Native Users panel (Figure 43: Native Users Panel) appears.

2. Select the user entry you want to delete and click the **Delete** button (or right-click an entry and choose **Delete** from the context menu).

3. When you are asked to confirm the deletion, click **Yes**.

Adding a Checklist or Return List Attribute for a User

A checklist attribute is an item of information that must accompany a request for connection before the connection can be authenticated.

A return list attribute is an item of information that Steel-Belted Radius includes in the Access-Accept message when a connection request is approved.

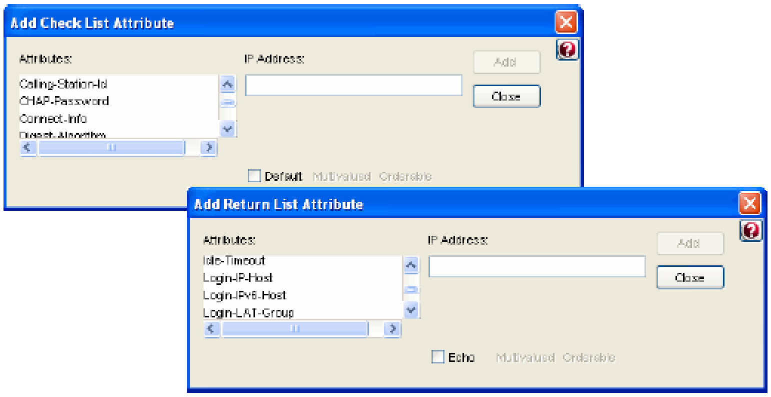
To add a checklist or return list attribute to a user's entry:

1. Open the appropriate user entry.

2. Click the **Checklist** tab or the **Return list** tab.

3. Click **Add**. The Add Checklist Attribute dialog or the Add Return List Attribute dialog opens.

Figure 46: Add Checklist Attribute and Add Return List Attribute Dialogs



4. Select the attribute you want to add from the **Attributes** list.

5. Select or enter a value for the attribute.
   The dialog changes according to the attribute you choose. Some attributes require that you enter a value, string, or IP address. Other attributes require that you choose from a fixed list of values.
   If the **Multivalued** indicator is dimmed, an attribute can have only one value. If the **Multivalued** attribute is undimmed, you can add multiple values for the attribute.

   (Checklist attributes only) To set this value to the default value for the attribute (which is useful in situations where the attribute is not included in the RADIUS request), check the **Default value** check box.

   (Return list single-valued attributes only) If you do not want to specify a particular value, but want to make sure that whatever value of the attribute appears in the RADIUS request is echoed to the client in the RADIUS response, click the **Echo** check box.

6. Click **Add** to add this attribute/value pair to the list.

7. When you are finished adding attribute/value pairs, click **Close** to return to the Add User dialog.

Setting Up Windows Domain Users

To use the Windows Domain Authentication plug-in, the RADIUS service must be run under the LocalSystem account on a Windows XP computer that is part of a domain. (The LocalSystem account is a standard authenticated domain user account in Windows.) Groups and users to be authenticated can reside in any domain within the forest, as well as in those domains outside the forest for which a trust relationship exists.

When you use Windows domain authentication, the domain name can be present in the User-Name attribute of the Access Request, which can be of the form \\domain\user, domain\user, or simply user. Additionally, the form user@domain can be used.

Prequalification Checklists

By default, when Steel-Belted Radius uses Windows domain membership to authenticate a user, it processes attributes for the first group that the user matches. The attributes consist of checklist and reply-list attributes, and checklist processing is performed to determine the user's authorization rights after authentication succeeds.

If an enterprise sets up separate Windows domain groups for different access methods (for example, one domain group for users accessing the network through a VPN and another domain group for users accessing the network through a WLAN Access Point) and then assigns users to more than one domain group (so that the users get different permissions based on what access method they use), Steel-Belted Radius can authenticate the user against the first group the user matches and process the wrong attributes for that user, causing checklist processing to fail and the user's access to be rejected.

Prequalification checklists allow a site to perform checklist processing before it authenticates a user, so that the attributes returned by every group a user belongs to can be evaluated (and the appropriate membership chosen) before authentication proceeds.

Example: CandyCorp sets up two groups (WLAN_USERS and VPN_USERS) in the CORP domain and creates access policies for each. Mary is a member of both groups; when she accesses the corporate network through a WLAN Access Point, her traffic should be tagged for a specific VLAN, and when she accesses the corporate network through a VPN, an Ascend-Data-Filt

- Without prequalification checklist processing, Steel-Belted Radius responds to Mary's connection through an Access Point by using the first domain group membership it finds (which might be VPN_USERS), authenticating Mary and returning the attributes associated with that group, and then rejecting Mary because post-authentication checklist processing fails when the group used for authentication (VPN_USERS) didn't provide the appropriate access attributes.

- With prequalification checklist processing enabled, Steel-Belted Radius responds to Mary's connection through the Access Point by running checklist processing before it authenticates Mary: Steel-Belted Radius tests each group to which Mary belongs to see if authentication and authorization will ultimately be successful. If checklist processing for a domain group fails, that group is skipped and the next group is tried; if checklist processing for all groups fails, Mary's access request is denied. If checklist processing successfully matches Mary to a domain group, authentication proceeds, and Mary's traffic is processed according to corporate policies (that is, it is tagged with the VLAN identifier appropriate for her WLAN access).

The application of prequalification checklist processing is not limited to domain groups. Prequalification checklists can be used to direct a user request to an appropriate domain user entry based on the presence of attributes in the user's request. For example, if a user's name ("ADMIN") is specified in an Access-Request and both \\CORP\ADMIN and \\LAB\ADMIN are listed in the Steel-Belted Radius database with the same password, prequalification checklist processing could be used to select the appropriate domain user object for authentication and authorization.

Note: Prequalification checklist processing can be relatively expensive in terms of processing time. Each access request might entail multiple database operations, since Steel-Belted Radius must potentially review every domain group to find one with attributes that match the user's checklist requirements.

Prequalification processing is enabled through the PrequalifyChecklist argument in the [Windows Domain] section of the winauth.aut file.

MS-CHAP Considerations

If the user is successfully authenticated, any appropriate encryption keys (obtained through either MS-CHAP or

MS-CHAP-V2) are returned to Steel-Belted Radius and the user's profile is retrieved from the Steel-Belted Radius database. To enable encryption, the appropriate attributes, such as Mppe-Send-Key and Mppe-Recv-Key, must be included in the user's profile. You do not need to prepend the username with the domain name to avoid timeout problems when dealing with large number of domains.

The Windows Domain Authentication plug-in does not support EAP pre-fetch.

Expired Domain Passwords

The Windows domain authentication method allows users to be authenticated against domain security using an expired domain password. This lets Steel-Belted Radius handle security policies that force domain passwords to be changed automatically after a certain number of days. Typically, after the password expires, at the next attempt to log in, the domain recognizes the password supplied by the user as expired. The domain then returns a special status code to its client application indicating these conditions. Typically, the user is then prompted to change his or her domain password, but the client application (for example, Microsoft Remote Access Client) must support the ability to change passwords.

When Steel-Belted Radius passes a username/password pair through to a domain for authentication, the domain can indicate to Steel-Belted Radius that the password is expired. If so, Steel-Belted Radius's default response is to issue an Access-Reject. You can configure it to respond instead with an Access-Accept.

Windows Domain Authentication Configuration

As with other authentication plug-ins, winauth.dll is configured through a single .aut file (winauth.aut). The winauth.aut file must contain [Bootstrap] and [Windows Domain] sections:

```
[Bootstrap]

LibraryName=winauth.dll

Enable=1

InitializationString=Windows   domain   authentication
```

Processing for users with expired passwords is configured in the [Windows Domain] section:

```
[Windows Domain]

AllowExpiredPasswordsForUsers = no

AllowExpiredPasswordsForGroups = no

RetryFailedAuthentications = no

AllowMachineLogin = yes

;ProfileForExpiredUsers = Profile

;ProfileForExpiredUsersInGroups = Profile

PrequalifyChecklist = no
```

Note: MS-CHAP and MS-CHAP-V2 users with expired passwords are not accepted. They might be prompted to change password if their login application supports password changing.

Adding a Domain User or Domain Group

To use domain authentication, the Steel-Belted Radius service must run on a Windows workstation or server that belongs to a domain. The Windows host running Steel-Belted Radius does not need to be a domain controller.

It is possible to authenticate against domains other than the one in which the Steel-Belted Radius service is running, provided that the other domain is trusted by the domain of the RADIUS service. The trust relationship might not be mutual; the other domain does not have to trust the RADIUS domain.

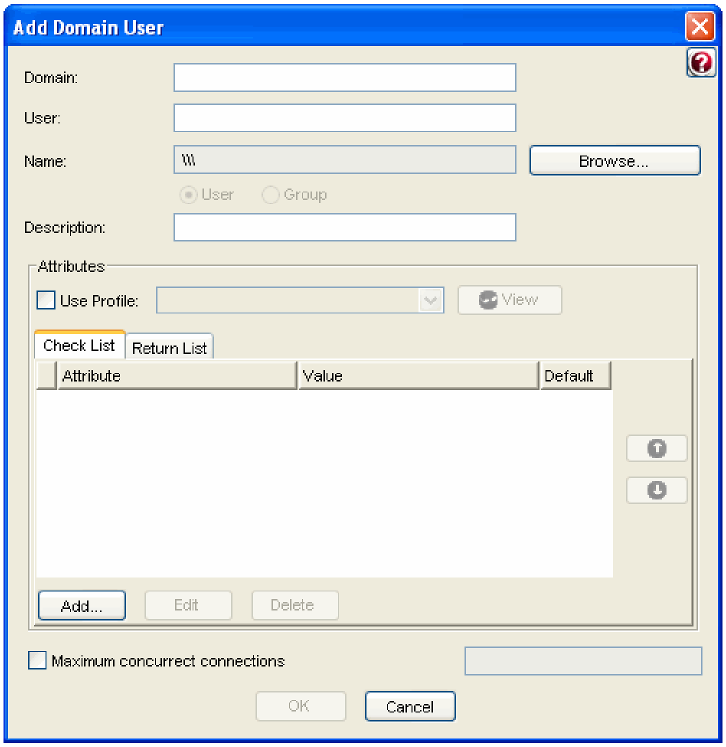
Example: An enterprise has three domains: A, B, and C, and Steel-Belted Radius is running in A. A trusts B and C trusts A. You can use Domains A and B for authentication, but not C, because A does not trust C.

You can add a Domain User entry to provide for the authentication of a specific user defined within a specific domain under Microsoft networking. For more flexibility, you can add a Domain Group, to provide for the authentication of all users that belong to a specific group defined within a specific domain.

To add a domain user or domain group:

1. Choose Users > Domain in the sidebar.
2. Click the Add button on the SBR Administrator toolbar to display the Add Domain User dialog.

Figure 47: Add Domain User Dialog



3. Specify the domain and username for the user you want to add.

   You can enter the user’s domain in the **Domain** field and the user’s login name in the **User** field.

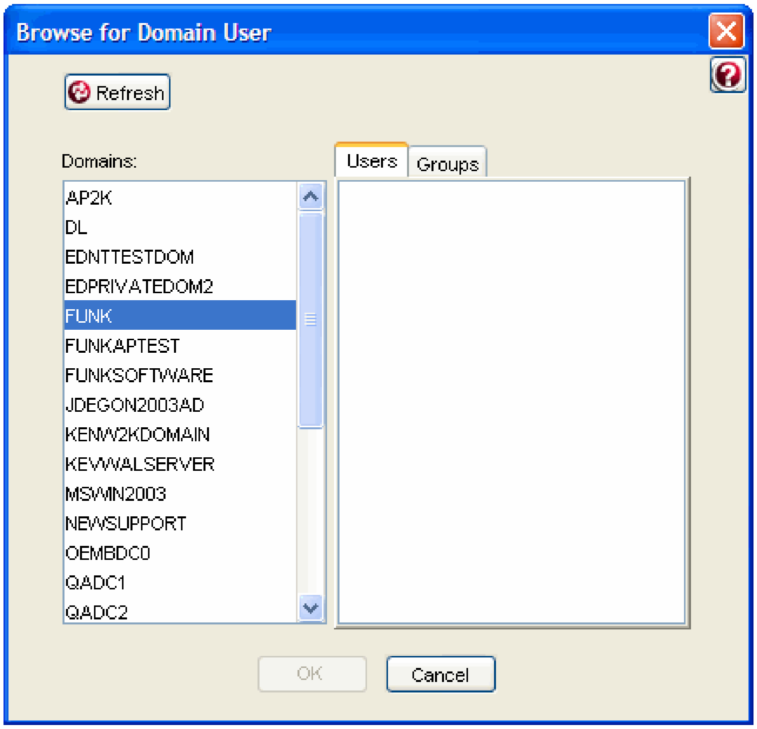
   Domain usernames must be in the format \\domain\user. Domain usernames cannot contain the following characters:

   \ / “ [ ] : | < > + = ; , ? * @

   If you want to browse for an existing user or group, click the **Browse** button. When the Browse for

Domain User dialog (Figure 48: Browse for Domain User Dialog) opens, click the name of the appropriate domain, and then click the name of the user or group in that domain you want to use. Click **OK** to finish.

Figure 48: Browse for Domain User Dialog



4. Optionally, enter a description of the domain user or group in the **Description** field.

   The description you associate with a native user is not used during processing.

5. If you want to use a profile to assign checklist and return list attributes to the user, click the **Use profile** check box and use the **Profile** list to select the profile you want.

   After you select a profile, you can click the **View** button to display the checklist and return list attributes in that profile.

   For more information on Profiles, refer to "Administering Proxy RADIUS via Legacy SBR Administrator".

6. If you want to specify checklist attributes or return list attributes for the domain user, click the **Checklist** tab or the **Return list** tab, and then click the **Add** button.

   Refer to "Adding a Checklist or Return List Attribute for a User" for information on how to add checklist and return list atttibutes.

7. After you have added the appropriate checklist and return list attributes for a user, select an attribute and use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.

8. If you want to specify the maximum number of concurrent connections this user can maintain, click the Maximum concurrent connections check box and enter a number in the accompanying field.

9. Click **OK.**

Setting Up SecurID Users

Usernames are case-sensitive. The case in which names are recorded depends on whether usernames are being stored in a local or external database. Usernames stored in an external database (UNIX, RSA SecurID, TACACS+) retain their case as stored in that database.

Adding a SecurID User

You can configure Steel-Belted Radius to use RSA SecurID authentication for your users by setting up communication between the RSA server and the RADIUS server (described in "Configuring SecurID Authentication"), and then adding SecurID users to the Steel-Belted Radius database using the instructions that follow.

Steel-Belted Radius attempts SecurID authentication only on usernames that match a SecurID entry in its User database. Steel-Belted Radius offers four types of SecurID entry, each providing a different matching rule:

- You can enter the name of a specific user.

  For example, you might create a SecurID user entry for the specific user George. This tells Steel-Belted Radius that SecurID can be used as an authentication method when an authentication request is received for username George. If username George is authenticated, the attributes of the user entry apply.

- You can enter a prefix.

  For example, you might create a SecurID user entry for the prefix sales$. This tells Steel-Belted Radius that SecurID can be used as an authentication method when an authentication request is received for a username such as sales$Harry or sales$Cynthia. Using a prefix lets you group multiple SecurID users into a single user entry instead of creating a separate entry for each SecurID user. If the sales$ user is authenticated, the attributes of the user entry apply.

  Note: Only the part of the username after the prefix (Harry or Cynthia in the example above) is sent to the RSA SecurID server.

  You can use different settings for different groups.

  Note: The user must type in the prefix as part of the username when dialing in and requesting a connection.

- You can enter a suffix.

  A suffix works like a prefix, but appears at the end of the username; for example, if the suffix were !sales, you might have usernames such as Harry!sales or Cynthia!sales.

- You can create an entry for Any user.

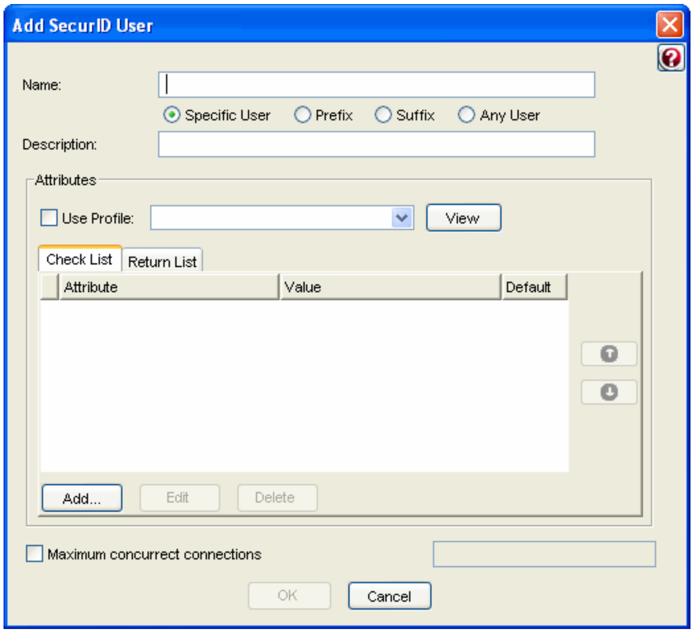
  This creates a single user entry named <ANY> that matches any username to be authenticated. SecurID can be used as an authentication method for any username, and, if successful, the attributes of the <ANY> entry apply.

  The <ANY> entry makes sense if a single set of attributes apply to all your SecurID users and if you want to make SecurID either the only authentication method used or the authentication method of last resort if other authentication methods fail.

To add a SecurID user entry:

1. Choose **Users > SecurID** in the sidebar.
2. Click the **Add** button on the SBR Administrator toolbar to display the Add SecurID User dialog.

Figure 49: Add SecurID User Dialog



3. Enter the specific username, a prefix, or a suffix in the **Name** field.

   Select the user type: **Specific user, Prefix, Suffix,** or **Any user.**

   Optionally, enter a description of the user in the **Description** field.

4. If you want to use a profile to assign checklist and return list attributes to the user, click the **Use profile** check box and use the Profile list to select the **profile** you want.

   After you select a profile, you can click the **View** button to display the checklist and return list attributes in that profile.

5. If you want to specify checklist attributes or return list attributes for the user, click the **Checklist** tab or the **Return list** tab, and then click the **Add** button.

   Refer to "Adding a Checklist or Return List Attribute for a User" for information on how to add checklist and return list attributes.

6. After you have added the appropriate checklist and return list attributes for a user, use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.

7. If you want to specify the maximum number of concurrent connections this user can maintain, click the **Maximum concurrent connections** check box and enter a number in the accompanying field.

8. Click OK.

   Each new suffix or prefix entry that you add appears in the Users dialog with the username represented by the string USERNAME; for example, !USERNAME or SALES<USERNAME>.

Setting Up TACACS+ Users

You can configure Steel-Belted Radius to authenticate your users by querying a TACACS+ server.

ⓘ Note: Before you add TACACS+ users, you must configure communication between the Steel-Belted Radius server and the TACACS+ server by editing the tacplus.ini file. For more information on the tacplus.ini file, refer to the Steel-Belted Radius Reference Guide.

Steel-Belted Radius attempts TACACS+ authentication only on usernames that match a TACACS+ entry in its user database. Each type of TACACS+ entry specifies a different matching rule:

- You can enter the name of a specific user.

  For example, you might create a TACACS+ user entry for the specific user George. This tells Steel-Belted Radius that when an authentication request is received for username George, TACACS+ can be used as an authentication method and, if successful, the attributes of this user entry apply.

- You can enter a prefix.

  For example, you might create a TACACS+ user entry for the prefix sales$. This tells Steel-Belted Radius that when an authentication request is received for a username such as sales$Harry or sales$Cynthia, TACACS+ can be used as an authentication method and, if successful, the attributes of this user entry apply.

  ⓘ Note: Only the part of the username after the prefix (Harry or Cynthia in the example above) is sent to the TACACS+ server.

  Using a prefix lets you group multiple TACACS+ into a single user entry instead of creating a separate entry for each TACACS+ user. You can use different settings for different groups.

  ⓘ Note: The user must type in the prefix as part of the username he or she is using to dial in and request a connection.

- You can enter a suffix.

  A suffix works like a prefix, but appears at the end of the username; for example, if the suffix were !sales, you might have usernames such as Harry!sales or Cynthia!sales.

- You can create an entry for Any user.

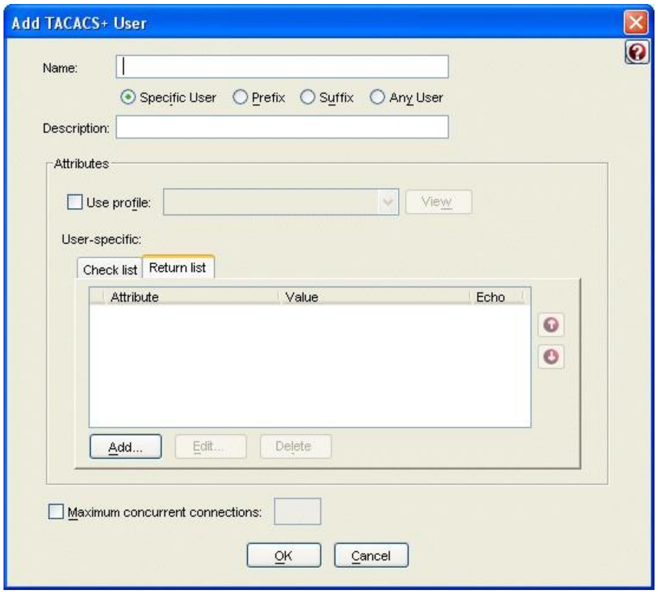
  This creates a single user entry named <ANY> that matches any username to be authenticated. TACACS+ can be used as an authentication method for any username, and, if successful, the attributes of the <ANY> entry apply. The <ANY> entry makes sense if a single set of attributes apply to all your TACACS+ users and if you want to make TACACS+ either the only authentication method used or the authentication method of last resort if other authentication methods fail.

To add a TACACS+ user:

1. Choose Users > TACACS+ in the sidebar.

2. Click the Add button on the SBR Administrator toolbar to display the Add TACACS+ User dialog.

Figure 50: Add TACACS+ User Dialog



3. Enter the specific username, a prefix, or a suffix in the Name field.

4. Select the user type: Specific user, Prefix, Suffix, or Any user.

5. Optionally, enter a description of the user in the Description field.

6. If you want to use a profile to assign checklist and return list attributes to the user, click the Use profile check box and use the Profile list to select the profile you want.

   After you select a profile, you can click the View button to display the checklist and return list attributes in that profile.

7. If you want to specify checklist attributes or return list attributes for the user, click the Checklist tab or the Return list tab, and then click the Add button.

   Refer to "Adding a Checklist or Return List Attribute for a User" for information on how to add checklist and return list attributes.

8. After you have added the appropriate checklist and return list attributes for a user, use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.

9. If you want to specify the maximum number of concurrent connections this user can maintain, click the Maximum concurrent connections check box and enter a number in the accompanying field.

10.Click OK.

Each new suffix or prefix entry that you add appears in the Users dialog with the username represented by the string USERNAME; for example, !USERNAME or SALES<USERNAME>.

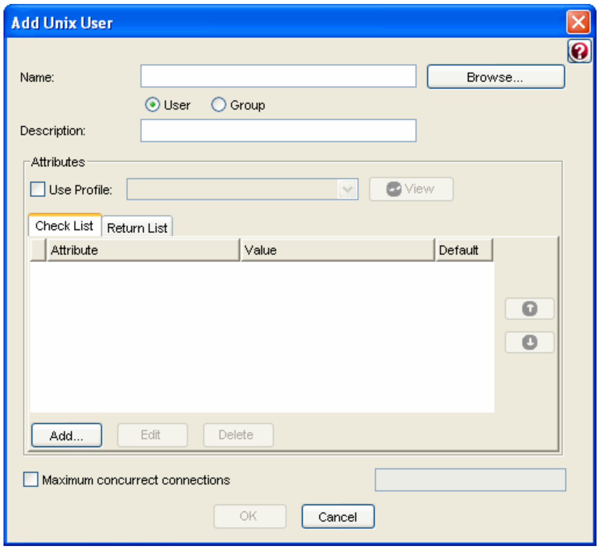
Setting Up UNIX Users

You can add a UNIX user entry to provide for the authentication of a specific user defined on a Linux server. For more flexibility, you can add a UNIX group to provide for the authentication of all users that belong to a specific group defined on the server.

To add a UNIX user or group:

1. Choose **Users > UNIX** in the sidebar.
2. Click the **Add** button on the SBR Administrator toolbar to display the Add UNIX User dialog.

Figure 51: Add UNIX User Dialog



3. Click the **Browse** button and select a user or group from the list. Click **OK**.
4. Optionally, enter a description of the user in the **Description** field.
5. If you want to use a profile to assign checklist and return list attributes to the user, click the **Use profile** check box and use the **Profile** list to select the profile you want.

   After you select a profile, you can click the **View** button to display the checklist and return list attributes in that profile.
6. If you want to specify checklist attributes or return list attributes for the user, click the **Checklist** tab or the **Return list** tab, and then click the **Add** button.

   Refer to "Adding a Checklist or Return List Attribute for a User" for information on how to add checklist and return list attributes.
7. After you have added the appropriate checklist and return list attributes for a user, use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.
8. If you want to specify the maximum number of concurrent connections this user can maintain, click the **Maximum concurrent connections** check box and enter a number in the accompanying field.
9. Click OK.

Editing User Settings

This section describes fields that you can set for any user entry, regardless of user type. For more information, see "User Attribute Lists" and "About Profiles".

Selecting a Profile

To select a profile for a user:

1. Open the the appropriate user panel by clicking a **User** > entry in the sidebar.
2. Select the user whose entry you want to modify.
3. Click **Edit** (or double-click the user entry).
4. Click the **Use Profile** check box.
5. Select the list to select the profile you want to use.

   To display the settings associated with the selected profile, click the **View** button.
6. When you are finished, click Save.

Setting Attribute Values

To change the value of an attribute already in the checklist or return list for a user entry:

1. Click the Checklist tab or Return List tab.
2. Select the attribute whose value you want to change.
3. Click Edit or double-click the attribute.
4. When the Change dialog opens, enter or select the new value.

   Depending on the attribute, you can enter a new value or select a value from a list. For some attributes, Steel-Belted Radius retrieves the value from the server and you cannot enter a value in this dialog.
5. Click **OK.**

Removing Attribute/Value Pairs

To remove an attribute/value pair already in the checklist or return list for a User entry:

1. Click the **Checklist** tab or **Return List** tab.
2. Select the attribute/value pair you'd like to remove.
3. Click Delete.

Reordering Attributes

Certain attributes are multi-valued and orderable; that is, the attribute/value pair can appear more than once in a RADIUS response, and the order in which the attribute/value pairs appear is significant. To reorder attributes in a User entry:

1. Click the **Checklist** tab or **Return List** tab.

2. Highlight an attribute/value pair in the list.

3. Click the Up or Down arrow to move the selected attribute within the list.

    - The Up arrow moves the selected attribute/value up in the list. If the attribute is not orderable, or if the selected item already the first value for this attribute, the button is disabled.

    - The Down arrow moves the selected attribute/value down in the list. If the attribute is not orderable, or if the selected item is already the last value for this attribute, the button is disabled.

Changing Attributes Inherited from a Profile

Checklist and return list attributes can be specified for a user, or they can be inherited from a profile associated with a user. Attributes inherited from a profile are overridden by attributes assigned to a specific user.

Concurrent Connection Limits

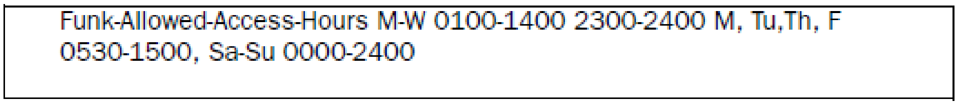
A maximum number of open connections can be set for each user entry by checking the **Maximum concurrent connections** check box and entering a number in the accompanying field. When the user requests access, the user can be authenticated using the given authentication method only if fewer than this number of connections are currently open for this user.

Allowed Access Hours

The user's allowed access hours can be specified by adding the Funk-Allowed-Access-Hours attribute to the user's checklist.

Funk-Allowed-Access-Hours is a variable-length string that identifies time periods in a 7-day week of 24-hour days. This string consists of one or more day specifiers (each of which can list one or more days and/or ranges of days) followed by one or more ranges of 24-hour times, in minutes.

Figure 52: Sample Funk-Allowed-Access-Hours Attribute Value



```
Funk-Allowed-Access-Hours M-W 0100-1400 2300-2400 M, Tu,Th, F
0530-1500, Sa-Su 0000-2400
```

The syntax rules for Funk-Allowed-Access-Hours are as follows:

- Time ranges can be inclusive (1000–1100 allows access only between 10 a.m. and 11 a.m.) or exclusive (1100–1000 allows access any time except between 10 a.m. and 11 a.m.).

- Day specifiers, and ranges of days and times can be separated by commas or spaces; ranges of days or times are indicated by hyphens (m–w or 0239–1459).

- Days can be specified by the minimum number of case-insensitive letters necessary to distinguish them (Su, M, Tu, W, Th, F, Sa) and can wrap around the end of the week (Sa-Su).

- At least one time period is required for each day; that is, each day, list, or range of days must be followed by one or more ranges of times.

- Times are specified using four digits, with leading zeroes where needed (0001 for 12:01 a.m., 0630 for 6:30 a.m., and so forth).

When assigned to a user's checklist, the Funk-Allowed-Access-Hours value in Figure 52: Sample Funk-Allowed-

Access-Hours Attribute Value allows the user access during the following time periods:

- 1 a.m. to 2 p.m. and 11:00 p.m. to midnight, Monday through Wednesday
- 5:30 a.m. to 3:00 p.m. Monday, Tuesday, Thursday, and Friday
- Any time Saturday or Sunday

The total access times Monday are 1 a.m. to 3:00 p.m. and 11:00 p.m. to midnight

If the user attempts access on Sunday at 11:30 p.m., access would be allowed and a Session-Timeout attribute specifying a value of 1800 seconds (30 minutes, until midnight Sunday, when the access period ends) would be returned. However, if the user's return list includes a Session-Timeout with a value less than 1800, this lesser value would be returned.

Deleting a User

To delete a user:

1. Open the the appropriate user panel by clicking a User > entry in the sidebar.
2. Select the user you want to delete.
3. Click the Delete button from the SBR Administrator toolbar (or right-click the user entry and choose Delete from the context menu).
4. When you are prompted to confirm the deletion, click Yes.

Chapter 10
Administering Users via WebGUI

This chapter describes how to add users to the Steel-Belted Radius database via WebGUI.

User Files

The following files establish settings for setting up users. For more information about these files, refer to the Steel-Belted Radius Reference Guide.

Table 22: User Account Files

File Name	Function
lockout.ini	Configures settings for when Steel-Belted Radius should lock user accounts after repeated failed login attempts.
redirect.ini	Configures settings for when Steel-Belted Radius should redirect users after repeated failed login attempts.
radius.ini	Specifies (among other things) the settings relating to RSA SecurID support in Steel-Belted Radius.
securid.ini	Specifies the prompt strings returned to SecurID users during login and authentication.
tacplus.ini	Specifies the name of the TACACS+ server and the shared secret used to validate communication between the Steel-Belted Radius server and the TACACS+ server.

Users Pages

The Users entry in the menubar has as many as five sub-entries, as described in **Table 23**. Each user entry in the Steel-Belted Radius database identifies one method by which the server can authenticate a specific user.

Table 23: User Pages

User Pages	Function	Available On Server Running OS
Local	Lists the native users in the local Steel-Belted Radius database. You must display the Native Usler page to add, edit and delete native users.	Windows Linux
Domain	Lists the users authenticated using WindowsDomain authentication. You must display the Domain page to add, edit, and delete Domain users.	Windows

User Pages	Function	Available On Server Running OS
SecurID	Lists the users authenticated using RSA SecurID authentication. You must display the **SecurID** page to add, edit, and delete SecurID users	Windows Linux
TACACS+	Lists the users authenticated using TACACS+. You must display the **TACACS+** page to add, edit, and delete TACACS+ users.	Windows Linux
UNIX	Lists the users and groups authenticated using UNIX authentication. You must display the **UNIX** page to add, edit, and delete UNIX users.	Linux

Note: You can populate the user database for Steel-Belted Radius by entering information in the Users page or by importing data from other servers. For more information on importing user information, see Appendix E.

Setting Up Native Users

Native user entries require you to enter the user's name and password into the Steel-Belted Radius database. For all other types of user entry, the server relies on another database to confirm the user's password.

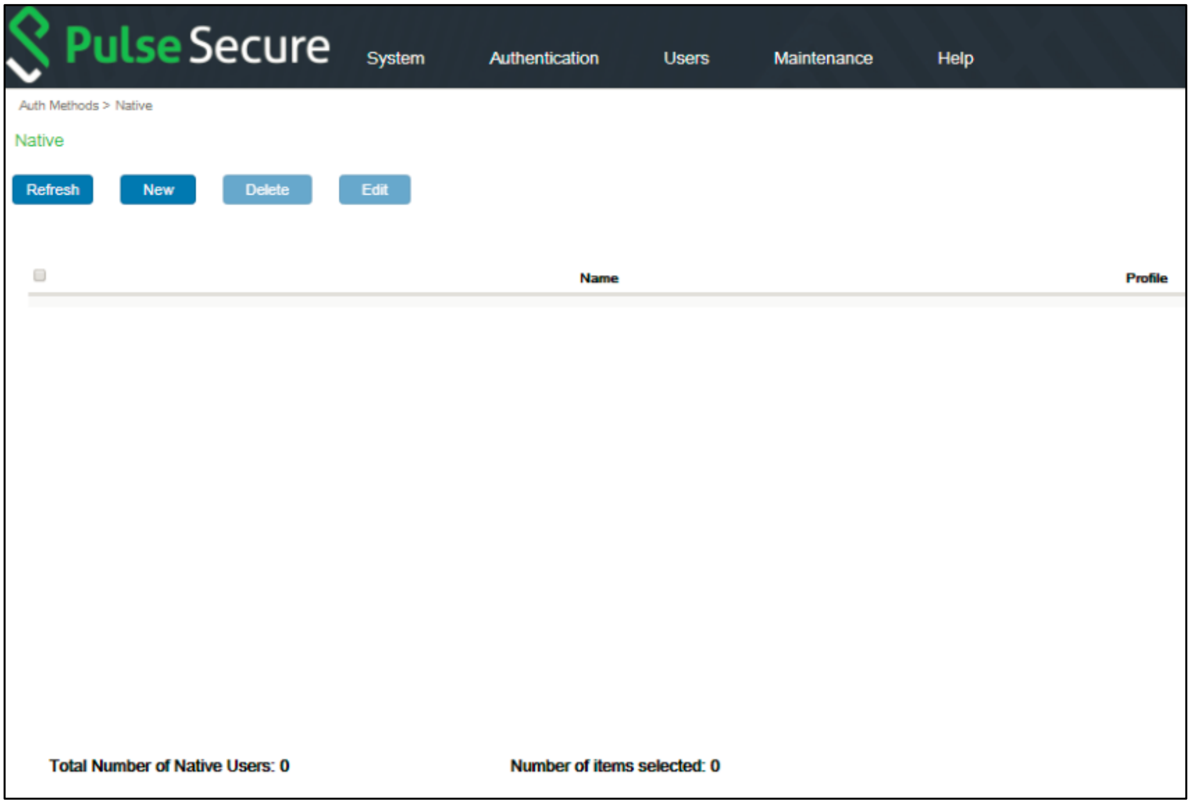
Note: You must define a native user entry for every user who requires remote access to a Windows network. For example, you can accommodate Linux or Macintosh users by adding them as native users.

Adding a Native User

To add a native user to the Steel-Belted Radius database:

1. Choose **Users > Auth Methods > Native**. The Native Users page (Figure 53: Native Users Page) appears.

Figure 53: Native Users Page

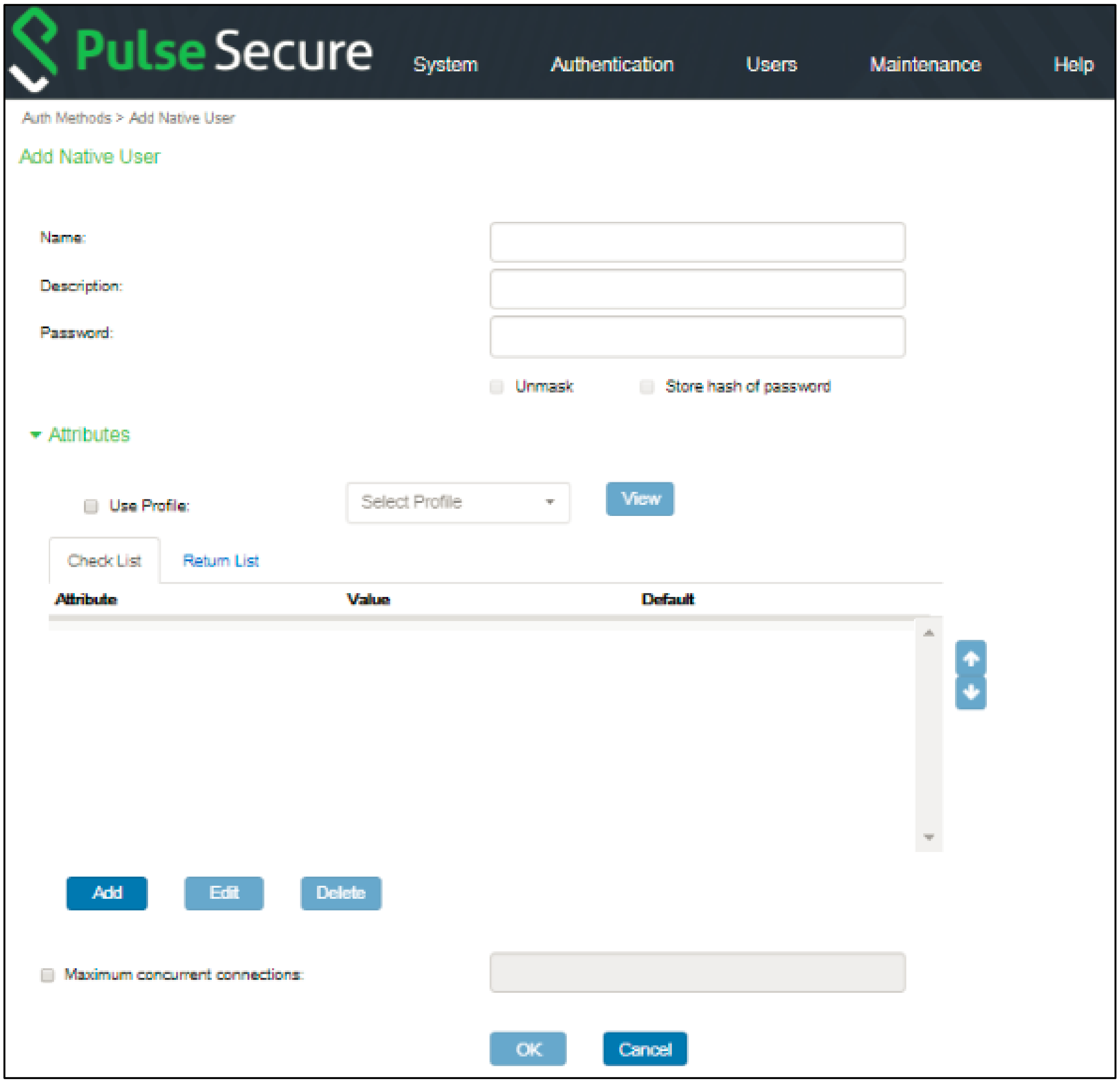


Total Number of Native Users: 0

Number of items selected: 0

2. Click New.
   The Add Native User page appears.

Figure 54: Add Native User Page



3. Enter the user's login name in the **Name** field.

   Native user entries in the Steel-Belted Radius database have all-uppercase names; names are converted to all-uppercase letters when the native user entry is created, and they remain all-uppercase for the life of the entry. For example, a native username entered as **realLife1** is stored as **REALLIFE1** in the Steel-Belted Radius database.
4. Optionally, enter a description of the user in the **Description** field.

   The description you associate with a native user is not used during processing.

5. Enter the user's login password in the **Password** field.
   If you want the characters in the password (rather than asterisks) to appear as you type, click the **Unmask** check box. Note that passwords are case-sensitive: swordfish, SwordFish, and SWORDFISH are three different passwords.
6. Specify whether you want the user's password to be encrypted before it is stored.

   - If this user requires only PAP authentication and you want to **store the hash of the password** in the Steel-Belted Radius database, click the Store hash of password check box. This option allows the user to authenticate using only PAP.

- If this user requires CHAP authentication, do not click the **Store hash of password** check box.

7. If you want to use a profile to assign checklist and return list attributes to the user, click the **Use profile** check box and use the **Profile** list to select the profile you want.

   After you select a profile, you can click the **View** button to display the checklist and return list attributes in that profile.

   For more information on profiles, refer to "Administering Profiles via WebGUI."

8. If you want to specify checklist attributes or return list attributes for the user, click the **Checklist** tab or the **Return list** tab, and then click the **Add** button.

   Refer to Adding a Checklist or Return List Attribute for a User for information on how to add checklist and return list attributes.

9. After you have added the appropriate checklist and return list attributes for a user, select an attribute and use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.

10. If you want to specify the maximum number of concurrent connections this user can maintain, click the **Maximum concurrent connections** check box and enter a number in the accompanying field.

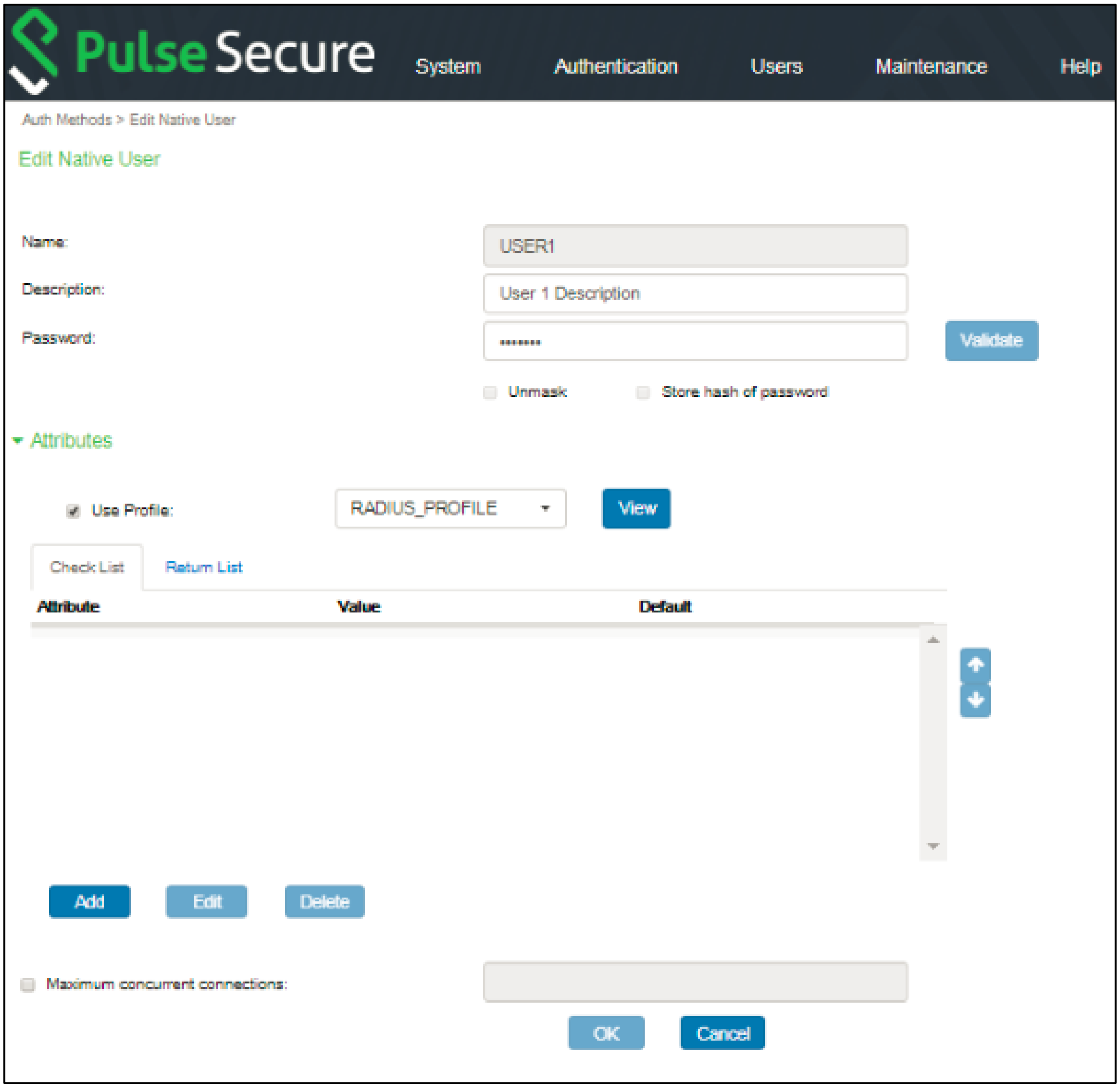
11. Click **OK**.

Editing a Native User

After you have added a user, you can modify any setting for that user except the username. To edit a native user who already exists in the Steel-Belted Radius database:

1. Choose **Users > Auth Methods > Native**. The Native Users page (Figure 55: Edit Native User Page) appears.
2. Select the user entry you want to edit and click the **Edit.**
   The Edit Native User page opens.

Figure 55: Edit Native User Page



Auth Methods > Edit Native User

Edit Native User

Name: USER1

Description: User 1 Description

Password: *******

□ Unmask □ Store hash of password

Attributes

☑ Use Profile: RADIUS_PROFILE

Check List | Return List

Attribute	Value	Default

□ Maximum concurrent connections:

3. Edit the settings for the user as appropriate.
   Refer to “Adding a Native User” for more information on the fields in the native user pages. You can modify any setting except the user’s name.

4. Click **OK.**

Deleting a Native User

To delete a native user:

1. Choose **Users > Auth Methods > Native.**
   The Native Users page (Figure 53: Native Users Page) appears.

2. Select the user entry you want to delete and click the **Delete** button.

3. When you are asked to confirm the deletion, click **Yes**.

Adding a Checklist or Return List Attribute for a User

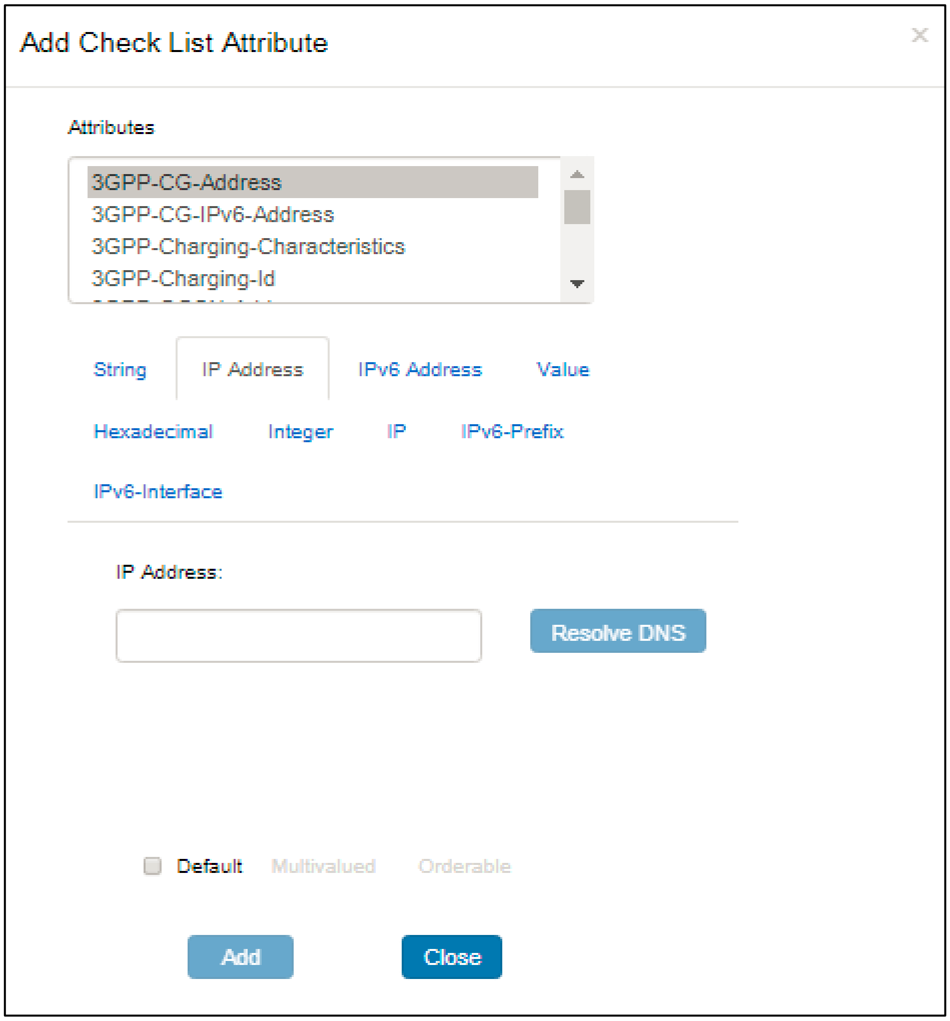
A checklist attribute is an item of information that must accompany a request for connection before the connection can be authenticated.

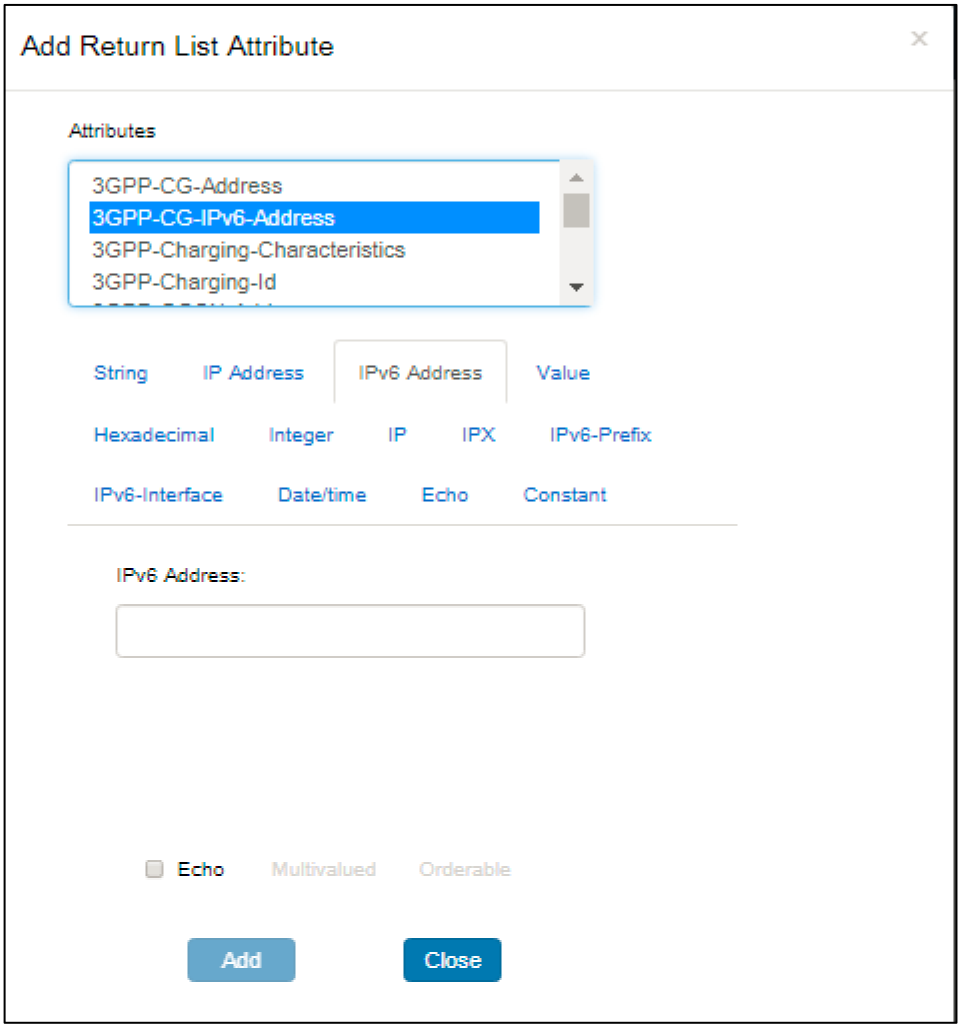
A return list attribute is an item of information that Steel-Belted Radius includes in the Access-Accept message when a connection request is approved.

To add a checklist or return list attribute to a user's entry:

1. Open the appropriate user entry.
2. Click the **Checklist** tab or the **Return list** tab.
3. Click **Add**. The Add Checklist Attribute page or the Add Return List Attribute page opens.

Figure 56: Add Checklist Attribute and Add Return List Attribute Pages





4. Select the attribute you want to add from the **Attributes** list.

5. Select or enter a value for the attribute.
   The page changes according to the attribute you choose. Some attributes require that you enter a value, string, or IP address. Other attributes require that you choose from a fixed list of values.
   If the **Multivalued** indicator is dimmed, an attribute can have only one value. If the **Multivalued** attribute is undimmed, you can add multiple values for the attribute.

   (Checklist attributes only) To set this value to the default value for the attribute (which is useful in situations where the attribute is not included in the RADIUS request), check the **Default value** check box.

   (Return list single-valued attributes only) If you do not want to specify a particular value, but want to make sure that whatever value of the attribute appears in the RADIUS request is echoed to the client in the RADIUS response, click the **Echo** check box.

6. Click **Add** to add this attribute/value pair to the list.

7. When you are finished adding attribute/value pairs, click **Close** to return to the Add User page.

Setting Up Windows Domain Users

To use the Windows Domain Authentication plug-in, the RADIUS service must be run under the LocalSystem account on a Windows computer that is part of a domain. (The LocalSystem account is a standard authenticated domain user account in Windows.) Groups and users to be authenticated can reside in any domain within the forest, as well as in those domains outside the forest for which a trust relationship exists.

When you use Windows domain authentication, the domain name can be present in the User-Name attribute of the Access Request, which can be of the form \\domain\user, domain\user, or simply user. Additionally, the form user@domain can be used.

Prequalification Checklists

By default, when Steel-Belted Radius uses Windows domain membership to authenticate a user, it processes attributes for the first group that the user matches. The attributes consist of checklist and reply-list attributes, and checklist processing is performed to determine the user's authorization rights after authentication succeeds.

If an enterprise sets up separate Windows domain groups for different access methods (for example, one domain group for users accessing the network through a VPN and another domain group for users accessing the network through a WLAN Access Point) and then assigns users to more than one domain group (so that the users get different permissions based on what access method they use), Steel-Belted Radius can authenticate the user against the first group the user matches and process the wrong attributes for that user, causing checklist processing to fail and the user's access to be rejected.

Prequalification checklists allow a site to perform checklist processing before it authenticates a user, so that the attributes returned by every group a user belongs to can be evaluated (and the appropriate membership chosen) before authentication proceeds.

Example: CandyCorp sets up two groups (WLAN_USERS and VPN_USERS) in the CORP domain and creates access policies for each. Mary is a member of both groups; when she accesses the corporate network through a WLAN Access Point, her traffic should be tagged for a specific VLAN, and when she accesses the corporate network through a VPN, an Ascend-Data-Filt

- Without prequalification checklist processing, Steel-Belted Radius responds to Mary's connection through an Access Point by using the first domain group membership it finds (which might be VPN_USERS), authenticating Mary and returning the attributes associated with that group, and then rejecting Mary because post-authentication checklist processing fails when the group used for authentication (VPN_USERS) didn't provide the appropriate access attributes.

- With prequalification checklist processing enabled, Steel-Belted Radius responds to Mary's connection through the Access Point by running checklist processing before it authenticates Mary: Steel-Belted Radius tests each group to which Mary belongs to see if authentication and authorization will ultimately be successful. If checklist processing for a domain group fails, that group is skipped and the next group is tried; if checklist processing for all groups fails, Mary's access request is denied. If checklist processing successfully matches Mary to a domain group, authentication proceeds, and Mary's traffic is processed according to corporate policies (that is, it is tagged with the VLAN identifier appropriate for her WLAN access).

The application of prequalification checklist processing is not limited to domain groups. Prequalification checklists can be used to direct a user request to an appropriate domain user entry based on the presence of attributes in the user's request. For example, if a user's name ("ADMIN") is specified in an Access-Request and both \\CORP\ADMIN and \\LAB\ADMIN are listed in the Steel-Belted Radius database with the same password, prequalification checklist processing could be used to select the appropriate domain user object for authentication and authorization.

Note: Prequalification checklist processing can be relatively expensive in terms of processing time. Each access request might entail multiple database operations, since Steel-Belted Radius must potentially review every domain group to find one with attributes that match the user's checklist requirements.

Prequalification processing is enabled through the PrequalifyChecklist argument in the [Windows Domain] section of the winauth.aut file.

MS-CHAP Considerations

If the user is successfully authenticated, any appropriate encryption keys (obtained through either MS-CHAP or MS-CHAP-V2) are returned to Steel-Belted Radius and the user's profile is retrieved from the Steel-Belted Radius database. To enable encryption, the appropriate attributes, such as Mppe-Send-Key and Mppe-Recv-Key, must be included in the user's profile. You do not need to prepend the username with the domain name to avoid timeout problems when dealing with large number of domains.

The Windows Domain Authentication plug-in does not support EAP pre-fetch.

Expired Domain Passwords

The Windows domain authentication method allows users to be authenticated against domain security using an expired domain password. This lets Steel-Belted Radius handle security policies that force domain passwords to be changed automatically after a certain number of days. Typically, after the password expires, at the next attempt to log in, the domain recognizes the password supplied by the user as expired. The domain then returns a special status code to its client application indicating these conditions. Typically, the user is then prompted to change his or her domain password, but the client application (for example, Microsoft Remote Access Client) must support the ability to change passwords.

When Steel-Belted Radius passes a username/password pair through to a domain for authentication, the domain can indicate to Steel-Belted Radius that the password is expired. If so, Steel-Belted Radius's default response is to issue an Access-Reject. You can configure it to respond instead with an Access-Accept.

Windows Domain Authentication Configuration

As with other authentication plug-ins, winauth.dll is configured through a single .aut file (winauth.aut). The winauth.aut file must contain [Bootstrap] and [Windows Domain] sections:

```
[Bootstrap]

LibraryName=winauth.dll

Enable=1

InitializationString=Windows domain authentication
```

Processing for users with expired passwords is configured in the [Windows Domain] section:

```
[Windows Domain]

AllowExpiredPasswordsForUsers = no

AllowExpiredPasswordsForGroups = no

RetryFailedAuthentications = no
```

```
AllowMachineLogin = yes

;ProfileForExpiredUsers = Profile

;ProfileForExpiredUsersInGroups = Profile

PrequalifyChecklist = no
```

Note: MS-CHAP and MS-CHAP-V2 users with expired passwords are not accepted. They might be prompted to change password if their login application supports password changing.

Adding a Domain User or Domain Group

To use domain authentication, the Steel-Belted Radius service must run on a Windows workstation or server that belongs to a domain. The Windows host running Steel-Belted Radius does not need to be a domain controller.

It is possible to authenticate against domains other than the one in which the Steel-Belted Radius service is running, provided that the other domain is trusted by the domain of the RADIUS service. The trust relationship might not be mutual; the other domain does not have to trust the RADIUS domain.

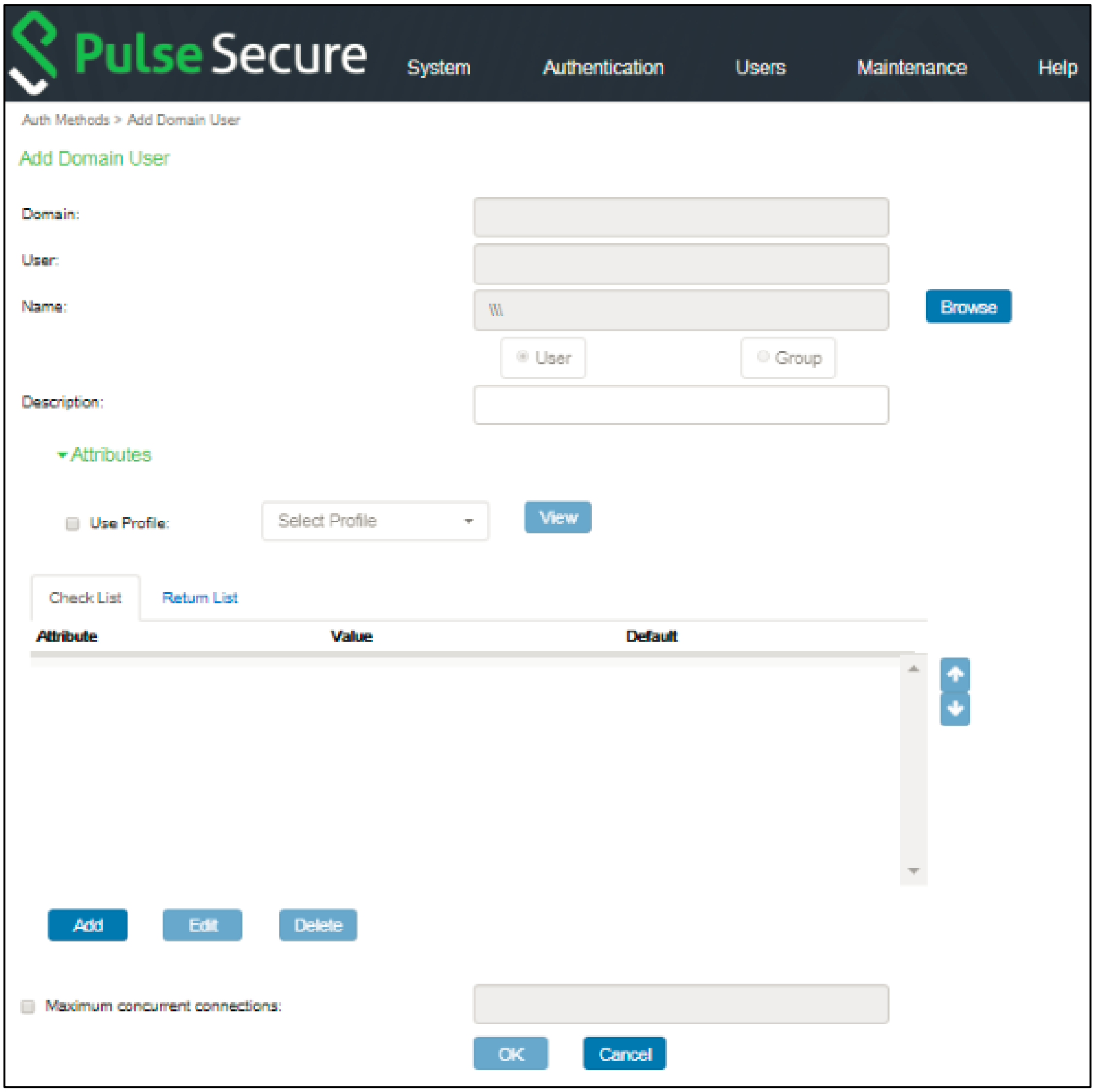
Example: An enterprise has three domains: A, B, and C, and Steel-Belted Radius is running in A. A trusts B and C trusts A. You can use Domains A and B for authentication, but not C, because A does not trust C.

You can add a Domain User entry to provide for the authentication of a specific user defined within a specific domain under Microsoft networking. For more flexibility, you can add a Domain Group, to provide for the authentication of all users that belong to a specific group defined within a specific domain.

To add a domain user or domain group:

1. Choose Users > Auth Methods > Domain.
2. Click the New button on the SBR Administrator toolbar to display the Add Domain User page.

Figure 57: Add Domain User Page



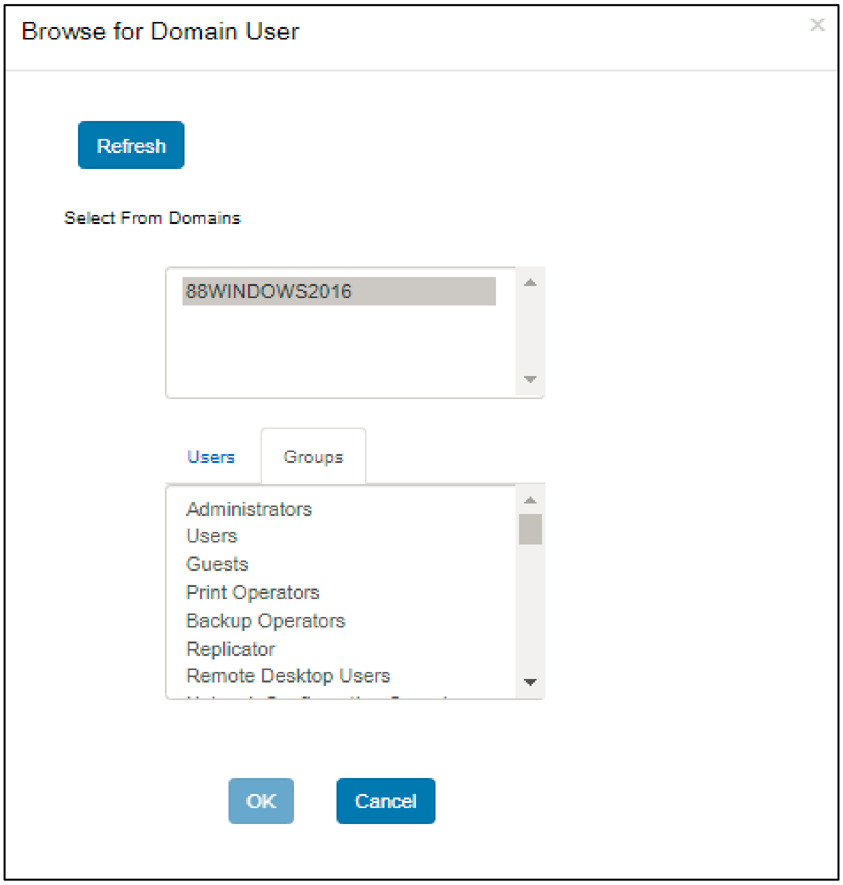
3. Specify the domain and username for the user you want to add.

Domain usernames must be in the format \\domain\user. Domain usernames cannot contain the following characters:

\ / “ [ ] : | < > + = ; , ? * @

If you want to browse for an existing user or group, click the **Browse** button. When the Browse for Domain User page (Figure 58: Browse for Domain User) opens, click the name of the appropriate domain, and then click the name of the user or group in that domain you want to use. Click **OK** to finish.

Figure 58: Browse for Domain User Page



4. Optionally, enter a description of the domain user or group in the **Description** field.

   The description you associate with a native user is not used during processing.

5. If you want to use a profile to assign checklist and return list attributes to the user, click the **Use profile** check box and use the **Profile** list to select the profile you want.

   After you select a profile, you can click the **View** button to display the checklist and return list attributes in that profile.

   For more information on profiles, refer to "Administering Profiles via WebGUI."

6. If you want to specify checklist attributes or return list attributes for the domain user, click the **Checklist** tab or the **Return list** tab, and then click the **Add** button.

   Refer to "Adding a Checklist or Return List Attribute for a User" for information on how to add checklist and return list atttibutes.

7. After you have added the appropriate checklist and return list attributes for a user, select an attribute and use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.

8. If you want to specify the maximum number of concurrent connections this user can maintain, click the Maximum concurrent connections check box and enter a number in the accompanying field.

9. Click **OK.**

Setting Up SecurID Users

Usernames are case-sensitive. The case in which names are recorded depends on whether usernames are being stored in a local or external database. Usernames stored in an external database (UNIX, RSA SecurID, TACACS+) retain their case as stored in that database.

Adding a SecurID User

You can configure Steel-Belted Radius to use RSA SecurID authentication for your users by setting up communication between the RSA server and the RADIUS server (described in "Configuring SecurID Authentication"), and then adding SecurID users to the Steel-Belted Radius database using the instructions that follow.

Steel-Belted Radius attempts SecurID authentication only on usernames that match a SecurID entry in its User database. Steel-Belted Radius offers four types of SecurID entry, each providing a different matching rule:

- You can enter the name of a specific user.

  For example, you might create a SecurID user entry for the specific user George. This tells Steel-Belted Radius that SecurID can be used as an authentication method when an authentication request is received for username George. If username George is authenticated, the attributes of the user entry apply.

- You can enter a prefix.

  For example, you might create a SecurID user entry for the prefix sales$. This tells Steel-Belted Radius that SecurID can be used as an authentication method when an authentication request is received for a username such as sales$Harry or sales$Cynthia. Using a prefix lets you group multiple SecurID users into a single user entry instead of creating a separate entry for each SecurID user. If the sales$ user is authenticated, the attributes of the user entry apply.

  Note: Only the part of the username after the prefix (Harry or Cynthia in the example above) is sent to the RSA SecurID server.

  You can use different settings for different groups.

  Note: The user must type in the prefix as part of the username when dialing in and requesting a connection.

- You can enter a suffix.

  A suffix works like a prefix, but appears at the end of the username; for example, if the suffix were !sales, you might have usernames such as Harry!sales or Cynthia!sales.

- You can create an entry for Any user.

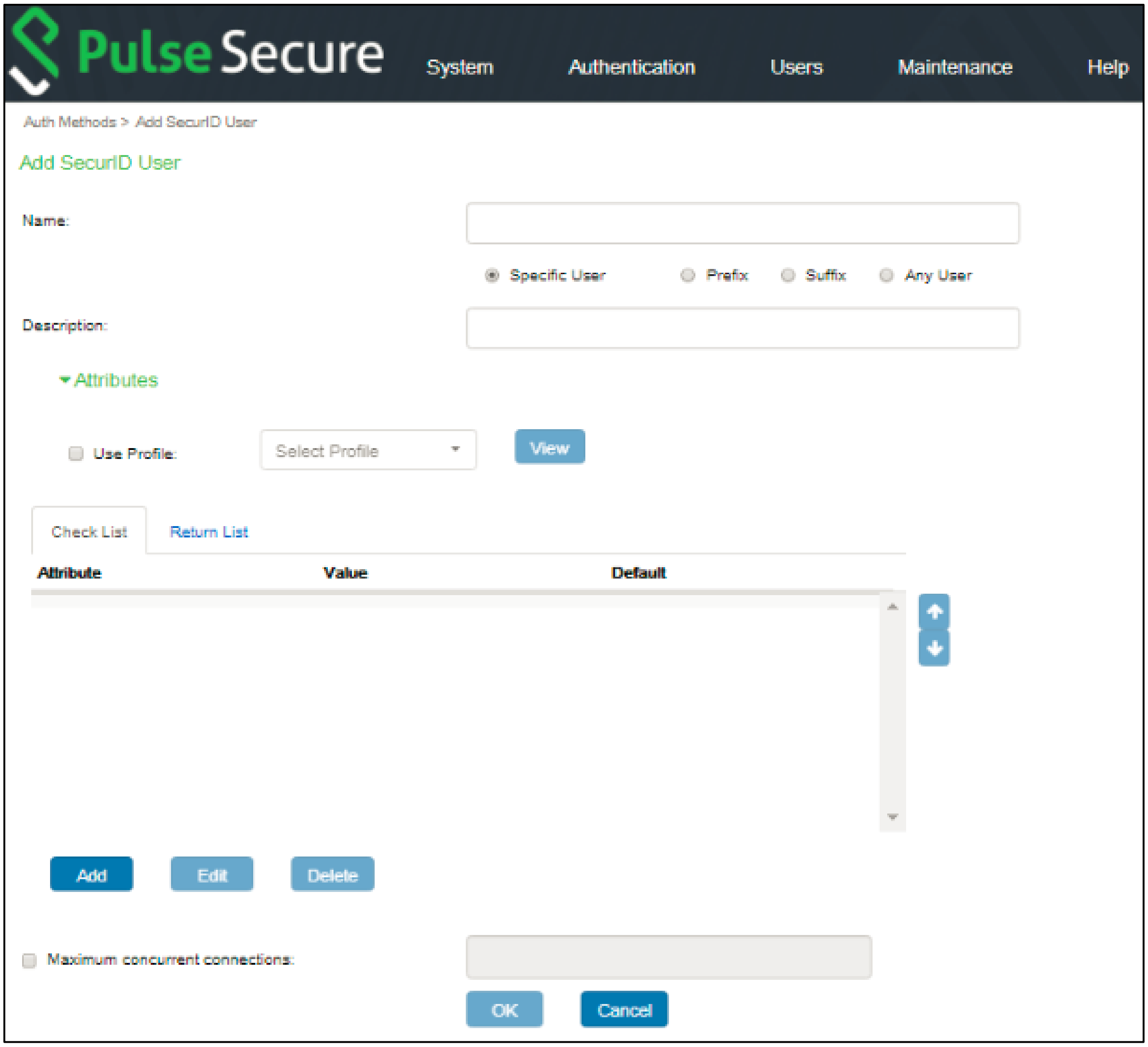
  This creates a single user entry named <ANY> that matches any username to be authenticated. SecurID can be used as an authentication method for any username, and, if successful, the attributes of the <ANY> entry apply.

  The <ANY> entry makes sense if a single set of attributes apply to all your SecurID users and if you want to make SecurID either the only authentication method used or the authentication method of last resort if other authentication methods fail.

To add a SecurID user entry:

1. Choose **Users > Auth Methods > SecurID**.
2. Click the **New** button on the SBR Administrator toolbar to display the Add SecurID User page.

Figure 59: Add SecurID User Page



Name:

● Specific User ○ Prefix ○ Suffix ○ Any User

Description:

▾ Attributes

□ Use Profile: Select Profile

Attribute	Value	Default

□ Maximum concurrent connections:

3. Enter the specific username, a prefix, or a suffix in the **Name** field.

4. Select the user type: **Specific user, Prefix, Suffix,** or **Any user.**

5. Optionally, enter a description of the user in the **Description** field.

6. If you want to use a profile to assign checklist and return list attributes to the user, click the **Use**

profile check box and use the Profile list to select the **profile** you want.

After you select a profile, you can click the **View** button to display the checklist and return list attributes in that profile.

7. If you want to specify checklist attributes or return list attributes for the user, click the **Checklist** tab or the **Return list** tab, and then click the **Add** button.

   Refer to "Adding a Checklist or Return List Attribute for a User" for information on how to add checklist and return list attributes.

8. After you have added the appropriate checklist and return list attributes for a user, use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.

9. If you want to specify the maximum number of concurrent connections this user can maintain, click the **Maximum concurrent connections** check box and enter a number in the accompanying field.

10. Click OK.

    Each new suffix or prefix entry that you add appears in the Users page with the username represented by the string USERNAME; for example, !USERNAME or SALES<USERNAME>.

Setting Up TACACS+ Users

You can configure Steel-Belted Radius to authenticate your users by querying a TACACS+ server.

ⓘ Note: Before you add TACACS+ users, you must configure communication between the Steel-Belted Radius server and the TACACS+ server by editing the tacplus.ini file. For more information on the tacplus.ini file, refer to the Steel-Belted Radius Reference Guide.

Steel-Belted Radius attempts TACACS+ authentication only on usernames that match a TACACS+ entry in its user database. Each type of TACACS+ entry specifies a different matching rule:

- You can enter the name of a specific user.

  For example, you might create a TACACS+ user entry for the specific user George. This tells Steel-Belted Radius that when an authentication request is received for username George, TACACS+ can be used as an authentication method and, if successful, the attributes of this user entry apply.

- You can enter a prefix.

  For example, you might create a TACACS+ user entry for the prefix sales$. This tells Steel-Belted Radius that when an authentication request is received for a username such as sales$Harry or sales$Cynthia, TACACS+ can be used as an authentication method and, if successful, the attributes of this user entry apply.

  ⓘ Note: Only the part of the username after the prefix (Harry or Cynthia in the example above) is sent to the TACACS+ server.

  Using a prefix lets you group multiple TACACS+ into a single user entry instead of creating a separate entry for each TACACS+ user. You can use different settings for different groups.

Note: The user must type in the prefix as part of the username he or she is using to dial in and request a connection.

- You can enter a suffix.

  A suffix works like a prefix, but appears at the end of the username; for example, if the suffix were !sales, you might have usernames such as Harry!sales or Cynthia!sales.

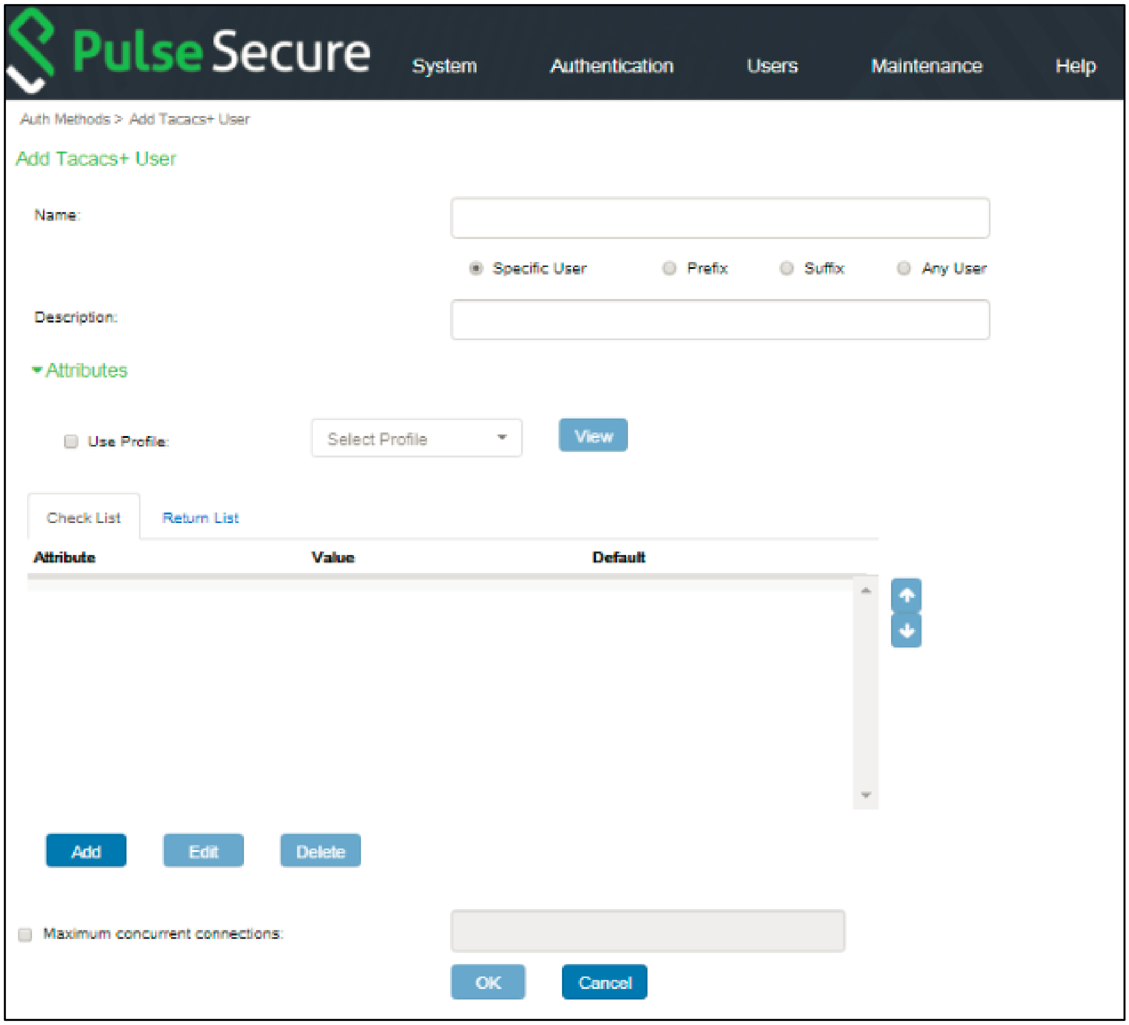
- You can create an entry for Any user.

  This creates a single user entry named <ANY> that matches any username to be authenticated. TACACS+ can be used as an authentication method for any username, and, if successful, the attributes of the <ANY> entry apply. The <ANY> entry makes sense if a single set of attributes apply to all your TACACS+ users and if you want to make TACACS+ either the only authentication method used or the authentication method of last resort if other authentication methods fail.

To add a TACACS+ user:

1. Choose Users > Auth Methods > TACACS+.
2. Click the New button on the SBR Administrator toolbar to display the Add TACACS+ User page.

Figure 60: Add TACACS+ User Page



3. Enter the specific username, a prefix, or a suffix in the Name field.
4. Select the user type: Specific user, Prefix, Suffix, or Any user.
5. Optionally, enter a description of the user in the Description field.
6. If you want to use a profile to assign checklist and return list attributes to the user, click the Use profile check box and use the Profile list to select the profile you want.

   After you select a profile, you can click the View button to display the checklist and return list attributes in that profile.
7. If you want to specify checklist attributes or return list attributes for the user, click the Checklist tab or the Return list tab, and then click the Add button.

   Refer to "Adding a Checklist or Return List Attribute for a User" for information on how to add checklist and return list attributes.
8. After you have added the appropriate checklist and return list attributes for a user, use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.

9. If you want to specify the maximum number of concurrent connections this user can maintain, click the Maximum concurrent connections check box and enter a number in the accompanying field.

10. Click OK.

Each new suffix or prefix entry that you add appears in the Users page with the username represented by the string USERNAME; for example, !USERNAME or SALES<USERNAME>.

Setting Up UNIX Users

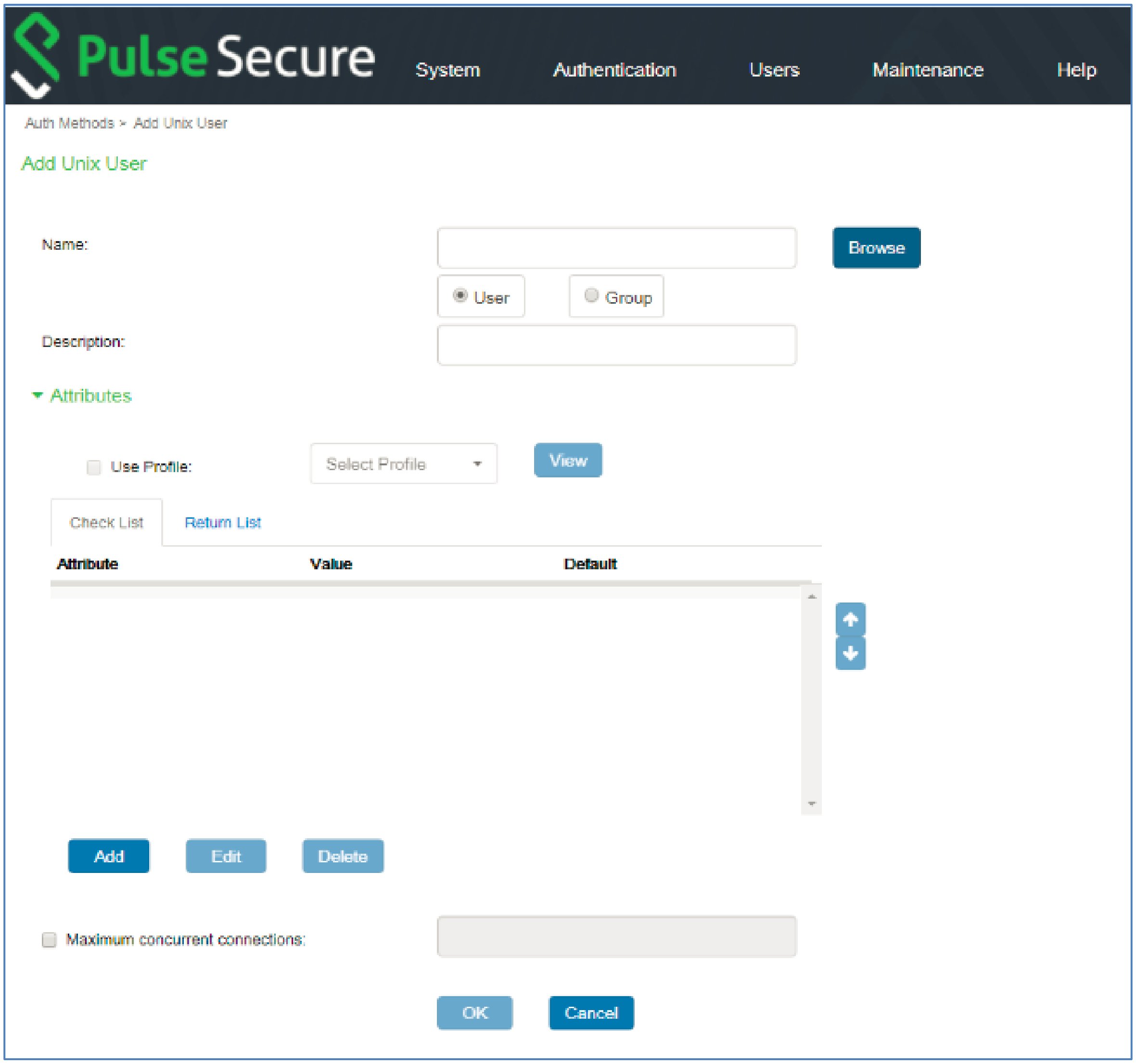
You can add a UNIX user entry to provide for the authentication of a specific user defined on a Linux server. For more flexibility, you can add a UNIX group to provide for the authentication of all users that belong to a specific group defined on the server.

To add a UNIX user or group:

1. Choose **Users > Auth Methods > UNIX.**

2. Click the **New** button on the SBR Administrator toolbar to display the Add UNIX User page.

Figure 61: Add UNIX User Page



3. Click the **Browse** button and select a user or group from the list. Click **OK**.
4. Optionally, enter a description of the user in the **Description** field.
5. If you want to use a profile to assign checklist and return list attributes to the user, click the **Use profile** check box and use the **Profile** list to select the profile you want.

   After you select a profile, you can click the **View** button to display the checklist and return list attributes in that profile.
6. If you want to specify checklist attributes or return list attributes for the user, click the **Checklist** tab or the **Return list** tab, and then click the **Add** button.

   Refer to "Adding a Checklist or Return List Attribute for a User" for information on how to add checklist and return list attributes.
7. After you have added the appropriate checklist and return list attributes for a user, use the Up and Down buttons to the right of the attribute list to put the attributes in the correct sequence.
8. If you want to specify the maximum number of concurrent connections this user can maintain, click the **Maximum concurrent connections** check box and enter a number in the accompanying field.
9. Click OK.

Editing User Settings

This section describes fields that you can set for any user entry, regardless of user type. For more information, see "User Attribute Lists" and "About Profiles".

Selecting a Profile

To select a profile for a user:

1. Open the the appropriate user page by clicking a **User** > entry.
2. Select the user whose entry you want to modify.
3. Click **Edit.**
4. Click the **Use Profile** check box.
5. Select the list to select the profile you want to use.

   To display the settings associated with the selected profile, click the **View** button.
6. When you are finished, click OK.

Setting Attribute Values

To change the value of an attribute already in the checklist or return list for a user entry:

1. Click the Checklist tab or Return List tab.
2. Select the attribute whose value you want to change.
3. Click Edit.

4. When the Change page opens, enter or select the new value.

   Depending on the attribute, you can enter a new value or select a value from a list. For some attributes, Steel-Belted Radius retrieves the value from the server and you cannot enter a value in this page.

5. Click **OK.**

Removing Attribute/Value Pairs

To remove an attribute/value pair already in the checklist or return list for a User entry:

1. Click the **Checklist** tab or **Return List** tab.
2. Select the attribute/value pair you'd like to remove.
3. Click Delete.

Reordering Attributes

Certain attributes are multi-valued and orderable; that is, the attribute/value pair can appear more than once in a RADIUS response, and the order in which the attribute/value pairs appear is significant. To reorder attributes in a User entry:

1. Click the **Checklist** tab or **Return List** tab.
2. Highlight an attribute/value pair in the list.
3. Click the Up or Down arrow to move the selected attribute within the list.
   - The Up arrow moves the selected attribute/value up in the list. If the attribute is not orderable, or if the selected item already the first value for this attribute, the button is disabled.
   - The Down arrow moves the selected attribute/value down in the list. If the attribute is not orderable, or if the selected item is already the last value for this attribute, the button is disabled.

Changing Attributes Inherited from a Profile

Checklist and return list attributes can be specified for a user, or they can be inherited from a profile associated with a user. Attributes inherited from a profile are overridden by attributes assigned to a specific user.

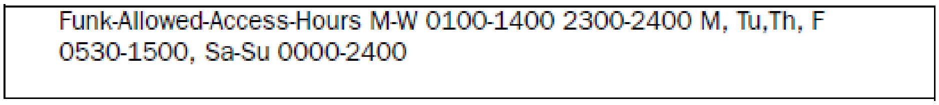
Concurrent Connection Limits

A maximum number of open connections can be set for each user entry by checking the **Maximum concurrent connections** check box and entering a number in the accompanying field. When the user requests access, the user can be authenticated using the given authentication method only if fewer than this number of connections are currently open for this user.

Allowed Access Hours

The user's allowed access hours can be specified by adding the Funk-Allowed-Access-Hours attribute to the user's checklist.

Funk-Allowed-Access-Hours is a variable-length string that identifies time periods in a 7-day week of 24-hour days. This string consists of one or more day specifiers (each of which can list one or more days and/or ranges of days) followed by one or more ranges of 24-hour times, in minutes.



```
Funk-Allowed-Access-Hours M-W 0100-1400 2300-2400 M, Tu,Th, F
0530-1500, Sa-Su 0000-2400
```

The syntax rules for Funk-Allowed-Access-Hours are as follows:

- Time ranges can be inclusive (1000–1100 allows access only between 10 a.m. and 11 a.m.) or exclusive (1100–1000 allows access any time except between 10 a.m. and 11 a.m.).
- Day specifiers, and ranges of days and times can be separated by commas or spaces; ranges of days or times are indicated by hyphens (m–w or 0239–1459).
- Days can be specified by the minimum number of case-insensitive letters necessary to distinguish them (Su, M, Tu, W, Th, F, Sa) and can wrap around the end of the week (Sa-Su).
- At least one time period is required for each day; that is, each day, list, or range of days must be followed by one or more ranges of times.
- Times are specified using four digits, with leading zeroes where needed (0001 for 12:01 a.m., 0630 for 6:30 a.m., and so forth).

When assigned to a user's checklist, the Funk-Allowed-Access-Hours value in Figure 62: Sample Funk-Allowed-Access-Hours Attribute Value allows the user access during the following time periods:

- 1 a.m. to 2 p.m. and 11:00 p.m. to midnight, Monday through Wednesday
- 5:30 a.m. to 3:00 p.m. Monday, Tuesday, Thursday, and Friday
- Any time Saturday or Sunday

The total access times Monday are 1 a.m. to 3:00 p.m. and 11:00 p.m. to midnight

If the user attempts access on Sunday at 11:30 p.m., access would be allowed and a Session-Timeout attribute specifying a value of 1800 seconds (30 minutes, until midnight Sunday, when the access period ends) would be returned. However, if the user's return list includes a Session-Timeout with a value less than 1800, this lesser value would be returned.

Deleting a User

To delete a user:

1. Open the appropriate user page by clicking a User > entry.
2. Select the user you want to delete.
3. Click the Delete button in the appropriate User page.
4. When you are prompted to confirm the deletion, click Yes

Chapter 11
Administering Profiles via Legacy SBR Administrator

This chapter describes how to set up and administer user profiles via legacy SBR administrator.

About Profiles

Steel-Belted Radius lets you define default templates of checklist and return list pairs called profiles. A profile provides specific attributes for one or both lists. You can define as many profiles as you require. Profiles provide a powerful means of managing and configuring accounts.

When you edit a User entry, you can assign a profile to the User; the checklist and return list attributes of that profile then become the default settings for the User entry. After you assign a profile to a User entry, you can modify the new entries on the user's checklist and return list. Changes you make apply only to the specific user entry; they do not affect the profile itself. Assigning a profile and then overriding individual attributes is a convenient way to leverage Steel-Belted Radius's features to your advantage.

To change attributes settings across many users immediately, edit the profile that you have assigned to these users. The changes you make to a profile are automatically reflected in each user's checklist and return list.

Adding a Checklist or Return List Attribute for a Profile

A checklist attribute is an item of information that must accompany a request for connection before the connection can be authenticated. A return list attribute is an item of information that Steel-Belted Radius includes in the Access-Accept message when a connection request is approved.

Resolving Profile and User Attributes

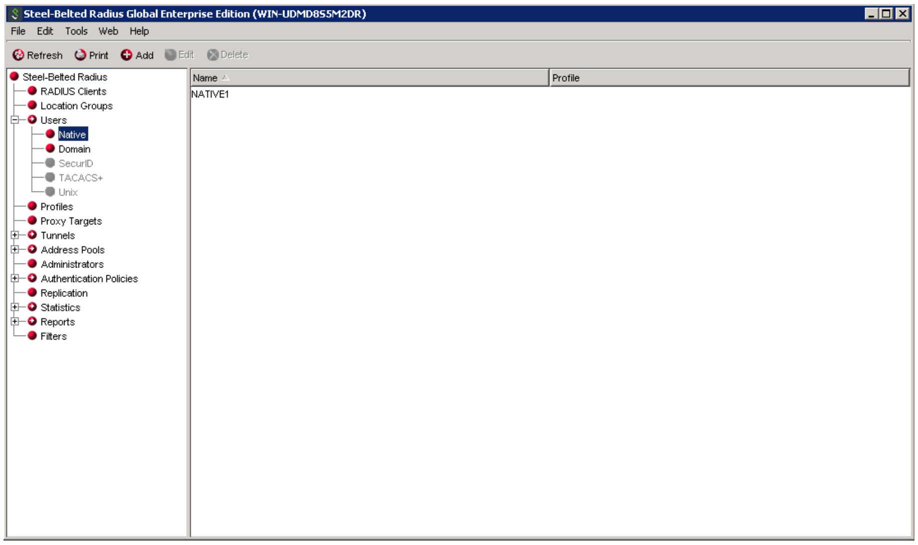
If user-specific attributes are stored in an external database, Steel-Belted Radius determines the final set of attributes for a user by merging the attributes stored in the native database with those retrieved from the external database. This calculation is performed as follows:

1. The attributes from the profile (or Alias user) assigned to the user are first retrieved.
2. These attributes are then merged with the user-specific modifications to the attributes in the following manner:
   - If the attribute is multi-valued, then the attribute(s) retrieved from the external database is added to the overall list of attributes.
   - If the attribute is single-valued, then the attribute(s) retrieved from the external database replaces any attribute of the same name in the profile or associated with the alias.
   - If the attribute is orderable, then the attribute(s) retrieved from the external database replaces any orderable attribute of the same name in the profile or associated with the alias.

Setting Up Profiles

The Profiles panel (Figure 63: Profiles Panel) lets you define sets of checklist and return list attributes. You can then assign these profiles to users to simplify user administration.

Figure 63: Profiles Panel



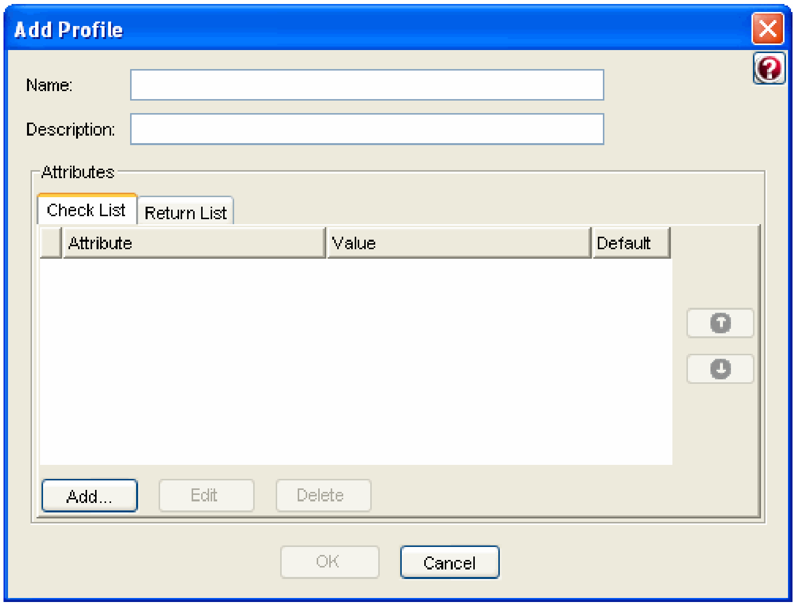
Name	Profile
NATIVE1	

Adding a Profile

To add a profile:

1. Click Profiles to open the Profiles panel.
2. Click the Add button on the SBR Administrator toolbar. The Add Profile dialog appears.

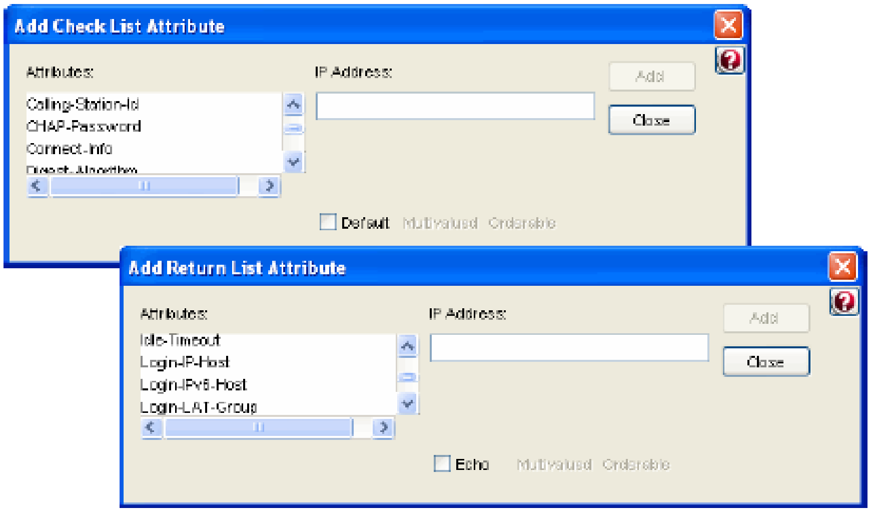
Figure 64: Add Profile Dialog



3. Enter a name for the new profile in the Name field.
4. Optionally, enter a description for the profile in the Description field.
5. Add checklist and return list attributes to the profile.
   a. Click the Checklist tab or the Return List tab.

b. Click Add. The Add Checklist Attribute dialog or the Add Return List Attribute dialog opens.

Figure 65: Add Checklist Attribute and Add Return List Attribute Dialogs



c. Select the attribute you want to add from the Attributes list.

d. Select or enter a value for the attribute.

The dialog changes according to the attribute you choose. Some attributes require that you enter a value, string, or IP address. Other attributes require that you choose from a fixed list of values.

If the **Multivalued** indicator is dimmed, an attribute can have only one value. If the **Multivalued** attribute is undimmed, you can add multiple values for the attribute.

(Checklist attributes only) To set this value to the default value for the attribute (which is useful in situations where the attribute is not included in the RADIUS request), check the **Default value** check box.

(Return list single-valued attributes only) If you do not want to specify a particular value, but want to make sure that whatever value of the attribute appears in the RADIUS request is echoed to the client in the RADIUS response, click the **Echo** check box.

e. Click **Add** to add this attribute/value pair to the list.

f. When you are finished adding attribute/value pairs, click **Close** to return to the Add Profile dialog.

6. Click **OK** to save the profile.

Removing a Profile

To remove a profile:

1. Open the Profiles panel.
2. Select the entry for the profile you want to remove.
3. Click the Delete button on the SBR Administrator toolbar (or right-click the profile entry and choose Delete from the context menu).

4. When you are prompted to confirm the deletion, click Yes.

   Note: Do not delete a profile that is assigned to a user. If you delete an active profile, the attributes defined in the profile are removed from user’s settings, possibly resulting in authentication failures.

Chapter 12
Administering Profiles via WebGUI

This chapter describes how to set up and administer user profiles via WebGUI.

About Profiles

Steel-Belted Radius lets you define default templates of checklist and return list pairs called profiles. A profile provides specific attributes for one or both lists. You can define as many profiles as you require. Profiles provide a powerful means of managing and configuring accounts.

When you edit a User entry, you can assign a profile to the User; the checklist and return list attributes of that profile then become the default settings for the User entry. After you assign a profile to a User entry, you can modify the new entries on the user's checklist and return list. Changes you make apply only to the specific user entry; they do not affect the profile itself. Assigning a profile and then overriding individual attributes is a convenient way to leverage Steel-Belted Radius's features to your advantage.

To change attributes settings across many users immediately, edit the profile that you have assigned to these users. The changes you make to a profile are automatically reflected in each user's checklist and return list.

Adding a Checklist or Return List Attribute for a Profile

A checklist attribute is an item of information that must accompany a request for connection before the connection can be authenticated. A return list attribute is an item of information that Steel-Belted Radius includes in the Access-Accept message when a connection request is approved.

Resolving Profile and User Attributes

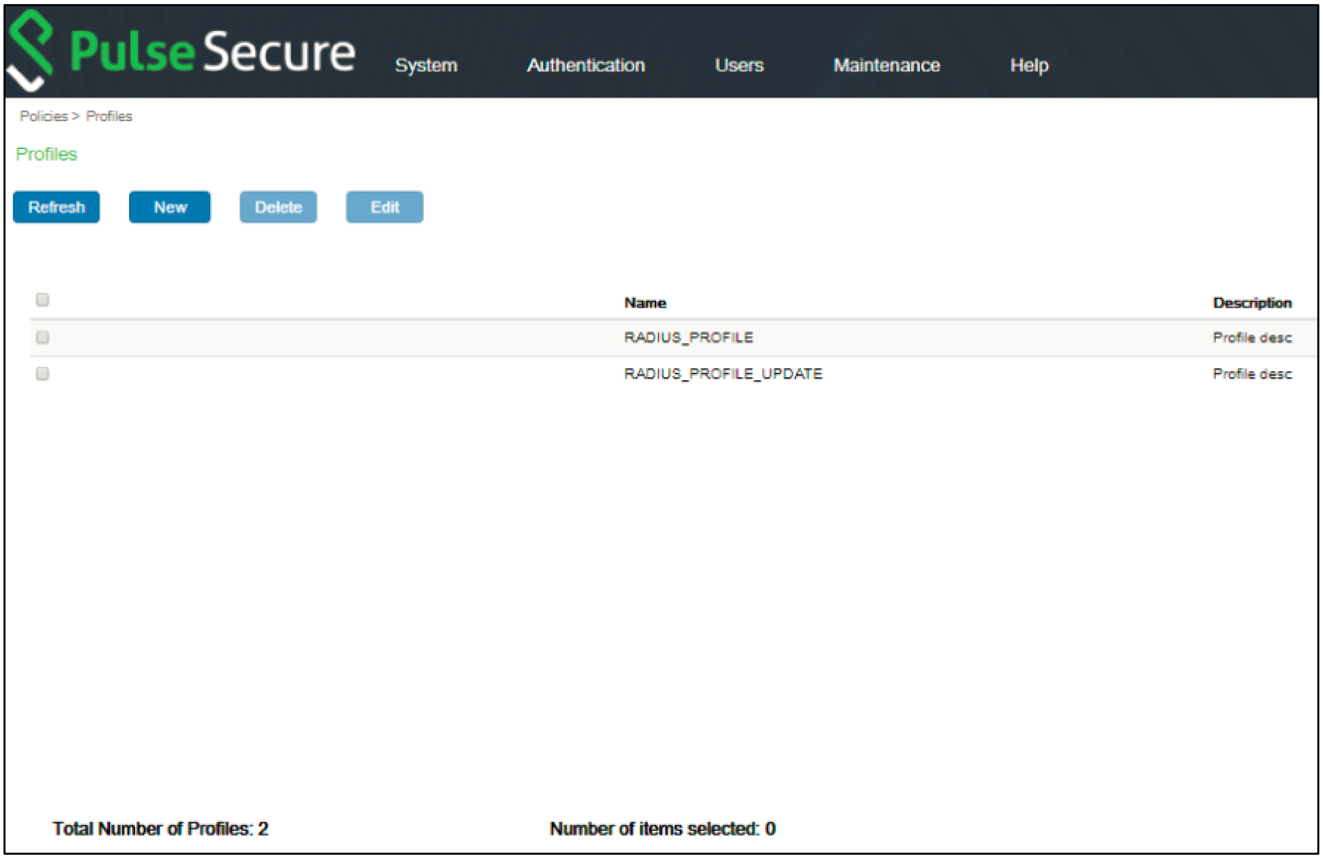
If user-specific attributes are stored in an external database, Steel-Belted Radius determines the final set of attributes for a user by merging the attributes stored in the native database with those retrieved from the external database. This calculation is performed as follows:

1. The attributes from the profile (or Alias user) assigned to the user are first retrieved.
2. These attributes are then merged with the user-specific modifications to the attributes in the following manner:
   - If the attribute is multi-valued, then the attribute(s) retrieved from the external database is added to the overall list of attributes.
   - If the attribute is single-valued, then the attribute(s) retrieved from the external database replaces any attribute of the same name in the profile or associated with the alias.
   - If the attribute is orderable, then the attribute(s) retrieved from the external database replaces any orderable attribute of the same name in the profile or associated with the alias.

Setting Up Profiles

The Profiles page (Figure 66: Profiles Page) lets you define sets of checklist and return list attributes. You can then assign these profiles to users to simplify user administration.

Figure 66: Profiles Page



□	Name	Description
□	RADIUS_PROFILE	Profile desc
□	RADIUS_PROFILE_UPDATE	Profile desc

Total Number of Profiles: 2

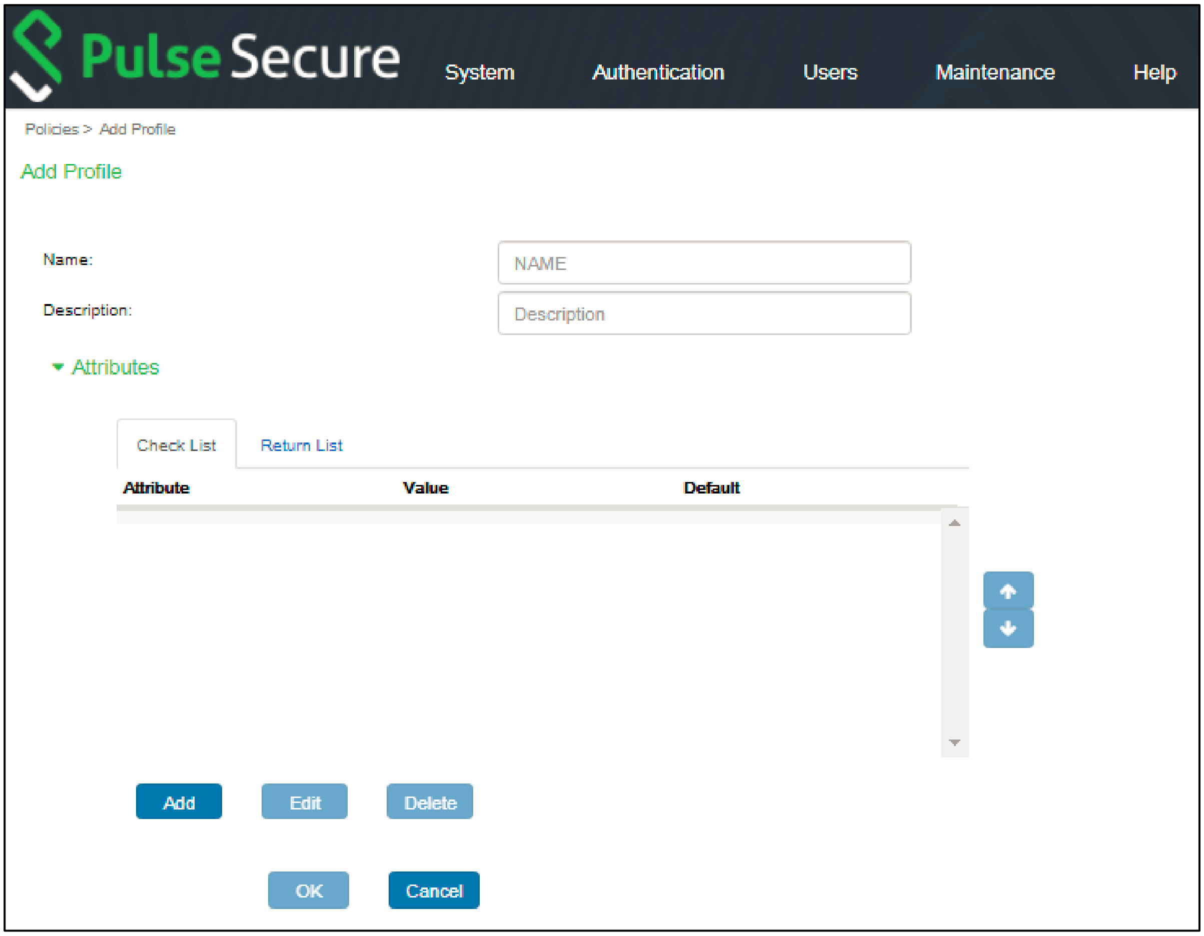
Number of items selected: 0

Adding a Profile

To add a profile:

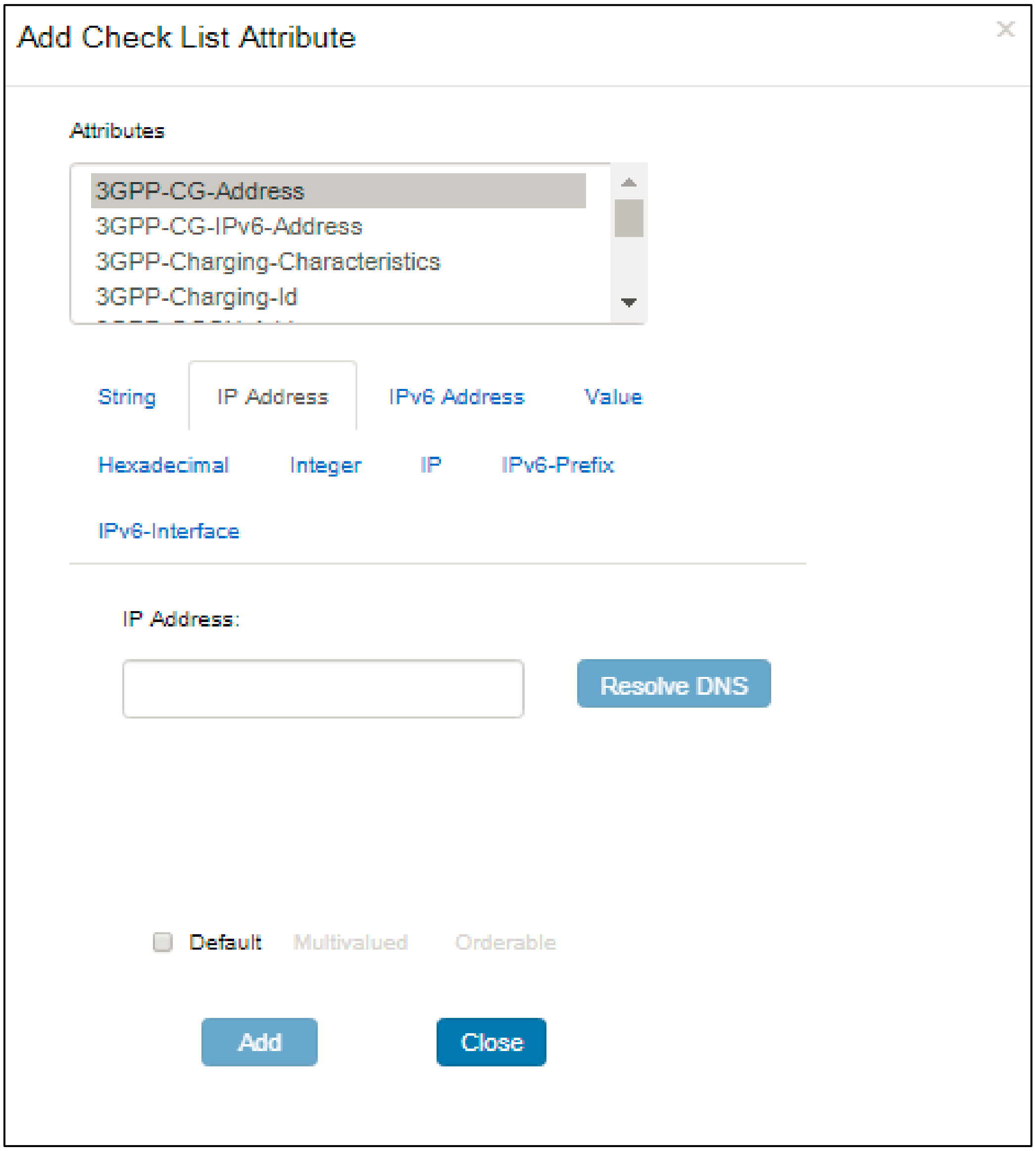
1. Choose **System > Policies > Profiles** to open the Profiles page.
2. Click the New button on the SBR Administrator toolbar. The Add Profile page appears.

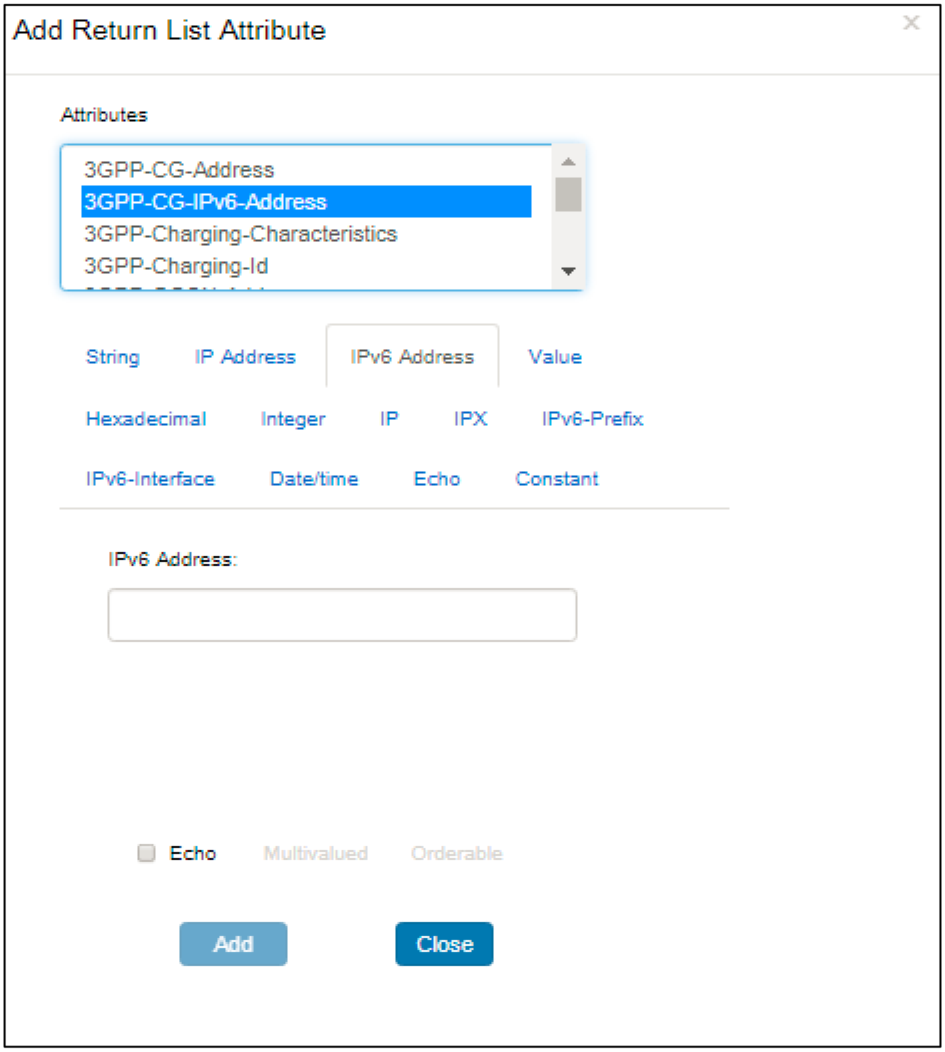
Figure 67: Add Profile Page



3. Enter a name for the new profile in the Name field.
4. Optionally, enter a description for the profile in the Description field.
5. Add checklist and return list attributes to the profile.
   a. Click the Checklist tab or the Return List tab.
   b. Click Add. The Add Checklist Attribute page or the Add Return List Attribute page opens.

Figure 68: Add Checklist Attribute and Add Return List Attribute Pages





c. Select the attribute you want to add from the Attributes list.

d. Select or enter a value for the attribute.

   The page changes according to the attribute you choose. Some attributes require that you enter a value, string, or IP address. Other attributes require that you choose from a fixed list of values.

   If the **Multivalued** indicator is dimmed, an attribute can have only one value. If the **Multivalued** attribute is undimmed, you can add multiple values for the attribute.

   (Checklist attributes only) To set this value to the default value for the attribute (which is useful in situations where the attribute is not included in the RADIUS request), check the **Default value** check box.

   (Return list single-valued attributes only) If you do not want to specify a particular value, but want to make sure that whatever value of the attribute appears in the RADIUS request is echoed to the client in the RADIUS response, click the **Echo** check box.

e. Click **Add** to add this attribute/value pair to the list.

f. When you are finished adding attribute/value pairs, click **Close** to return to the Add Profile page.

6. Click **OK** to save the profile.

Removing a Profile

To remove a profile:

1. Choose System > Policies > Profiles to open the Profiles page.
2. Select the entry for the profile you want to remove.
3. Click the Delete button.
4. When you are prompted to confirm the deletion, click Yes.

Note: Do not delete a profile that is assigned to a user. If you delete an active profile, the attributes defined in the profile are removed from user’s settings, possibly resulting in authentication failures.

Chapter 13
Administering Proxy RADIUS via Legacy SBR Administrator

This chapter presents an overview of proxy RADIUS and describes how to set up proxy targets via legacy SBR administrator.

About Proxy RADIUS

Steel-Belted Radius can forward a RADIUS request to another server for processing and relay the other server's result back to its client. Steel-Belted Radius is acting as a proxy for the target server, and that Steel-Belted Radius is proxy-forwarding the request to the target server.

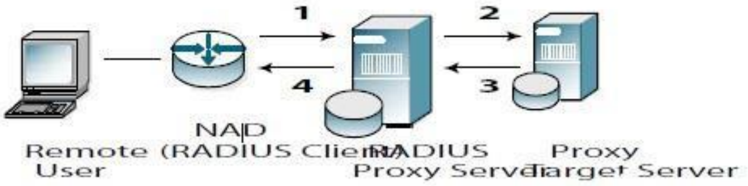
Any Steel-Belted Radius server can act as proxy or target for authentication or accounting messages (or both).

Proxy RADIUS Authentication

Figure 69: RADIUS Proxy Forwarding illustrates how RADIUS authentication messages are proxy-forwarded:

1. A network access device (RADIUS client) sends an authentication request to a RADIUS proxy server.
2. The proxy RADIUS server forwards the message to a RADIUS target server.
3. The target RADIUS server performs the authentication services indicated by the message, then returns a response message to the proxy RADIUS server.
4. The proxy RADIUS server relays the response message to the RADIUS client.

Figure 69: RADIUS Proxy Forwarding



Proxy RADIUS Accounting

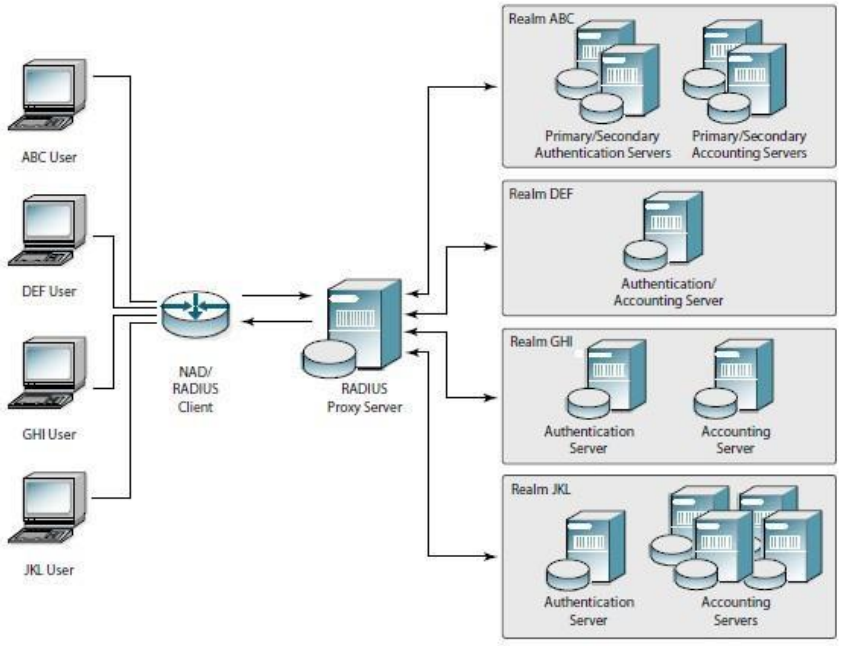
RADIUS accounting messages are proxy-forwarded as follows:

1. A RADIUS server receives an accounting request.
2. Depending on its configuration, the RADIUS server forwards the accounting message to a target server, records accounting attributes locally on the proxy server, or records the information in both places.
3. If the proxy server does not receive an acknowledgement of the forwarded packet, it periodically re-sends the packet according to its retry policy.

Proxy RADIUS Realms

Proxy RADIUS realms are pools of RADIUS servers to which Steel-Belted Radius can forward RADIUS requests. Proxy RADIUS realms can be configured to support workload distribution, redundancy, fault tolerance, retry policies, primary-secondary server roles, and separation of authentication and accounting responsibilities by server. For more information, see “Configuring a Proxy RADIUS Realm”.

Figure 70: RADIUS Server and Proxy Realms



Target Selection within a Realm

For proxy RADIUS realms, after the destination realm is identified, Steel-Belted Radius must select a target within the realm. Target selection depends upon a number of factors, all of which you can set up in advance by editing the realm configuration files on the Steel-Belted Radius server: proxy.ini, radius.ini, filter.ini, and one RealmName.pro file per realm.

After the target is selected, Steel-Belted Radius matches the target name with a proxy entry in its database. Using the data in this entry (IP address, UDP port, shared secret) Steel-Belted Radius establishes a connection between itself and the target, and proxy-forwards the RADIUS request. Note that you can configure the realm so that all realm routing information and delimiters are stripped from the User-Name before forwarding.

The target processes the request as it normally would for RADIUS authentication or accounting. In the case of authentication, Steel-Belted Radius waits for a response from the target, then relays this response to its RADIUS client.

Message-Authenticator Support

The Message-Authenticator attribute enables Steel-Belted Radius to determine whether the packet received is from an actual proxy server. It might also sign the forward request.

Steel-Belted Radius can be configured to use the Message-Authenticator attribute when forwarding packets using proxy RADIUS. It can also be configured to validate or ignore the Message-Authenticator if included in the packets received.

Proxy Fast-Fail

During proxy forwarding, Steel-Belted Radius acts as the RADIUS client of another RADIUS server. Since RADIUS clients take responsibility for delivering RADIUS packets, all of them have a retry policy that determines how often and for how long they continue to try to deliver a packet until they receive the response that they expect from the RADIUS server.

This includes the Steel-Belted Radius server when it acts as the RADIUS client of a proxy RADIUS target server. Steel-Belted Radius provides a fast-fail option for proxy RADIUS realms. This fast-fail feature saves Steel-Belted Radius from continuing to send packets to a target server that appears to be down temporarily. For example, if Steel-Belted Radius is sending a packet to a target and it is not getting the timely response it expects, it periodically tries to send the packet until it reaches the number of tries in its retry policy. If it still hasn't received a response from the target at that point, Steel-Belted Radius removes the target from the active list and places it on the fast-fail list.

Each time a request from a realm is received, Steel-Belted Radius sends a probing request to all fast-fail entries for this realm. No response is expected or required from the probes. No retry policy is followed. If a response to a probe is received, that target is removed from the fast-fail list. When the fast-fail timer expires for a target, it is placed back on the active list.

We strongly recommend that you specify a [FastFail] section in each proxy RADIUS realm configuration (.pro) file. The [FastFail] section permits you to fine-tune retry policies for individual realms, or for specific targets within realms. Any [FastFail] settings that you supply in a .pro file override the current ProxyFastFail setting.

The radius.ini file offers a ProxyFastFail setting for single-target proxy entries that are not a member of any realm. ProxyFastFail has an integer value, usually 1800. If a target remains on the fast-fail list longer than this number of seconds, it is automatically removed from the fast-fail list. If conditions warrant, a target might be returned to the fast-fail list at any time.

For information on configuring the radius.ini file to support the fast-fail feature, see the Steel-Belted Radius Reference Guide.

Static Proxy Accounting

Static proxy accounting allows you to send copies of certain types of accounting messages to proxy RADIUS realms, as well as to the normal routing of the original accounting message. The number of copies is not limited.

Static proxy accounting does not prevent the request from being dynamically routed for RADIUS accounting services based on User-Name decoration, DNIS number, or attribute mapping, nor does it prevent local logging or other accounting methods from occurring. If static proxy-forwarding fails (due to a lack of response from the target), this does not prevent the original RADIUS accounting request from being acknowledged.

An important function of static proxy accounting is to ensure that Accounting-On and Accounting-Off messages can be routed to realms. A NAD (RADIUS client) normally issues these accounting messages to its RADIUS server when it goes online (Accounting-On) and offline (Accounting-Off). In such cases, all connections previously made by this NAD are considered invalid, and the RADIUS server can free resources that it allocated to those sessions.

Static proxy accounting is necessary to deliver Accounting-On and Accounting-Off messages to realms, because these messages do not contain the User-Name or Called-Station-Id attributes that Steel-Belted Radius would normally use to route packets to realms.

For example, assume the original Access-Request, an authentication message, was used to determine the realm destination for both authentication and accounting for a particular session. The attribute used to route the Access-Request might have been the User-Name, the Called-Station-Id, or any other RADIUS attribute in the Access-Request, depending on how you have configured request routing for authentication messages.

Accounting packets for this same session can be matched with the realm destination only if the server knows which session is involved (as it does in Start, Stop, and Interim messages). The Accounting-On and Accounting-Off messages are independent of specific sessions, therefore, it is impossible to route them to realms without additional information.

By setting up static proxy accounting, and listing all realms as targets for Accounting-On and Accounting-Off messages, you can ensure that network information (such as NAD status) is sent to everyone who might require it.

Proxy AutoStop Feature

A user session can be removed from the Current Sessions table in ways other than the usual Accounting-Stop message from the NAD:

- An Accounting-On or Accounting-Off message received from the NAD causes all sessions originating from the NAD to be purged, as these messages signal that the NAD has been restarted or is going down.
- The administrator can remove users by means of the LDAP configuration interface (LCI).
- The administrator can remove users by means of the SBR Administrator application.

Termination information must be passed on if the users exist as proxied sessions on downstream RADIUS servers because these servers must free the resources previously allocated to the session(s), which have now been terminated. The Proxy AutoStop feature handles such cases. In addition, if you use the LCI command to free resources from the central administrative server, the appropriate messages are propagated so that the resources associated with the user in each of the downstream servers are automatically freed.

Adding a Proxy Target

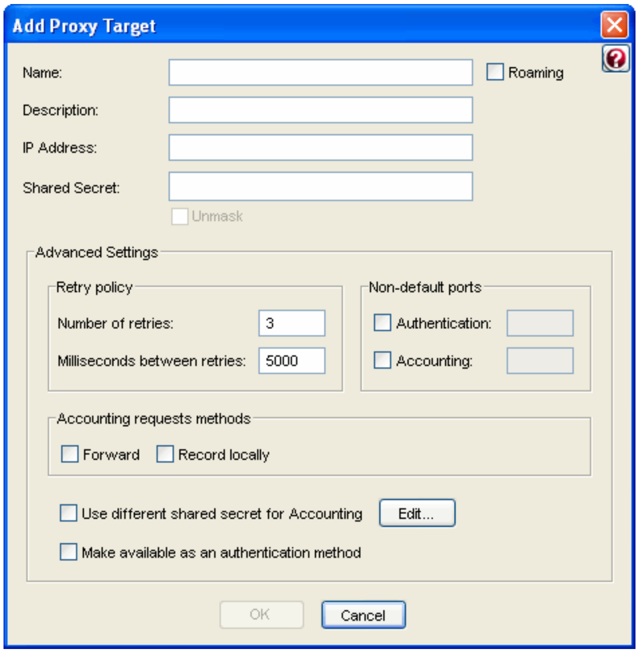
This section explains how to set up proxy forwarding from the Steel-Belted Radius server (the proxy) to another RADIUS server (the target).

To add a proxy target:

1. Choose **Proxy Targets** to open the Proxy Target panel.
2. Click the **Add** button on the SBR Administrator toolbar. The Add Proxy Target dialog

   (Figure 71: Add Proxy Target Dialog appears.

Figure 71: Add Proxy Target Dialog



Add Proxy Target

Name: □ Roaming

Description:

IP Address:

Shared Secret:

□ Unmask

Advanced Settings

Retry policy

Number of retries: 3

Milliseconds between retries: 5000

Non-default ports

□ Authentication:

□ Accounting:

Accounting requests methods

□ Forward □ Record locally

□ Use different shared secret for Accounting

□ Make available as an authentication method

3. Enter the name of the proxy target in the **Name** field.

   The target name must not duplicate any other target name, realm name, or tunnel name in your Steel-Belted Radius configuration. The name you record for a proxy target is not used in processing; Steel-Belted Radius uses the proxy target's IP address to route RADIUS packets.

4. Enter a description for the proxy target in the **Description** field.

5. Enter the IP address or DNS name of the proxy target in the **IP Address** field.

   If you enter the DNS name of the proxy target, the SBR Administrator resolves the name you enter to an IP address automatically.

6. Enter the shared secret for the proxy target in the **Shared Secret** field.

   Shared secrets are case-sensitive.

   If you want the characters in the shared secret (rather than asterisks) to appear as you type, click the **Unmask** check box.

   The shared secret configured for the proxy target in Steel-Belted Radius must match the shared secret configured on the proxy target.

7. Specify how many times Steel-Belted Radius should try to reach the proxy target and how long to wait between attempts in the **Number of retries and Milliseconds** between retries fields.

   When Steel-Belted Radius acts as a proxy, it emulates the characteristics of a network access device. This includes the ability to retransmit a request if the first attempt does not get a timely response from the proxy target.

   - The **Number of retries** field specifies the number of times a request is retransmitted if an acknowledgment from the target is not received; if the number of retries is exhausted, then the original request is rejected. By default, Steel-Belted Radius retries three times before giving up.

- The **Milliseconds between retries** field specifies the time interval between each retry in milliseconds (thousandths of a second). By default, Steel-Belted Radius waits 5000 milliseconds (5 seconds) between retries.

8. If the proxy target uses ports different from what the proxy intermediary uses for authentication or accounting, click the **Authentication or Accounting** check box and enter the port number you want Steel-Belted Radius to use when exchanging RADIUS authentication or accounting information with the proxy target.

   The port numbers configured for the proxy target in Steel-Belted Radius must match the port numbers configured on the proxy target. By default, Steel-Belted Radius uses port 1645 for authentication and port 1646 for accounting.

9. Specify whether you want accounting requests to be forwarded or recorded locally.

   - If you click the **Forward** check box, Steel-Belted Radius forwards the accounting transaction to the same proxy target that received the authentication transaction.

   - If you click the **Record locally** check box, Steel-Belted Radius logs the accounting transaction locally (regardless of whether an authentication request was forwarded to the proxy target.

   You can click both check boxes if you want accounting requests to be forwarded and logged locally.

10. If you want Steel-Belted Radius to use a different shared secret for accounting when communicating with the proxy target, click the **Use different shared secret for accounting** check box and click the **Edit** button to specify an accounting shared secret.

    Refer to "Maintaining an Accounting Shared Secret" for information on using the Edit Accounting Shared Secret dialog.

11. If you want to use a proxy target as an authentication method, click the **Make available as an authentication method check box**.

    If you enable this option, the name of the proxy target appears in the Authentication Methods tab of the Authentication Policies panel as proxy:name. This is useful if you have user records defined on an older RADIUS server and you want to provide a seamless migration to Steel-Belted Radius. Using the older server as a proxy RADIUS target means that RADIUS requests that arrive addressed to this target are handled by Steel-Belted Radius automatically, without requiring end users to change their addressing conventions.

12. Click **OK**.

Note: If the proxy target that you are configuring is a member of a proxy RADIUS realm, you should ensure that the Make available as an authentication method check box is unchecked.

Ask the administrator at the target site to log into the target server's RADIUS configuration program and add Steel-Belted Radius as a RADIUS client of the target server. You will need to provide this administrator with the IP address of the Steel-Belted Radius server.

Maintaining an Accounting Shared Secret

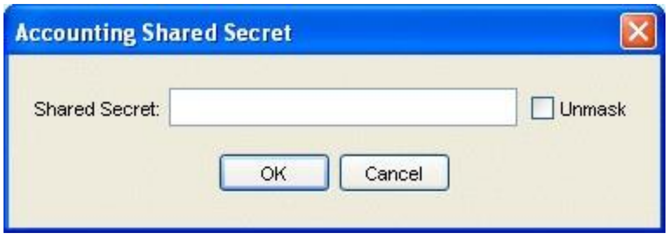
To specify a shared secret for accounting:

1. Click the Use **different shared secret for accounting** check box.

2. Click the **Edit** button.

3. When the Accounting Shared Secret dialog opens, enter the shared secret you want Steel-Belted Radius to use.c

Figure 72: Accounting Shared Secret Dialog



If you want the characters in the shared secret (rather than asterisks) to appear as you type, click the **Unmask** check box. Note that shared secrets are case-sensitive.

4. Click **OK.**

Deleting a Proxy Target

To remove a target server from the proxy target list:

1. In the Proxy Target panel, select the target server you'd like to remove.

2. Click **Delete.**

Steel-Belted Radius as a Target

This section describes how to set up proxy forwarding from some other RADIUS server (the proxy) to the Steel-Belted Radius server (the target):

1. Set up the proxy as a RADIUS client of Steel-Belted Radius.

   Add the entry using the RADIUS Clients panel. Specify the proxy's name, its IP address, and the shared secret that you want to use for encryption between the proxy and Steel-Belted Radius.

2. Ask the administrator at the target site to log into the proxy's RADIUS configuration program and set up Steel-Belted Radius as a proxy RADIUS target. You will need to provide this administrator with the IP address of the Steel-Belted Radius server.

   Note: Make sure that the same UDP port and shared secret are entered on both proxy and target sides.

Dictionaries when Steel-Belted Radius is the Target

When Steel-Belted Radius receives a proxy-forwarded packet, it consults its RADIUS client entry for that proxy server. The **Make/model** field of this entry determines which attribute dictionary Steel-Belted Radius uses.

At various different times, Steel-Belted Radius can receive requests from the same proxy server that have originated from different network access devices, possibly of different types. The single **Make/model** field that was entered for the proxy might not be adequate to handle the variety of RASs on the “other side” of the transaction.

One way to handle this problem is to add the originating network access devices to Steel-Belted Radius’s list of RADIUS clients. Steel-Belted Radius can be configured to examine each proxy-forwarded packet for clues as to the make and model of the originating device. If clues are found, Steel-Belted Radius does everything it can to map this information to a vendor-specific dictionary, and uses this dictionary in preference to the one for the proxy.

Accepting Packets from Any Proxy

If you’d like Steel-Belted Radius to be able to accept proxy requests from any IP address, you can use the RADIUS Clients panel to add a special entry called **<ANY>**, and specify a shared secret. The **<ANY>** entry permits forwarded requests from any proxy to be accepted, provided the shared secret is correct.

ⓘ Note: This feature requires that proxies are configured to use the shared secret you provide in the <ANY> entry.

Proxy RADIUS as an Authentication Method

Any target proxy RADIUS server can be configured as a Steel-Belted Radius authentication method by enabling the **Make available as an authentication method** check box in the Add Proxy Target/Edit Proxy Target dialog.

A target server can be set up as an authentication method even if the end users do not know anything about the target. That is, a user does not need to log in using a decorated username such as User@TargetName to be authenticated by the target server.

If you prioritize the proxy: TargetName authentication method above the Native User authentication method in the authentication methods list, the user can log in as User and Steel-Belted Radius automatically sends the request to the target for authentication. The authentication succeeds if the UserName and password are stored on the target, but if not, Steel-Belted Radius reaches the Native User method eventually, and the user can then be authenticated.

This technique is useful as a migration path to Steel-Belted Radius from other RADIUS servers. You can set up Steel-Belted Radius as the proxy and the old RADIUS server as the target. After proxy authentication is enabled (in the Proxy Targets panel) and prioritized (in the Authentication Policies panel), Steel-Belted Radius can authenticate users against the old RADIUS server, either as an automatic “first choice” or as an alternative when authentication against the new server’s “local” database fails.

Chapter 14
Administering Proxy RADIUS via WebGUI

This chapter presents an overview of proxy RADIUS and describes how to set up proxy targets via WebGUI.

About Proxy RADIUS

Steel-Belted Radius can forward a RADIUS request to another server for processing and relay the other server's result back to its client. Steel-Belted Radius is acting as a proxy for the target server, and that Steel-Belted Radius is proxy-forwarding the request to the target server.

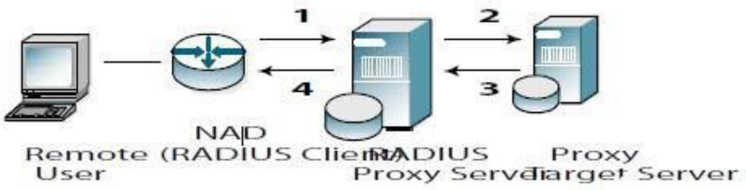
Any Steel-Belted Radius server can act as proxy or target for authentication or accounting messages (or both).

Proxy RADIUS Authentication

Figure 73: RADIUS Proxy Forwarding illustrates how RADIUS authentication messages are proxy-forwarded:

1. A network access device (RADIUS client) sends an authentication request to a RADIUS proxy server.
2. The proxy RADIUS server forwards the message to a RADIUS target server.
3. The target RADIUS server performs the authentication services indicated by the message, then returns a response message to the proxy RADIUS server.
4. The proxy RADIUS server relays the response message to the RADIUS client.

Figure 73: RADIUS Proxy Forwarding



Proxy RADIUS Accounting

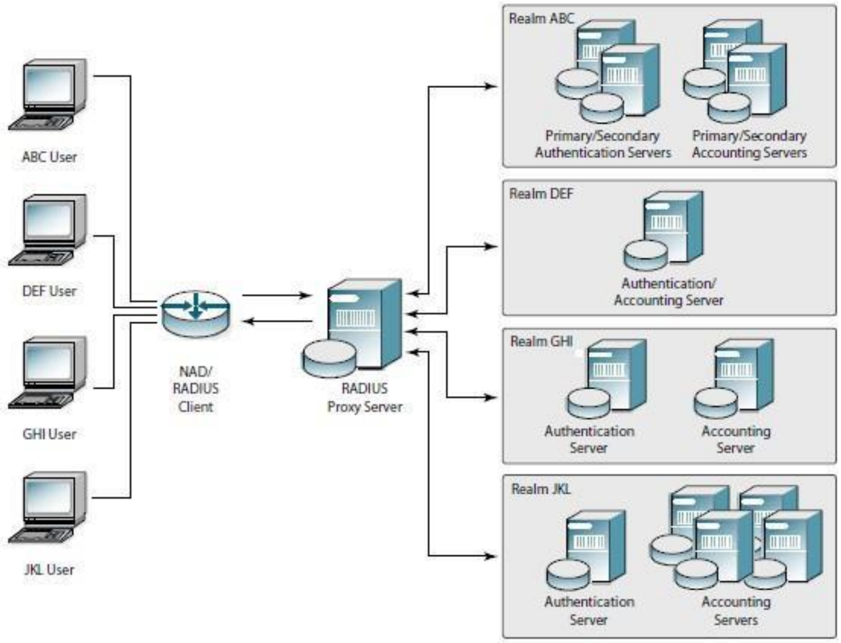
RADIUS accounting messages are proxy-forwarded as follows:

1. A RADIUS server receives an accounting request.
2. Depending on its configuration, the RADIUS server forwards the accounting message to a target server, records accounting attributes locally on the proxy server, or records the information in both places.
3. If the proxy server does not receive an acknowledgement of the forwarded packet, it periodically re-sends the packet according to its retry policy.

Proxy RADIUS Realms

Proxy RADIUS realms are pools of RADIUS servers to which Steel-Belted Radius can forward RADIUS requests. Proxy RADIUS realms can be configured to support workload distribution, redundancy, fault tolerance, retry policies, primary-secondary server roles, and separation of authentication and accounting responsibilities by server. For more information, see "Configuring a Proxy RADIUS Realm".

Figure 74: RADIUS Server and Proxy Realms



Target Selection within a Realm

For proxy RADIUS realms, after the destination realm is identified, Steel-Belted Radius must select a target within the realm. Target selection depends upon a number of factors, all of which you can set up in advance by editing the realm configuration files on the Steel-Belted Radius server: proxy.ini, radius.ini, filter.ini, and one RealmName.pro file per realm.

After the target is selected, Steel-Belted Radius matches the target name with a proxy entry in its database. Using the data in this entry (IP address, UDP port, shared secret) Steel-Belted Radius establishes a connection between itself and the target, and proxy-forwards the RADIUS request. Note that you can configure the realm so that all realm routing information and delimiters are stripped from the User-Name before forwarding.

The target processes the request as it normally would for RADIUS authentication or accounting. In the case of authentication, Steel-Belted Radius waits for a response from the target, then relays this response to its RADIUS client.

Message-Authenticator Support

The Message-Authenticator attribute enables Steel-Belted Radius to determine whether the packet received is from an actual proxy server. It might also sign the forward request.

Steel-Belted Radius can be configured to use the Message-Authenticator attribute when forwarding packets using proxy RADIUS. It can also be configured to validate or ignore the Message-Authenticator if included in the packets received.

Proxy Fast-Fail

During proxy forwarding, Steel-Belted Radius acts as the RADIUS client of another RADIUS server. Since RADIUS clients take responsibility for delivering RADIUS packets, all of them have a retry policy that determines how often and for how long they continue to try to deliver a packet until they receive the response that they expect from the RADIUS server.

This includes the Steel-Belted Radius server when it acts as the RADIUS client of a proxy RADIUS target server. Steel-Belted Radius provides a fast-fail option for proxy RADIUS realms. This fast-fail feature saves Steel-Belted

Radius from continuing to send packets to a target server that appears to be down temporarily. For example, if Steel-Belted Radius is sending a packet to a target and it is not getting the timely response it expects, it periodically tries to send the packet until it reaches the number of tries in its retry policy. If it still hasn't received a response from the target at that point, Steel-Belted Radius removes the target from the active list and places it on the fast-fail list.

Each time a request from a realm is received, Steel-Belted Radius sends a probing request to all fast-fail entries for this realm. No response is expected or required from the probes. No retry policy is followed. If a response to a probe is received, that target is removed from the fast-fail list. When the fast-fail timer expires for a target, it is placed back on the active list.

We strongly recommend that you specify a [FastFail] section in each proxy RADIUS realm configuration (.pro) file. The [FastFail] section permits you to fine-tune retry policies for individual realms, or for specific targets within realms. Any [FastFail] settings that you supply in a .pro file override the current ProxyFastFail setting.

The radius.ini file offers a ProxyFastFail setting for single-target proxy entries that are not a member of any realm. ProxyFastFail has an integer value, usually 1800. If a target remains on the fast-fail list longer than this number of seconds, it is automatically removed from the fast-fail list. If conditions warrant, a target might be returned to the fast-fail list at any time.

For information on configuring the radius.ini file to support the fast-fail feature, see the Steel-Belted Radius Reference Guide.

Static Proxy Accounting

Static proxy accounting allows you to send copies of certain types of accounting messages to proxy RADIUS realms, as well as to the normal routing of the original accounting message. The number of copies is not limited.

Static proxy accounting does not prevent the request from being dynamically routed for RADIUS accounting services based on User-Name decoration, DNIS number, or attribute mapping, nor does it prevent local logging or other accounting methods from occurring. If static proxy-forwarding fails (due to a lack of response from the target), this does not prevent the original RADIUS accounting request from being acknowledged.

An important function of static proxy accounting is to ensure that Accounting-On and Accounting-Off messages can be routed to realms. A NAD (RADIUS client) normally issues these accounting messages to its RADIUS server when it goes online (Accounting-On) and offline (Accounting-Off). In such cases, all connections previously made by this NAD are considered invalid, and the RADIUS server can free resources that it allocated to those sessions.

Static proxy accounting is necessary to deliver Accounting-On and Accounting-Off messages to realms, because these messages do not contain the User-Name or Called-Station-Id attributes that Steel-Belted Radius would normally use to route packets to realms.

For example, assume the original Access-Request, an authentication message, was used to determine the realm destination for both authentication and accounting for a particular session. The attribute used to route the Access-Request might have been the User-Name, the Called-Station-Id, or any other RADIUS attribute in the Access-Request, depending on how you have configured request routing for authentication messages.

Accounting packets for this same session can be matched with the realm destination only if the server knows which session is involved (as it does in Start, Stop, and Interim messages). The Accounting-On and Accounting-Off messages are independent of specific sessions, therefore, it is impossible to route them to realms without additional information.

By setting up static proxy accounting, and listing all realms as targets for Accounting-On and Accounting-Off messages, you can ensure that network information (such as NAD status) is sent to everyone who might require it.

Proxy AutoStop Feature

A user session can be removed from the Current Sessions table in ways other than the usual Accounting-Stop message from the NAD:

- An Accounting-On or Accounting-Off message received from the NAD causes all sessions originating from the NAD to be purged, as these messages signal that the NAD has been restarted or is going down.
- The administrator can remove users by means of the LDAP configuration interface (LCI).
- The administrator can remove users by means of the SBR Administrator application.

Termination information must be passed on if the users exist as proxied sessions on downstream RADIUS servers because these servers must free the resources previously allocated to the session(s), which have now been terminated. The Proxy AutoStop feature handles such cases. In addition, if you use the LCI command to free resources from the central administrative server, the appropriate messages are propagated so that the resources associated with the user in each of the downstream servers are automatically freed.

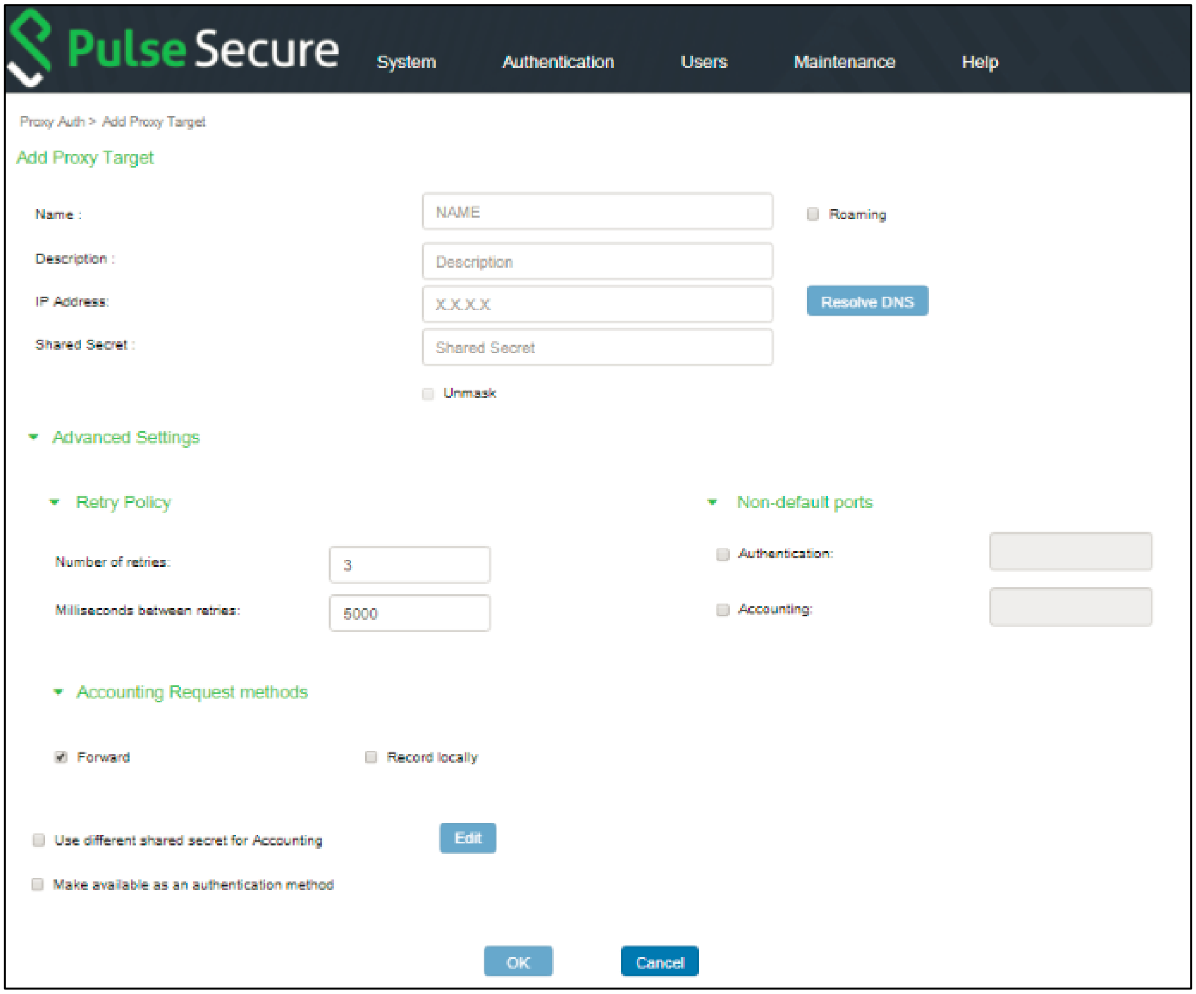
Adding a Proxy Target

This section explains how to set up proxy forwarding from the Steel-Belted Radius server (the proxy) to another RADIUS server (the target).

To add a proxy target:

1. Choose **Users > Proxy Auth > Proxy Targets** to open the Proxy Target page.
2. Click the **New** button on the SBR Administrator toolbar. The Add Proxy Target page (Figure 75: Add Proxy Target) appears.

Figure 75: Add Proxy Target Page



Proxy Auth > Add Proxy Target

Add Proxy Target

Name : NAME □ Roaming

Description : Description

IP Address: X.X.X.X Resolve DNS

Shared Secret : Shared Secret

□ Unmask

Advanced Settings

Retry Policy

Number of retries: 3

Milliseconds between retries: 5000

Non-default ports

□ Authentication:

□ Accounting:

Accounting Request methods

☑ Forward □ Record locally

□ Use different shared secret for Accounting Edit

□ Make available as an authentication method

OK Cancel

3. Enter the name of the proxy target in the **Name** field.

   The target name must not duplicate any other target name, realm name, or tunnel name in your Steel-Belted Radius configuration. The name you record for a proxy target is not used in processing; Steel-Belted Radius uses the proxy target's IP address to route RADIUS packets.
4. Enter a description for the proxy target in the **Description** field.

5. Enter the IP address or DNS name of the proxy target in the **IP Address** field.

   If you enter the DNS name of the proxy target then click Resolve DNS option, the SBR Administrator resolves the name you enter to an IP address.

6. Enter the shared secret for the proxy target in the **Shared Secret** field.

   Note: Shared secrets are case-sensitive.

   If you want the characters in the shared secret (rather than asterisks) to appear as you type, click the **Unmask** check box.

   The shared secret configured for the proxy target in Steel-Belted Radius must match the shared secret configured on the proxy target.
7. Specify how many times Steel-Belted Radius should try to reach the proxy target and how long to wait between

attempts in the **Number of retries and Milliseconds** between retries fields.

When Steel-Belted Radius acts as a proxy, it emulates the characteristics of a network access device. This includes the ability to retransmit a request if the first attempt does not get a timely response from the proxy target.

- The **Number of retries** field specifies the number of times a request is retransmitted if an acknowledgment from the target is not received; if the number of retries is exhausted, then the original request is rejected. By default, Steel-Belted Radius retries three times before giving up.
- The **Milliseconds between retries** field specifies the time interval between each retry in milliseconds (thousandths of a second). By default, Steel-Belted Radius waits 5000 milliseconds (5 seconds) between retries.

8. If the proxy target uses ports different from what the proxy intermediary uses for authentication or accounting, click the **Authentication or Accounting** check box and enter the port number you want Steel-Belted Radius to use when exchanging RADIUS authentication or accounting information with the proxy target.

   The port numbers configured for the proxy target in Steel-Belted Radius must match the port numbers configured on the proxy target. By default, Steel-Belted Radius uses port 1645 for authentication and port 1646 for accounting.

9. Specify whether you want accounting requests to be forwarded or recorded locally.

   - If you click the **Forward** check box, Steel-Belted Radius forwards the accounting transaction to the same proxy target that received the authentication transaction.
   - If you click the **Record locally** check box, Steel-Belted Radius logs the accounting transaction locally (regardless of whether an authentication request was forwarded to the proxy target.

   You can click both check boxes if you want accounting requests to be forwarded and logged locally.

10. If you want Steel-Belted Radius to use a different shared secret for accounting when communicating with the proxy target, click the **Use different shared secret for accounting** check box and click the **Edit** button to specify an accounting shared secret.

    Refer to "Maintaining an Accounting Shared Secret" for information on using the Edit Accounting Shared Secret page.

11. If you want to use a proxy target as an authentication method, click the **Make available as an authentication method check box**.

    If you enable this option, the name of the proxy target appears in the Authentication Methods tab of the Authentication Policies page as proxy:name. This is useful if you have user records defined on an older RADIUS server and you want to provide a seamless migration to Steel-Belted Radius. Using the older server as a proxy RADIUS target means that RADIUS requests that arrive addressed to this target are handled by Steel-Belted Radius automatically, without requiring end users to change their addressing conventions.

12. Click **OK**.

Note: If the proxy target that you are configuring is a member of a proxy RADIUS realm, you should ensure that the Make available as an authentication method check box is unchecked.

Ask the administrator at the target site to log into the target server's RADIUS configuration program and add

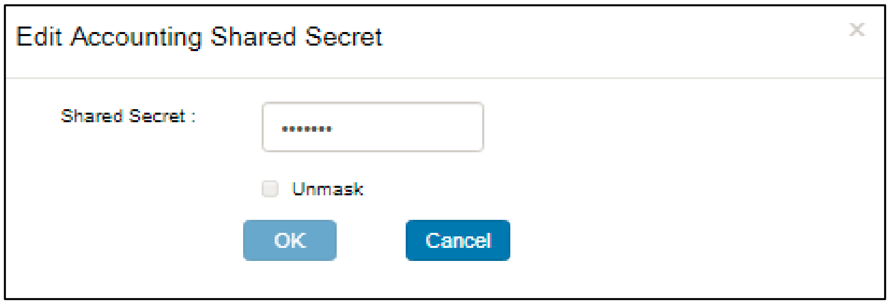
Steel-Belted Radius as a RADIUS client of the target server. You will need to provide this administrator with the IP address of the Steel-Belted Radius server.

Maintaining an Accounting Shared Secret

To specify a shared secret for accounting:

1. Click the Use **different shared secret for accounting** check box.
2. Click the **Edit** button.
3. When the Accounting Shared Secret page opens, enter the shared secret you want Steel-Belted Radius to use.

   Figure 76: Accounting Shared Secret Page

   Edit Accounting Shared Secret

   Shared Secret : ••••••••

   □ Unmask

   If you want the characters in the shared secret (rather than asterisks) to appear as you type, click the **Unmask** check box. Note that shared secrets are case-sensitive.

4. Click **OK.**

Deleting a Proxy Target

To remove a target server from the proxy target list:

1. In the Proxy Target page, select the target server you'd like to remove.
2. Click **Delete.**

Steel-Belted Radius as a Target

This section describes how to set up proxy forwarding from some other RADIUS server (the proxy) to the Steel-Belted Radius server (the target):

1. Set up the proxy as a RADIUS client of Steel-Belted Radius.

   Add the entry using the RADIUS Clients page. Specify the proxy's name, its IP address, and the shared secret that you want to use for encryption between the proxy and Steel-Belted Radius.

2. Ask the administrator at the target site to log into the proxy's RADIUS configuration program and set up Steel-Belted Radius as a proxy RADIUS target. You will need to provide this administrator with the IP address of the Steel-Belted Radius server.

   Note: Make sure that the same UDP port and shared secret are entered on both proxy and target sides.

Dictionaries when Steel-Belted Radius is the Target

When Steel-Belted Radius receives a proxy-forwarded packet, it consults its RADIUS client entry for that proxy server. The **Make/model** field of this entry determines which attribute dictionary Steel-Belted Radius uses.

At various different times, Steel-Belted Radius can receive requests from the same proxy server that have originated from different network access devices, possibly of different types. The single **Make/model** field that was entered for the proxy might not be adequate to handle the variety of RASs on the "other side" of the transaction.

One way to handle this problem is to add the originating network access devices to Steel-Belted Radius's list of RADIUS clients. Steel-Belted Radius can be configured to examine each proxy-forwarded packet for clues as to the make and model of the originating device. If clues are found, Steel-Belted Radius does everything it can to map this information to a vendor-specific dictionary, and uses this dictionary in preference to the one for the proxy.

Accepting Packets from Any Proxy

If you'd like Steel-Belted Radius to be able to accept proxy requests from any IP address, you can use the RADIUS Clients page to add a special entry called **<ANY>**, and specify a shared secret. The **<ANY>** entry permits forwarded requests from any proxy to be accepted, provided the shared secret is correct.

ℹ Note: This feature requires that proxies are configured to use the shared secret you provide in the <ANY> entry.

Proxy RADIUS as an Authentication Method

Any target proxy RADIUS server can be configured as a Steel-Belted Radius authentication method by enabling the **Make available as an authentication method** check box in the Add Proxy Target/Edit Proxy Target page.

A target server can be set up as an authentication method even if the end users do not know anything about the target. That is, a user does not need to log in using a decorated username such as User@TargetName to be authenticated by the target server.

If you prioritize the proxy: TargetName authentication method above the Native User authentication method in the authentication methods list, the user can log in as User and Steel-Belted Radius automatically sends the request to the target for authentication. The authentication succeeds if the UserName and password are stored on the target, but if not, Steel-Belted Radius reaches the Native User method eventually, and the user can then be authenticated.

This technique is useful as a migration path to Steel-Belted Radius from other RADIUS servers. You can set up Steel-Belted Radius as the proxy and the old RADIUS server as the target. After proxy authentication is enabled (in the Proxy Targets page) and prioritized (in the Authentication Policies page), Steel-Belted Radius can authenticate users against the old RADIUS server, either as an automatic "first choice" or as an alternative when authentication against the new server's "local" database fails.

Chapter 15
Administering RADIUS Tunnels via Legacy SBR Administrator

This chapter describes how to set up and administer RADIUS tunnels via legacy SBR administrator.

About RADIUS Tunnels

A tunnel is a uniquely secure type of remote connection. A tunnel passes data between a remote site and an enterprise site, providing an additional layer of encrypted protocol "wrapper" around the data. A tunnel offers authentication and encryption features that help secure the connection against network vandals and eavesdroppers. In addition, a tunnel can provide quality of service features such as guaranteed bandwidth.

ℹ Note: Steel-Belted Radius does not add tunnel functionality to your network. Steel-Belted Radius is able to support the authentication and accounting needs of any tunnels that you've already set up.

Administration and configuration of the tunnel happens at the remote site, since this is the side of the connection that requests remote access and opens the tunnel. An administrator at the remote site must configure the tunnel with various attributes: its destination IP address, what security protocols it supports, its password, and so on. These attributes are stored in a database to be retrieved when needed to set up a connection.

Storing tunnel attributes on a RADIUS server simplifies tunnel connections. At connection time, the tunnel is established by a network access device at the remote site. The NAD retrieves the tunnel configuration attributes from the RADIUS server and uses them to open the tunnel into the enterprise. After the tunnel is open, the user can be authenticated at the enterprise.

A RADIUS server is said to support tunnels if it has the ability to store and retrieve the configuration data that a NAD needs to open a tunnel. Steel-Belted Radius fully supports tunnels:

- Steel-Belted Radius can determine from the attributes in the incoming Access-Request whether the connection request involves a tunnel, and if so, which tunnel.
- Steel-Belted Radius can store and retrieve tunnel configuration data.
- Steel-Belted Radius can track the number of tunnels currently in use, compare to a maximum number, and refuse the connection if the number is exceeded.

Tunnel Authentication Sequence

1. Steel-Belted Radius receives an Access-Request message:
2. Steel-Belted Radius checks if the Access-Request contains a Called-Station-Id attribute. If it does, Steel-Belted Radius searches its database for a tunnel entry that contains the indicated telephone number in its Called-Station-Id list.

   If a match between the Called-Station-Id and a tunnel entry can be found, Steel-Belted Radius constructs an Access-Accept message using the Attributes list in the matching tunnel entry. It then returns the Access-Accept to the client NAD.

   ℹ Note: If realms are in use, Steel-Belted Radius also searches for this number in its realm configuration files. If a match is found, the Access-Request is routed to the realm, and the quest for a tunnel is abandoned. For this reason, it is important to ensure that DNIS numbers are unique across all tunnel entries and across all realm configuration files.

3. Steel-Belted Radius checks if the Access-Request contains a username in the form User<Delimiter>TunnelName or TunnelName<Delimiter>User. <Delimiter> is a single character that must match the server's tunnel delimiter character. The order of the realm name relative to the username must match the server's tunnel naming convention (prefix or suffix). Both of these values are determined per server (that is, all tunnels that use this server must follow the same conventions) by entering them in the Name Parsing tab of the Tunnels panel.

4. Steel-Belted Radius searches its database for a tunnel entry whose name matches the incoming TunnelName. If a match can be found, Steel-Belted Radius constructs an Access-Accept message using the Attributes list in the matching tunnel entry. It then returns the Access-Accept to the client NAD.

5. If Steel-Belted Radius was able to match the Access-Request with a tunnel entry, the NAD uses the attributes returned in the Access-Accept message to open a tunnel into the enterprise site. Authentication of the User-Name is attempted, usually at the enterprise site. If user authentication succeeds, the connection is complete. Otherwise, the user's connection request is denied.

6. If no matching tunnel entry was found in steps 1 or 2, Steel-Belted Radius concludes that a tunnel is not involved in making this connection. It then continues with its User-Name parsing sequence determine a destination for the authentication request.

The following is a wildcard example for IP Addresses:

```
NAS-IP-Address = 199.100.10.0
```

where NAS-IP-Address indicates any IP address on the 199.100.10.0 network.

Configuring Tunnel Support

To configure Steel-Belted Radius to support a tunnel, you must open the Tunnels panel in the SBR Administrator and add a tunnel entry.

A tunnel entry allows you to specify a list of connection Attributes such as the tunnel password, the IP address of the NAD at the enterprise site, encryption conventions to use, and so on. You can also enter the maximum number of tunnels that can be open at one time. You will need to coordinate with the administrator at the enterprise site to get some of this information.

Called Station Id

DNIS (Dialed Number Information Services) refers to a capability that many network access devices have to determine and use the telephone number that was dialed to make a connection request. The RADIUS standard supports DNIS by specifying the following attributes:

- Calling-Station-Id is the number from which the user originated the request.
- Called-Station-Id is the telephone number that was dialed to make the network connection.

When setting up a tunnel entry for the Steel-Belted Radius database, you can enter a telephone number or list of numbers in the Called Station Id list in the Tunnels panel. This list identifies **Called-Station-Id** attribute values that the server should expect to find in tunnel connection requests.

Dictionaries for Tunnel Support

The Tunnels panel allows you to create the Attributes list by selecting attributes from a drop-down list. The available selections include attributes from all standard and vendor-specific RADIUS dictionaries installed on the Steel-Belted Radius server.

When the server can accept a tunnel connection request, it consults the corresponding tunnel entry for the list of Attributes to return in the Access-Accept packet. Steel-Belted Radius always returns any standard RADIUS attributes that appear in the Attributes list. It also returns any vendor-specific attributes that are appropriate for the NAD that requested the tunnel connection. Vendor-specific attributes in the Attributes list that do not apply to the requesting NAD are ignored.

Concurrent Tunnel Connections

Steel-Belted Radius tracks the number of active connections for each tunnel. You can limit the number of concurrent connections that can be open through a specific tunnel. When a user requests a new connection through a tunnel, Steel-Belted Radius compares the number of active connections in a tunnel to the maximum number of connections: if a new connection would exceed the limit, Steel-Belted Radius rejects the additional connection.

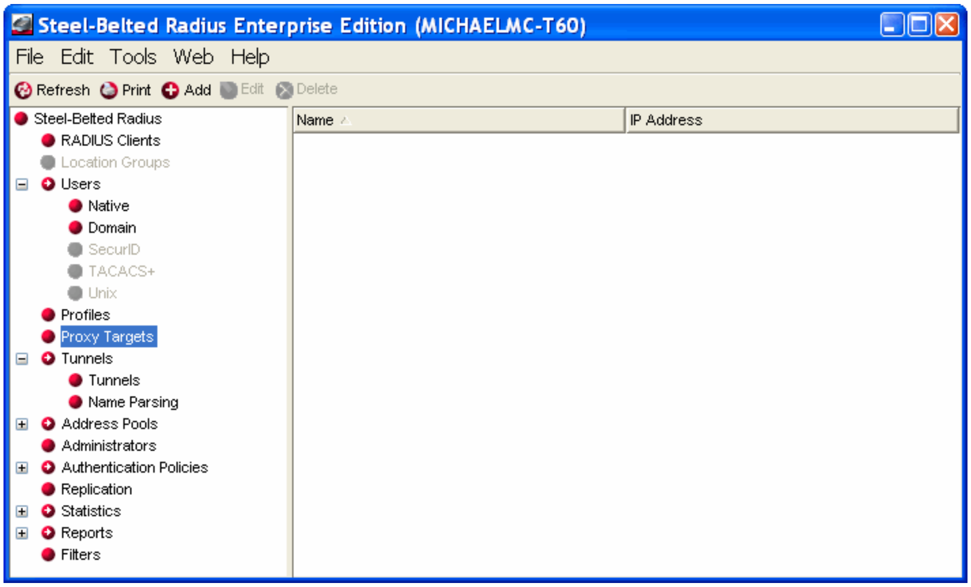
For concurrent connection limits to work, each NAD that can open a tunnel must be configured for RADIUS accounting and the same Steel-Belted Radius server must be specified for both authentication and accounting.

Note: Concurrent tunnel connections cannot be tracked across multiple Steel-Belted Radius servers without additional software extensions. Contact Pulse Secure for more information.

Configuring RADIUS Tunnels

The Tunnels panel (Figure 77: Tunnels Panel) lets you configure Steel-Belted Radius to support tunnels. When you add a tunnel entry, you are not creating a tunnel; you are enabling Steel-Belted Radius to support an existing tunnel's authentication and accounting needs and specifying how the server should parse tunnel names.

Figure 77: Tunnels Panel

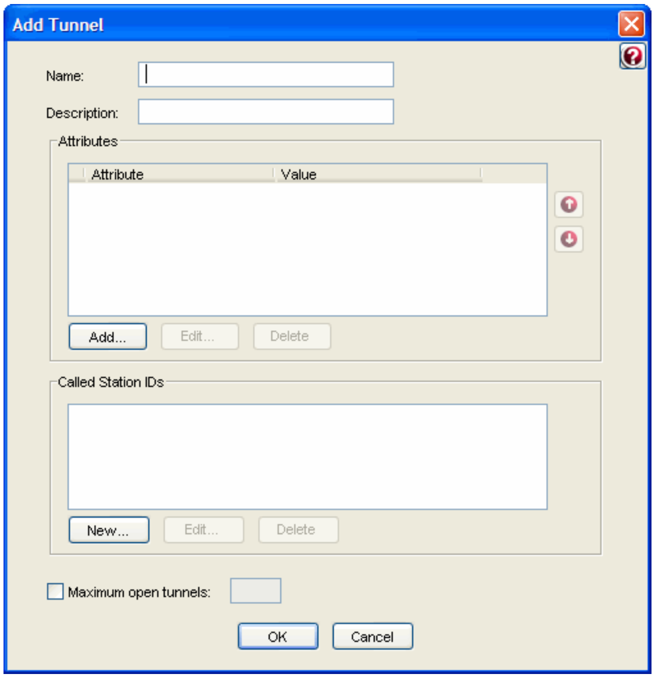


Adding a Tunnel

To add a tunnel entry:

1. Choose Tunnels > Tunnels in the sidebar.
2. Click the Add button in the Steel-Belted Radius toolbar.
   The Add Tunnel dialog opens.

Figure 78: Add Tunnel Dialog



3. Enter the name of the tunnel name in the Name field.

   Tunnel names do not need to match the actual node name of a client tunnel server. The name you assign to a tunnel must not match the name assigned to a proxy target, realm, or tunnel in your Steel-Belted Radius configuration.

4. Enter a description of the tunnel in the **Description** field.

   Tunnel descriptions are used only for administrative purposes and do not affect tunnel connections. This field is typically used to identify the user or organization that uses the tunnel.

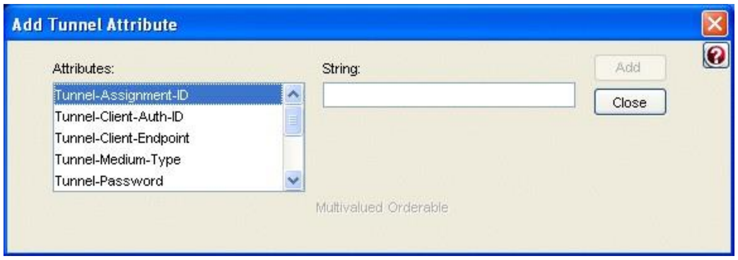
5. Associate attributes and values with the tunnel you are setting up.

   When a tunnel is used to make a connection, the attributes associated with the tunnel are filtered according to the make/model of the RADIUS client used to establish the connection.

   To associate attributes and values with a tunnel:

   a. Click the **Add** button below the **Attributes** list.
      The Add Tunnel Attribute dialog opens.

Figure 79: Add Tunnel Attribute Dialog



b. Select the attribute you want to add from the **Attributes** list.

c. Specify the string or IP address you want to use for the attribute value.

d. Click **Add**.

e. When you finish adding attributes for the tunnel, click **Close**.

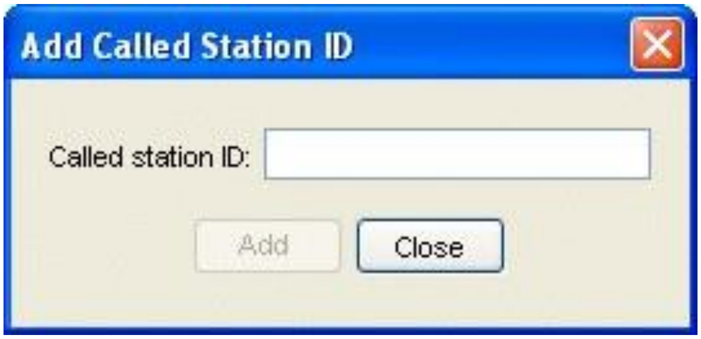
6. Optionally, specify one or more Called Station IDs for the tunnel.

   A Called Station ID is a telephone number that was dialed to make a network connection. The Called station ID list identifies the Called-Station-Id attribute values that the server expects to find in tunnel connection requests.

   To add one or more Called Station ID numbers for a tunnel:

   a. Click the **New** button to the right of the Called Station ID list. The Add Called Station ID dialog (Figure 80: Add Called Station ID Dialog) opens.

   Figure 80: Add Called Station ID Dialog



   b. Enter the number you want to use in the **Called station ID** field.

   c. Click **Add**.

   Repeat Steps a–c until you have added all called station IDs for the tunnel.

   d. When you are finished adding called station IDs, click **Close.**

7. If you want to limit the number of connections that can use the tunnel simultaneously, click the **Maximum open tunnels** check box and enter the **maximum number of tunnels** in the Maximum open tunnels field.

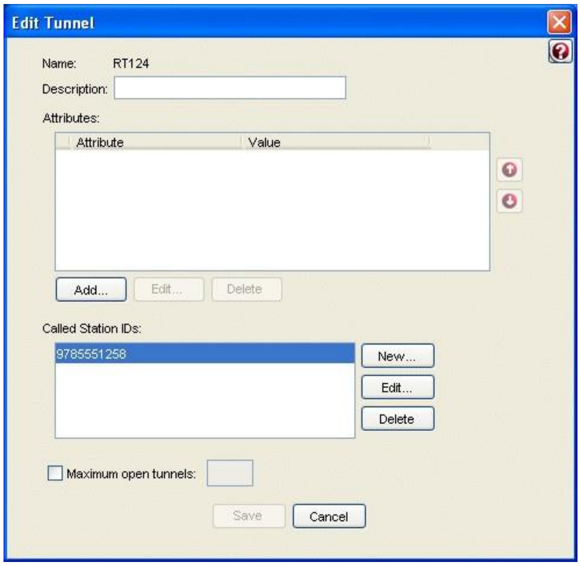
8. Click **OK.**

Editing a Tunnel

To edit a tunnel entry:

1. Choose **Tunnels > Tunnels** in the sidebar and select the tunnel you want to edit. Click **Edit.** The Edit Tunnel dialog appears.

Figure 81: Edit Tunnel Dialog



Name: RT124

Description:

Called Station IDs: 9785551258

☐ Maximum open tunnels:

2. Modify the settings for the tunnel as appropriate.

3. Refer to "Adding a Tunnel" for information on how to use the fields and controls on the Edit Tunnel dialog.

4. When you are finished, click **Save.**

Deleting a Tunnel

To delete a tunnel entry from the Steel-Belted Radius database:

1. Choose **Tunnels > Tunnels** in the sidebar.

2. Select the tunnel you want to delete and click the **Delete** button on the Steel-Belted Radius toolbar (or right-click the entry and choose **Delete** from the context menu that appears).

3. When the Confirm Delete dialog opens, click **Yes.**

Configuring Tunnel Name Parsing

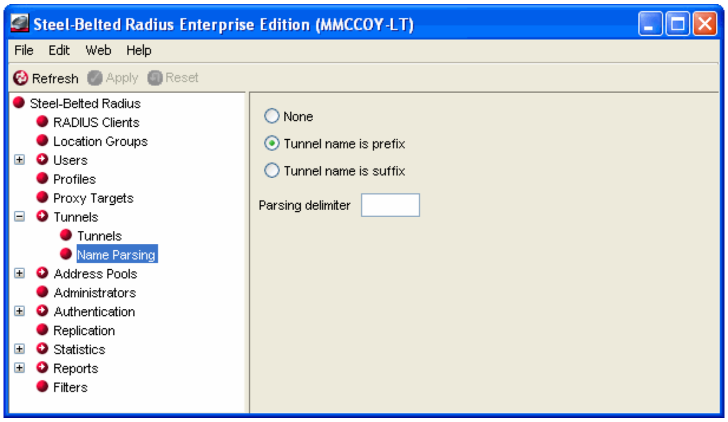
Tunnel name parsing lets Steel-Belted Radius determine whether the name string provided by a user includes a tunnel name by looking for the character configured as the delimiter for tunnel information. Tunnel name parsing options apply to all tunnels maintained by Steel-Belted Radius. You cannot set name parsing options for individual tunnels.

To configure tunnel name parsing:

1. Choose Tunnels > Name Parsing in the sidebar.

The Name Parsing dialog appears.

Figure 82: Tunnel Panel: Name Parsing Tab



○ None

● Tunnel name is prefix

○ Tunnel name is suffix

Parsing delimiter

2. Click one of the following radio buttons:

   - **None—**Tunnel name parsing is not supported. If you choose this option, the tunnel authentication sequence is bypassed for each Access-Request; the server uses the standard username/password authentication sequence only.
   - **Tunnel name is prefix—**If the tunnel delimiter character is detected, the User-Name is assumed to be TunnelName<PrefixDelimiter>User.
   - **Tunnel name is suffix—**If the tunnel delimiter character is detected, the User-Name is assumed to be User<SuffixDelimiter>TunnelName.

The option you choose applies to all tunnels defined on the server.

3. If you clicked T**unnel name is prefix** or **Tunnel name is suffix**, use the Parsing delimiter field to specify the character used to separate the tunnel name and the username.

The default delimiter character for tunnel name parsing is @.

Note: Choose different delimiter characters and different prefix/suffix name parsing conventions for tunnels and for proxies or realms.

Chapter 16
Administering RADIUS Tunnels via WebGUI

This chapter describes how to set up and administer RADIUS tunnels via WebGUI.

About RADIUS Tunnels

A tunnel is a uniquely secure type of remote connection. A tunnel passes data between a remote site and an enterprise site, providing an additional layer of encrypted protocol "wrapper" around the data. A tunnel offers authentication and encryption features that help secure the connection against network vandals and eavesdroppers. In addition, a tunnel can provide quality of service features such as guaranteed bandwidth.

Note: Steel-Belted Radius does not add tunnel functionality to your network. Steel-Belted Radius is able to support the authentication and accounting needs of any tunnels that you've already set up.

Administration and configuration of the tunnel happens at the remote site, since this is the side of the connection that requests remote access and opens the tunnel. An administrator at the remote site must configure the tunnel with various attributes: its destination IP address, what security protocols it supports, its password, and so on. These attributes are stored in a database to be retrieved when needed to set up a connection.

Storing tunnel attributes on a RADIUS server simplifies tunnel connections. At connection time, the tunnel is established by a network access device at the remote site. The NAD retrieves the tunnel configuration attributes from the RADIUS server and uses them to open the tunnel into the enterprise. After the tunnel is open, the user can be authenticated at the enterprise.

A RADIUS server is said to support tunnels if it has the ability to store and retrieve the configuration data that a NAD needs to open a tunnel. Steel-Belted Radius fully supports tunnels:

- Steel-Belted Radius can determine from the attributes in the incoming Access-Request whether the connection request involves a tunnel, and if so, which tunnel.
- Steel-Belted Radius can store and retrieve tunnel configuration data.
- Steel-Belted Radius can track the number of tunnels currently in use, compare to a maximum number, and refuse the connection if the number is exceeded.

Tunnel Authentication Sequence

1. Steel-Belted Radius receives an Access-Request message:
2. Steel-Belted Radius checks if the Access-Request contains a Called-Station-Id attribute. If it does, Steel-Belted Radius searches its database for a tunnel entry that contains the indicated telephone number in its Called-Station-Id list.

   If a match between the Called-Station-Id and a tunnel entry can be found, Steel-Belted Radius constructs an Access-Accept message using the Attributes list in the matching tunnel entry. It then returns the Access-Accept to the client NAD.

   Note: If realms are in use, Steel-Belted Radius also searches for this number in its realm configuration files. If a match is found, the Access-Request is routed to the realm, and the quest for a tunnel is abandoned. For this reason, it is important to ensure that DNIS numbers are unique across all tunnel entries and across all realm configuration files.

3. Steel-Belted Radius checks if the Access-Request contains a username in the form User<Delimiter>TunnelName or TunnelName<Delimiter>User. <Delimiter> is a single character that must match the server's tunnel delimiter character. The order of the realm name relative to the username must match the server's tunnel naming convention (prefix or suffix). Both of these values are determined per server (that is, all tunnels that use this server must follow the same conventions) by entering them in the Name Parsing tab of the Tunnels page.

4. Steel-Belted Radius searches its database for a tunnel entry whose name matches the incoming TunnelName. If a match can be found, Steel-Belted Radius constructs an Access-Accept message using the Attributes list in the matching tunnel entry. It then returns the Access-Accept to the client NAD.

5. If Steel-Belted Radius was able to match the Access-Request with a tunnel entry, the NAD uses the attributes returned in the Access-Accept message to open a tunnel into the enterprise site. Authentication of the User-Name is attempted, usually at the enterprise site. If user authentication succeeds, the connection is complete. Otherwise, the user's connection request is denied.

6. If no matching tunnel entry was found in steps 1 or 2, Steel-Belted Radius concludes that a tunnel is not involved in making this connection. It then continues with its User-Name parsing sequence determine a destination for the authentication request.

The following is a wildcard example for IP Addresses:

NAS-IP-Address = 199.100.10.0

where NAS-IP-Address indicates any IP address on the 199.100.10.0 network.

Configuring Tunnel Support

To configure Steel-Belted Radius to support a tunnel, you must open the Tunnels page in the SBR Administrator and add a tunnel entry.

A tunnel entry allows you to specify a list of connection Attributes such as the tunnel password, the IP address of the NAD at the enterprise site, encryption conventions to use, and so on. You can also enter the maximum number of tunnels that can be open at one time. You will need to coordinate with the administrator at the enterprise site to get some of this information.

Called Station Id

DNIS (Dialed Number Information Services) refers to a capability that many network access devices have to determine and use the telephone number that was dialed to make a connection request. The RADIUS standard supports DNIS by specifying the following attributes:

- Calling-Station-Id is the number from which the user originated the request.
- Called-Station-Id is the telephone number that was dialed to make the network connection.

When setting up a tunnel entry for the Steel-Belted Radius database, you can enter a telephone number or list of numbers in the Called Station Id list in the Tunnels page. This list identifies **Called-Station-Id** attribute values that the server should expect to find in tunnel connection requests.

Dictionaries for Tunnel Support

The Tunnels page allows you to create the Attributes list by selecting attributes from a drop-down list. The available selections include attributes from all standard and vendor-specific RADIUS dictionaries installed on the Steel-Belted Radius server.

When the server can accept a tunnel connection request, it consults the corresponding tunnel entry for the list of Attributes to return in the Access-Accept packet. Steel-Belted Radius always returns any standard RADIUS attributes that appear in the Attributes list. It also returns any vendor-specific attributes that are appropriate for the NAD that requested the tunnel connection. Vendor-specific attributes in the Attributes list that do not apply to the requesting NAD are ignored.

Concurrent Tunnel Connections

Steel-Belted Radius tracks the number of active connections for each tunnel. You can limit the number of concurrent connections that can be open through a specific tunnel. When a user requests a new connection through a tunnel, Steel-Belted Radius compares the number of active connections in a tunnel to the maximum number of connections: if a new connection would exceed the limit, Steel-Belted Radius rejects the additional connection.

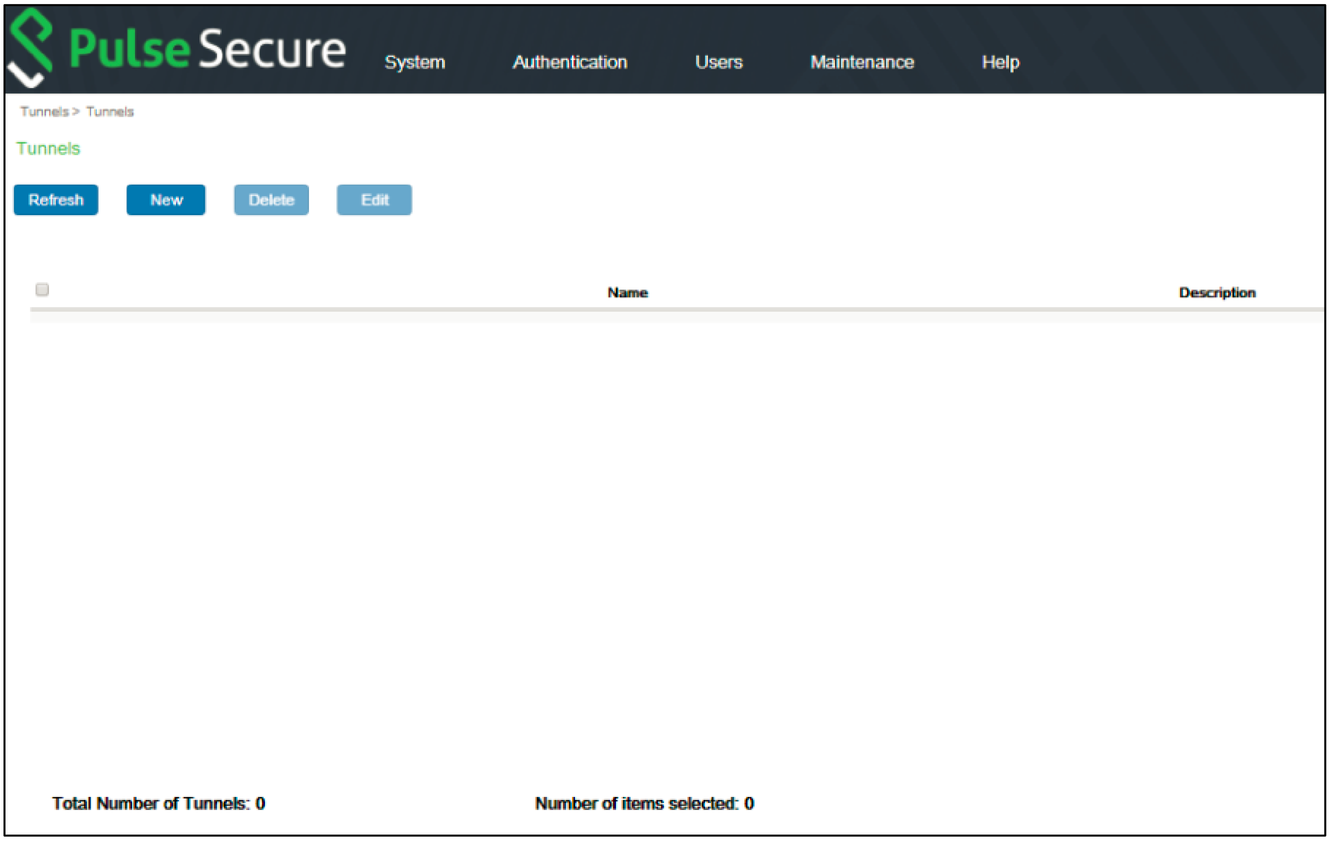
For concurrent connection limits to work, each NAD that can open a tunnel must be configured for RADIUS accounting and the same Steel-Belted Radius server must be specified for both authentication and accounting.

Note: Concurrent tunnel connections cannot be tracked across multiple Steel-Belted Radius servers without additional software extensions. Contact Pulse Secure for more information.

Configuring RADIUS Tunnels

The Tunnels page (Figure 83: Tunnels Page) lets you configure Steel-Belted Radius to support tunnels. When you add a tunnel entry, you are not creating a tunnel; you are enabling Steel-Belted Radius to support an existing tunnel's authentication and accounting needs and specifying how the server should parse tunnel names.

Figure 83: Tunnels Page



Total Number of Tunnels: 0

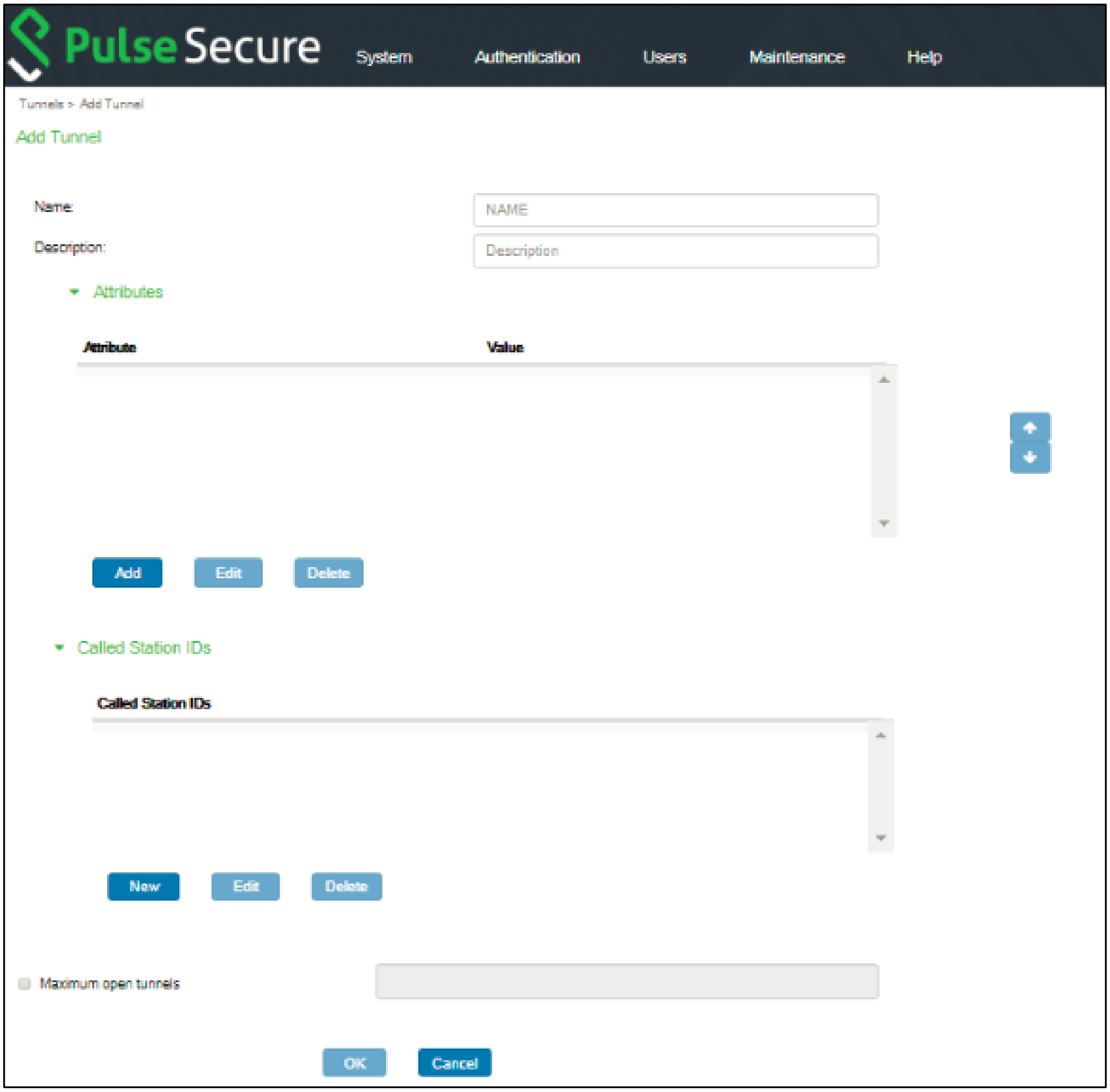
Number of items selected: 0

Adding a Tunnel

To add a tunnel entry:

1. Choose System > Tunnels > Tunnels.
2. Click the New button.
   The Add Tunnel page opens.

Figure 84: Add Tunnel Page



3. Enter the name of the tunnel name in the Name field.

   Tunnel names do not need to match the actual node name of a client tunnel server. The name you assign to a tunnel must not match the name assigned to a proxy target, realm, or tunnel in your Steel-Belted Radius configuration.

4. Enter a description of the tunnel in the **Description** field.

   Tunnel descriptions are used only for administrative purposes and do not affect tunnel connections. This field is typically used to identify the user or organization that uses the tunnel.

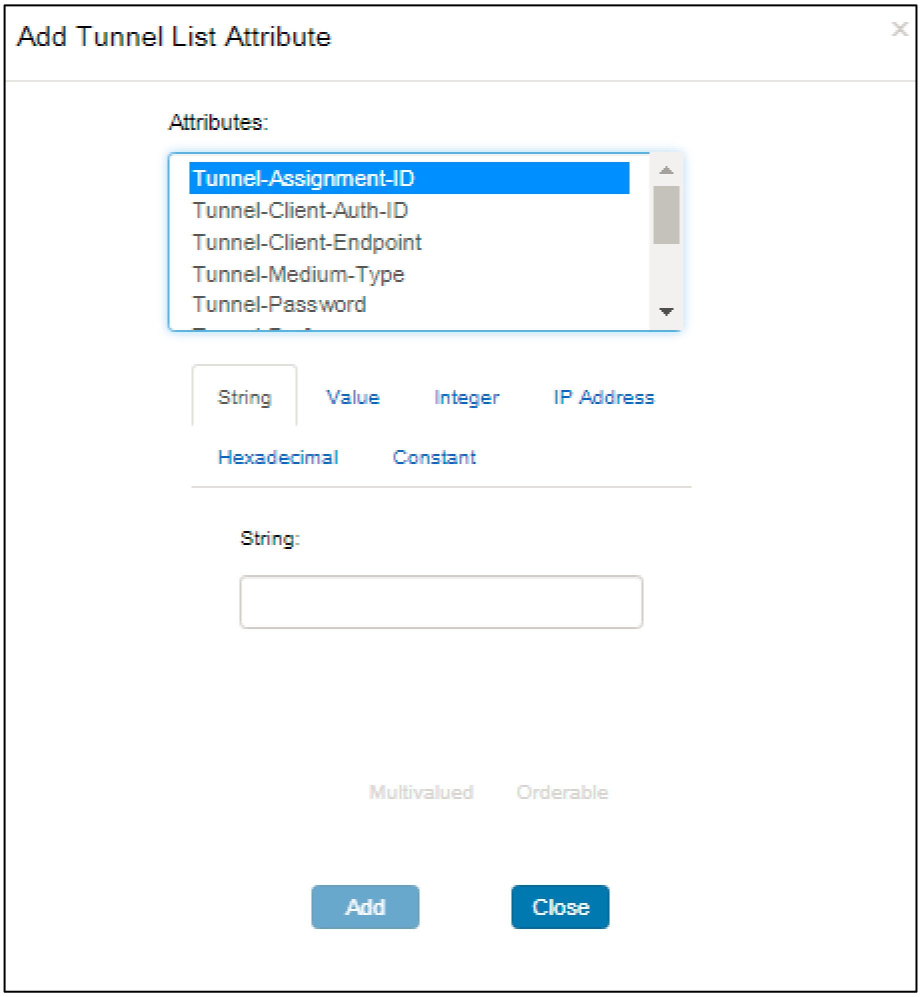
5. Associate attributes and values with the tunnel you are setting up.

   When a tunnel is used to make a connection, the attributes associated with the tunnel are filtered according to the make/model of the RADIUS client used to establish the connection.

To associate attributes and values with a tunnel:

a. Click the **Add** button below the **Attributes** list.
   The Add Tunnel Attribute page opens.

Figure 85: Add Tunnel Attribute Page



Add Tunnel List Attribute

Attributes:

- Tunnel-Assignment-ID
- Tunnel-Client-Auth-ID
- Tunnel-Client-Endpoint
- Tunnel-Medium-Type
- Tunnel-Password

String:

b. Select the attribute you want to add from the **Attributes** list.

c. Specify the string or IP address you want to use for the attribute value.

d. Click **Add**.

e. When you finish adding attributes for the tunnel, click **Close**.

6. Optionally, specify one or more Called Station IDs for the tunnel.

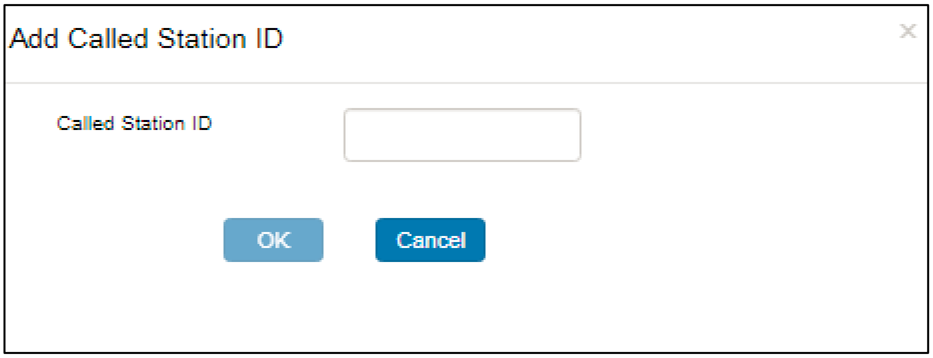
   A Called Station ID is a telephone number that was dialed to make a network connection. The Called station ID list identifies the Called-Station-Id attribute values that the server expects to find in tunnel connection requests.

   To add one or more Called Station ID numbers for a tunnel:

   a. Click the **New** button to the right of the Called Station ID list. The Add Called Station ID page

(Figure 86: Add Called Station ID Page) opens.

Figure 86: Add Called Station ID Page



Add Called Station ID

Called Station ID

b. Enter the number you want to use in the **Called station ID** field.

c. Click **Add**.

Repeat Steps a–c until you have added all called station IDs for the tunnel.

d. When you are finished adding called station IDs, click **Close.**

7. If you want to limit the number of connections that can use the tunnel simultaneously, click the **Maximum open tunnels** check box and enter the **maximum number of tunnels** in the Maximum open tunnels field.

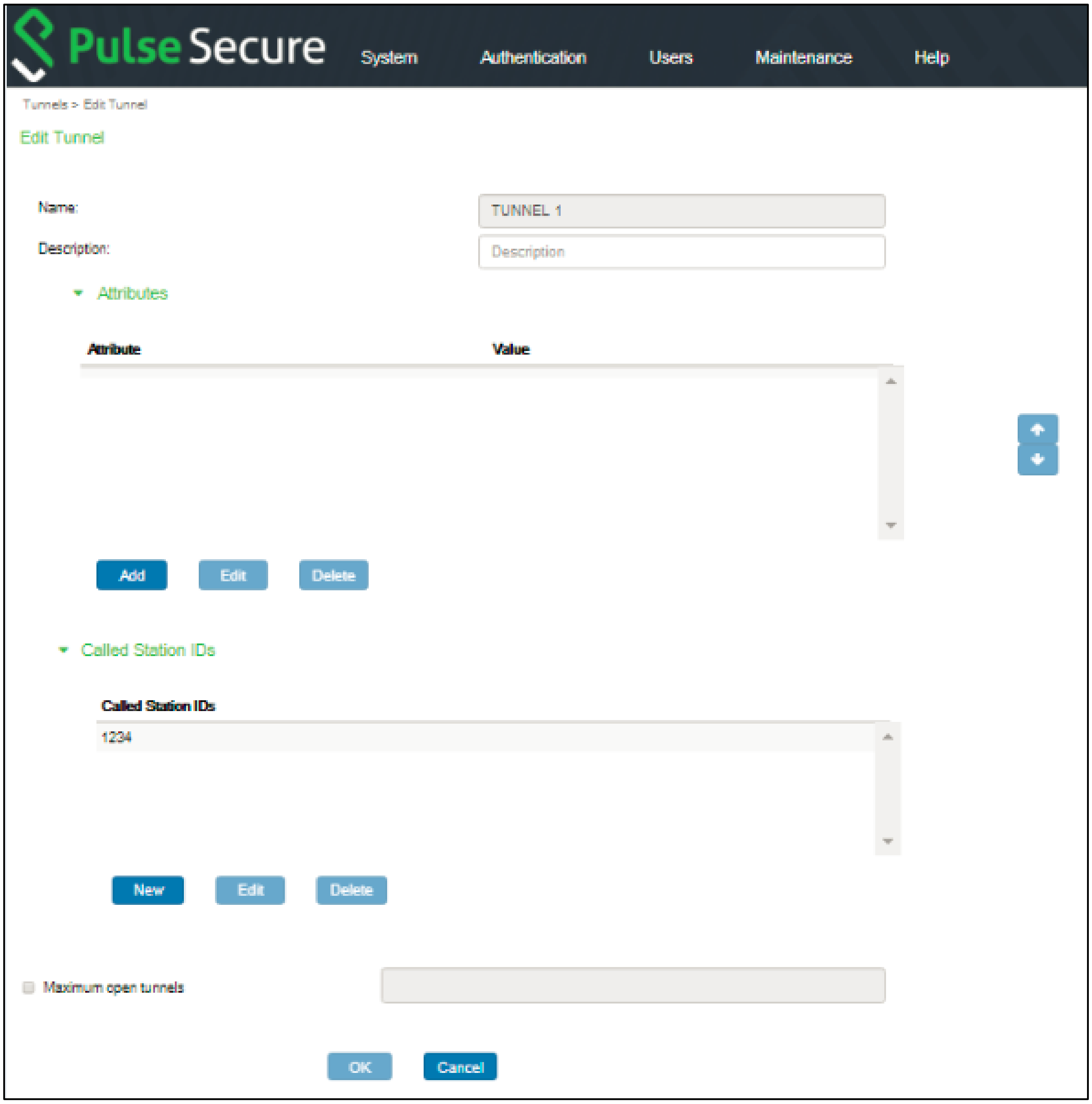
8. Click **OK.**

Editing a Tunnel

To edit a tunnel entry:

1. Choose **System > Tunnels > Tunnels** and select the tunnel you want to edit. Click **Edit.** The Edit Tunnel page appears.

Figure 87: Edit Tunnel Page



Tunnels > Edit Tunnel

Edit Tunnel

Name: TUNNEL 1

Description: Description

Attributes

Attribute	Value

Called Station IDs

Called Station IDs
1234

□ Maximum open tunnels

2. Modify the settings for the tunnel as appropriate.

3. Refer to "Adding a Tunnel" for information on how to use the fields and controls on the Edit Tunnel dialog.

4. When you are finished, click **Save.**

Deleting a Tunnel

To delete a tunnel entry from the Steel-Belted Radius database:

1. Choose **System > Tunnels > Tunnels.**

2. Select the tunnel you want to delete and click the **Delete** button on the Tunnels Page.

3. When the Confirm Delete page opens, click **Yes.**

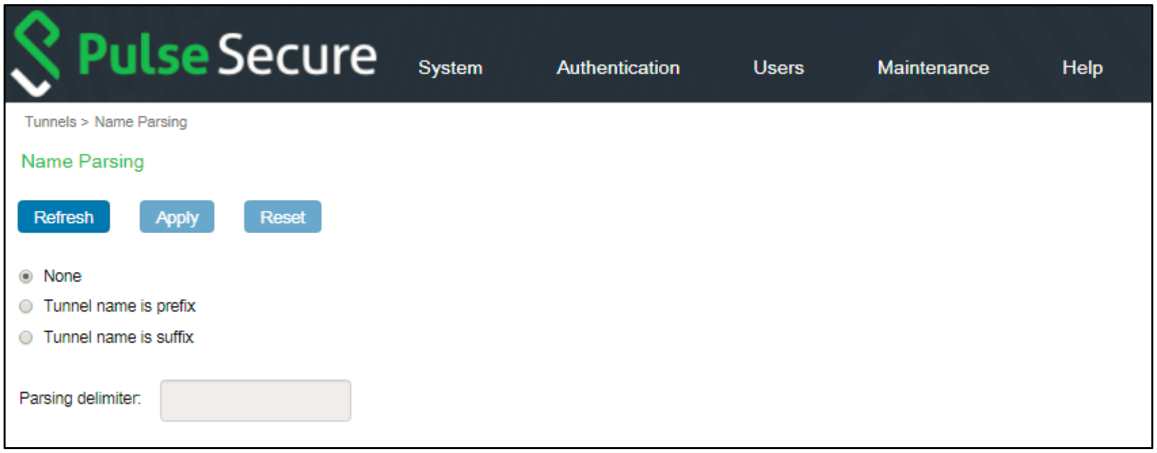
Configuring Tunnel Name Parsing

Tunnel name parsing lets Steel-Belted Radius determine whether the name string provided by a user includes a tunnel name by looking for the character configured as the delimiter for tunnel information. Tunnel name parsing options apply to all tunnels maintained by Steel-Belted Radius. You cannot set name parsing options for individual tunnels.

To configure tunnel name parsing:

1. Choose System > Tunnels > Name Parsing.
   The Name Parsing page appears.

Figure 88: Tunnel Page: Name Parsing Tab



● None
○ Tunnel name is prefix
○ Tunnel name is suffix

Parsing delimiter:

2. Click one of the following radio buttons:

   - **None—**Tunnel name parsing is not supported. If you choose this option, the tunnel authentication sequence is bypassed for each Access-Request; the server uses the standard username/password authentication sequence only.

   - **Tunnel name is prefix—**If the tunnel delimiter character is detected, the User-Name is assumed to be TunnelName<PrefixDelimiter>User.

   - **Tunnel name is suffix—**If the tunnel delimiter character is detected, the User-Name is assumed to be User<SuffixDelimiter>TunnelName.

   The option you choose applies to all tunnels defined on the server.

3. If you clicked **Tunnel name is prefix** or **Tunnel name is suffix**, use the Parsing delimiter field to specify the character used to separate the tunnel name and the username.

   The default delimiter character for tunnel name parsing is @.

   Note: Choose different delimiter characters and different prefix/suffix name parsing conventions for tunnels and for proxies or realms.

Chapter 17

Administering Address Pools via Legacy SBR Administrator

This chapter describes how to set up IPv4 and IPX address pools. Steel-Belted Radius does not support IPv6 address pools via legacy SBR administrator.

Note: Please contact Pulse Secure Global Support Center If you need address pools larger than 65,535 (216) addresses.

Address Pool Files

The following files establish settings for IP and IPX address pools. For more information about these files, refer to the Steel-Belted Radius Reference Guide.

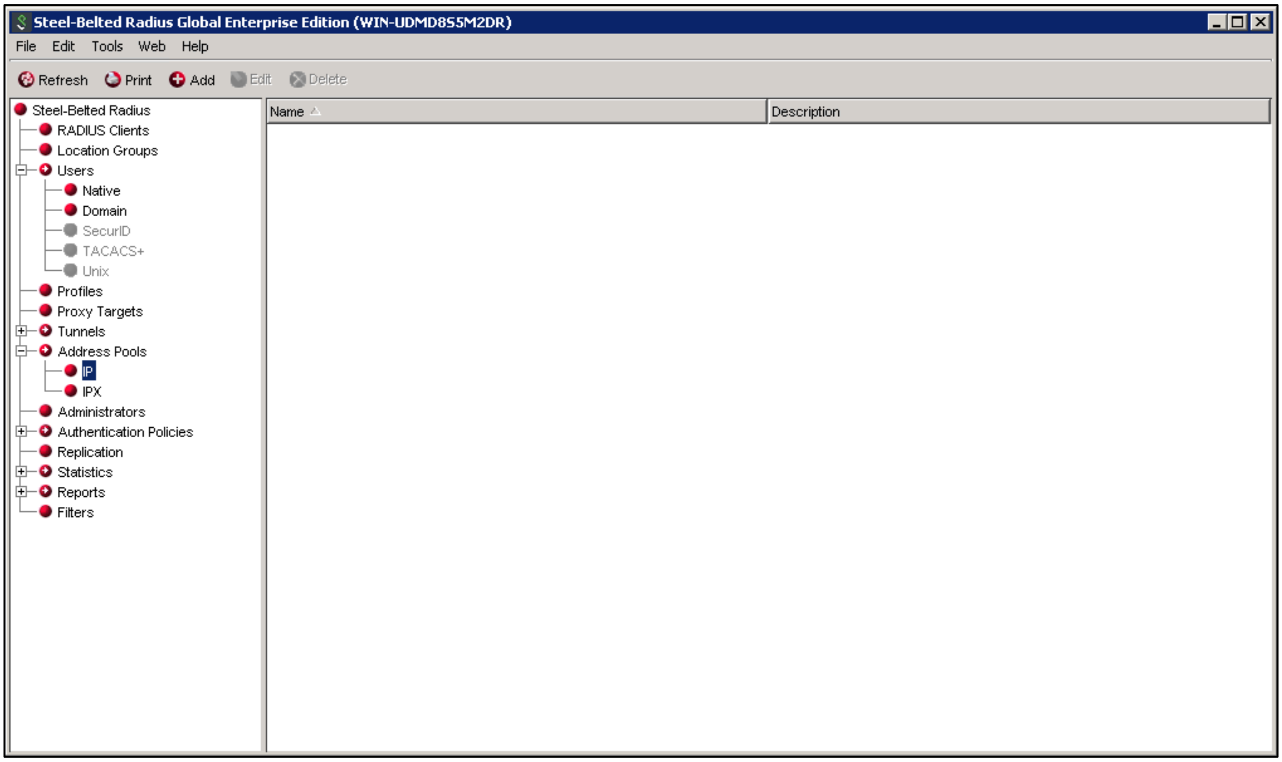
Table 24: Address Pool Files

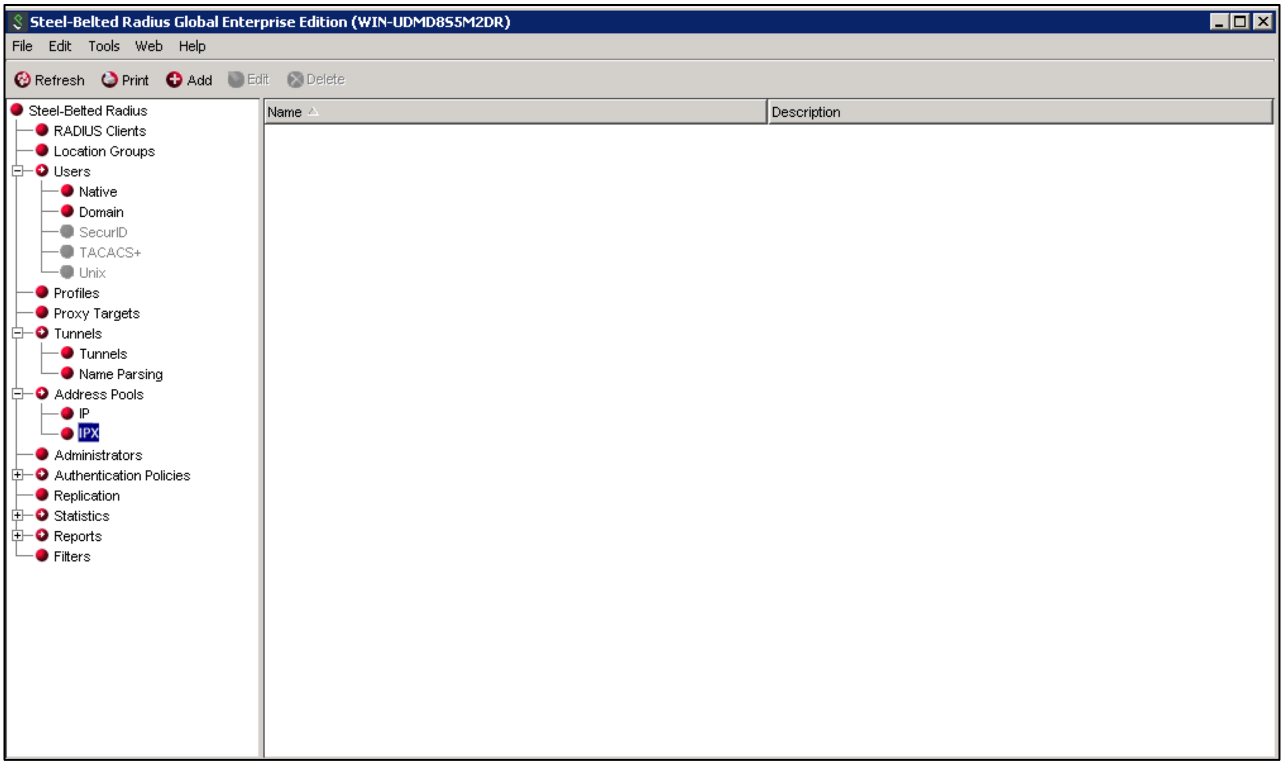
File Name	Function
dhcp.ini	Configures DHCP address pools so that IP addresses can be assigned from a backend DHCP server.
pool.dhc	Configures specific DHCP address pools, where pool is the name of an address pool listed in dhcp.ini.
radius.ini	Specifies (among other things) the suffixes used to set up NAD-specific IP pools.

Setting Up IP Address Pools

The IP Address Pool and IPX Address Pool panels Figure 89: IP Address Pools Panel and IPX Address Pools Panel Adding an) allow you to set up one or more pools out of which unique IPv4 or IPX addresses are assigned as users require them. Each address pool consists of a list of one or more ranges of addresses.

Figure 89: IP Address Pools Panel and IPX Address Pools Panel Adding an





Adding an IPv4 Address Pool

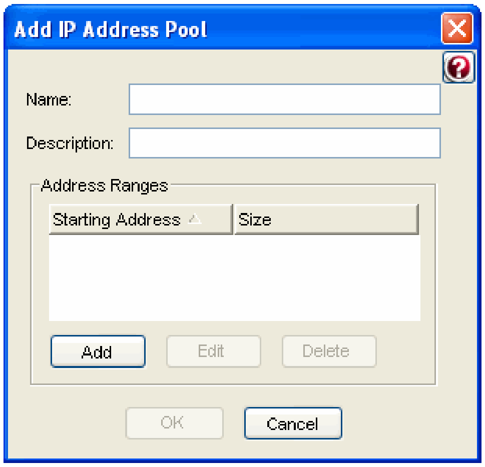
An IP address pool consists of one or more ranges of IPv4 addresses. You can add or delete ranges and set an optional description for each address pool.

To add an IP address pool:

1. Choose **Address Pools > IP** in the sidebar. The IP Address Pools panel appears.
2. Click the **Add** button in the toolbar.

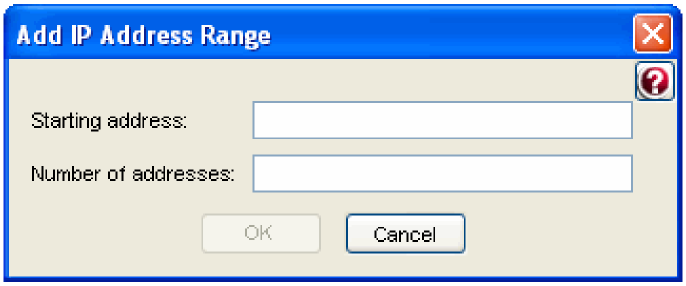
   The Add IP Address Pool dialog appears.

Figure 90: Add IP Address Pool Dialog



3. Enter the name of the IP address pool in the Name field.
4. Optionally, enter a description of the address pool in the Description field.
5. Identify the address range or ranges in the IP address pool.
   a. Click the Add button below the Address Ranges list.
   The Add IP Address Range dialog opens.

   Figure 91: Add IP Address Range Dialog

   b. Enter the first address in the **Starting address** field.
   c. Enter the number of addresses in the address range in the **Number of addresses** field.
   d. Click **Add**.
   e. Repeat steps a–d for each address range in the IP address pool. When you are finished, click **Close.**
6. Click **OK.**

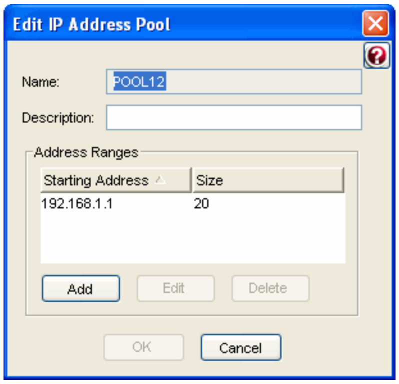
Editing an IP Address Pool

To edit an IP address pool:

1. Choose **Address Pools > IP** in the sidebar.
2. Select the entry you want to modify and click the **Edit** button (or right-click the entry and choose **Edit**).

   The Edit IP Address Pool dialog appears.

Figure 92: Edit IP Address Pool Dialog



Edit IP Address Pool

Name: POOL12

Description:

Address Ranges

Starting Address	Size
192.168.1.1	20

3. Modify the settings for the address pool as needed.
   - To add an address range to the address pool, click the **Add** button and specify the starting address and number of addresses in the range.
   - To modify an address range, select it and click the **Edit** button.
   - To delete an address range from the address pool, select it and click the **Delete** button.
4. When you are finished, click **Save**.

Removing an IP Address Pool

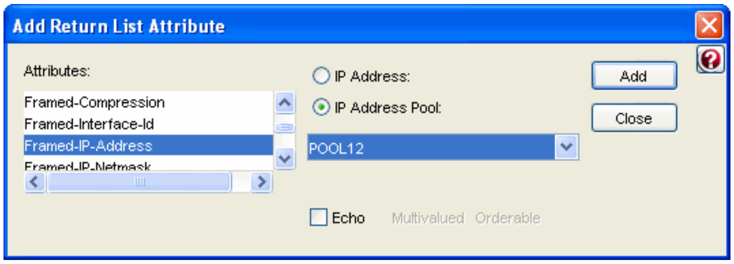
To delete an IP address pool:

1. Choose **Address Pools > IP** in the sidebar.
2. Select the entry you want to remove and click the **Delete** button (or right-click the entry and choose **Delete**).
3. When you are prompted to confirm the deletion, click **Yes**.

Specifying an IP Address Pool for User/Profile Records

The Framed-IP-Address return list attribute controls how the server assigns an IP address to a user making a connection. When you add or edit the Framed-IP-Address attribute in the Users or Profiles dialog, the Add Attribute dialog (Figure 93: Editing the Framed-IP-Address) allows you to choose an IP address pool instead of specifying an IP address.

Figure 93: Editing the Framed-IP-Address



Service-Level IP Address Pools

Steel-Belted Radius allows you to define a set of suffixes that define categories of IP address pools. For example, a pool category might correspond to the kinds of services available to users in that category. You might decide to define categories called Bronze, Silver, and Gold to identify different packet routing priorities.

To create a set of service-level address pools:

1. Define suffixes for the various service-level address pools in the [IPPoolSuffixes] section of radius.ini. For example:

   [IPPoolSuffixes]

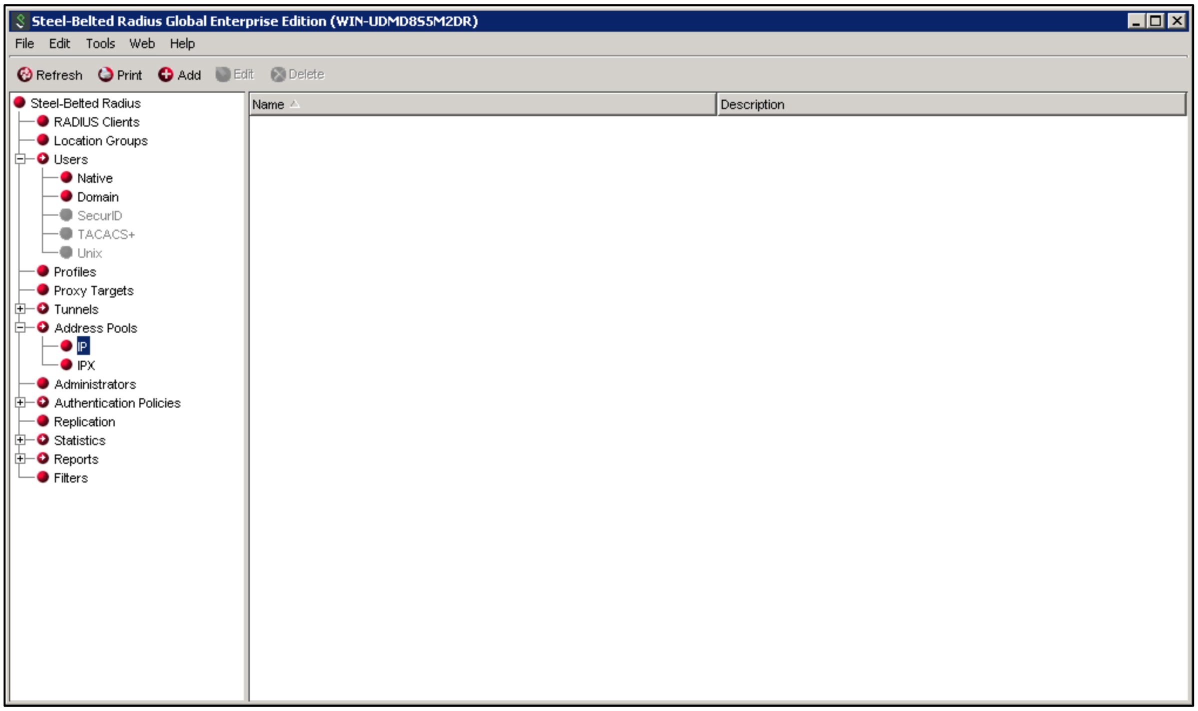
   -GOLD

   -SILVER

   -BRONZE

2. Define IP address pools using the suffixes configured in the [IPPoolSuffixes] section of radius.ini

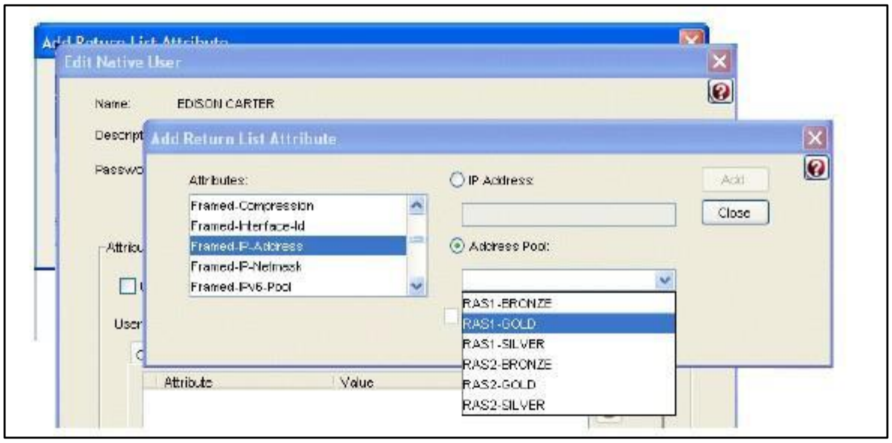
Figure 94: Service Level Suffixes in the IP Address Pools Panel



3. Associate the new IP address pool with the appropriate NAD by use of **IP Address** Pool field on the RADIUS Clients panel.

4. Assign a user to a NAD-specific IP address pool and suffix in the Users panel or the Profiles panel.

Figure 95: Associating IP Address Pools with RADIUS Clients



Name: EDISON CARTER

If user EDISON CARTER, who has been assigned to <RAS>-GOLD, logs into RAS1, he receives an IP address from the RAS1-GOLD address pool. If he logs into RAS2, he receives an address from the RAS2-GOLD address pool. If, however, he logs into RAS3 but RAS3-GOLD has not been defined in the IP Pools dialog, he is not assigned an IP address.

Specifying IP Address Assignment from a DHCP Server

IP addresses can be assigned from a backend DHCP server, rather than from a standard IP address pool. DHCP address pools function like internal address pools—Framed-IP-Address can be allocated from any address pool, either internal or DHCP.

DHCP address pools are defined in the dhcp.ini file and initialization files with the extension .dhc.

In addition, each DHCP address pool must be enabled by adding a placeholder IP address pool in the SBR Administrator. This placeholder pool should have the same name as the DHCP pool, and should have an empty list of address ranges. The placeholder pool allows the DHCP pool to appear in lists presented by the SBR Administrator, so it can be selected into an attribute.

When an IP address must be assigned from a DHCP pool during an Access-Request, DHCP DISCOVER and REQUEST messages are issued to trigger the allocation of an address. When an accounting Stop ends the session, DHCP RELEASE is issued to the server that allocated the address. Upon receipt of an accounting INTERIM request, a DHCP REQUEST message is issued to the server that allocated the address, attempting to extend the lease. If the server is specified as a broadcast address, DHCP failover occurs if the primary DHCP server go down.

DHCP leases can be acquired, extended, and released by different servers. The server that acquires the lease adds all the information for extending and releasing the lease to the Class attribute.

Flexible configuration features allow RADIUS attributes to be mapped to DHCP options. Therefore, information from a RADIUS request can be provided to the DHCP server, and information returned from the DHCP server can be returned to the network access device.

During authentication, if an address is assigned from a pool, the pool name must refer to either a DHCP pool or an internal pool. If the pool name is not found, the request is rejected.

Address Allocation

During address allocation, a DISCOVER message is issued. If an OFFER is received from a DHCP server and the offered lease time meets the minimum lease time requirements, the server issues a REQUEST message. If an ACK message is received, the allocated address is returned in the Access-Accept.

In addition to the options required for normal DHCP operation, additional options in the DHCP DISCOVER and REQUEST messages are constructed based on the attributes in the RADIUS request and the literal values specified in the [Request] section for the pool. A Parameter Request List option is also constructed, listing all return options required for populating the RADIUS response, as specified in the [Reply] section for the pool.

If an address is assigned by means of DHCP, the DH= field is added to the Class attribute. This field includes:

- The unique client identifier for this lease.
- The address of the DHCP server.
- The lease time.

The unique client identifier for each user session is placed in the client hardware address field of the DHCP request as well as in the Client ID option. This information is used by the DHCP server to associate IP addresses with clients.

Address Renewal

If an INTERIM accounting message whose Class attribute includes both the IP= and the DH= fields is received, a REQUEST message is unicast to the DHCP server that allocated the address in an attempt to renew the lease. It requests the same lease time as was granted for the original request. If the server is specified as a broadcast address, DHCP failover occurs if the primary DHCP server go down.

Note: If a renewal request is rejected, the DHCP server does not inform the network access device that the user's IP address is not renewed and might become invalid.

Address Release

If an accounting Stop message whose Class attribute includes both the IP= and the DH= fields is received, a RELEASE message is unicast to the DHCP server that allocated the address.

Note: The DHCP server does not reply to the RELEASE message.

An address to the DHCP server is also released when a session is deleted from its session database for reasons other than receiving an accounting Stop. For example, phantom session expiration or administrative deletion of a session result in the release of the temporary DHCP address.

DHCP Option Mapping

Options in a DHCP DISCOVER or REQUEST message can automatically be constructed based on attributes in the RADIUS request as well as pre-configured literal values. Also, options returned by the DHCP server in an OFFER message can be transmitted back to the network access device in RADIUS attributes.

The following applies to the mapping between RADIUS attributes and DHCP options:

- Both standard and vendor-specific DHCP options are supported. (Vendor-specific DHCP options must use standard encapsulation rules, as described in RFC 2132.)
- Format conversions between RADIUS attributes and DHCP options are performed. For example, a DHCP option containing an IP address is formatted into dotted notation when returned in a RADIUS string attribute.
- A single RADIUS request attribute can set more than one DHCP options in a request, and a single DHCP option can set more than one RADIUS response attribute.
- A single DHCP option containing multiple values can be mapped to multiple instances of a single RADIUS attribute.

  For example, a RADIUS attribute called IP-Router could appear multiple times in an Access-Accept. DHCP's Router option returns a list of IP addresses of routers. This single DHCP option can be configured to return multiple instances of the RADIUS IP-Router attribute -- one for each router address in the list.
- A single DHCP option containing multiple values can be mapped to multiple RADIUS attributes.

  For example, two RADIUS attributes exist, Primary-DNS-Server and Secondary-DNS-Server. DHCP's DNS Server option returns a list of IP addresses of DNS servers. This single DHCP option can be configured to set the first DNS server address in Primary-DNS-Server and the second in Secondary-DNS-Server.
- Only attributes appropriate to the dictionary are returned.

  Therefore, if network access devices from different vendors use different RADIUS attributes for the same information, each RADIUS attribute that might be required can be mapped to the same DHCP option. The correct attribute is returned to the network access device.

Using Multiple Servers

As the information required to renew or release a DHCP-assigned address is contained in the Class attribute, it is feasible to set up multiple servers, all utilizing a common DHCP server for address allocation. The network access device can issue requests to any of the servers, and addresses are assigned and released correctly even if different servers handle authentication and accounting requests for the same session.

This architecture requires that each server must be configured to be stateless—that is, the current sessions database must be turned off in the radius.ini file, as follows:

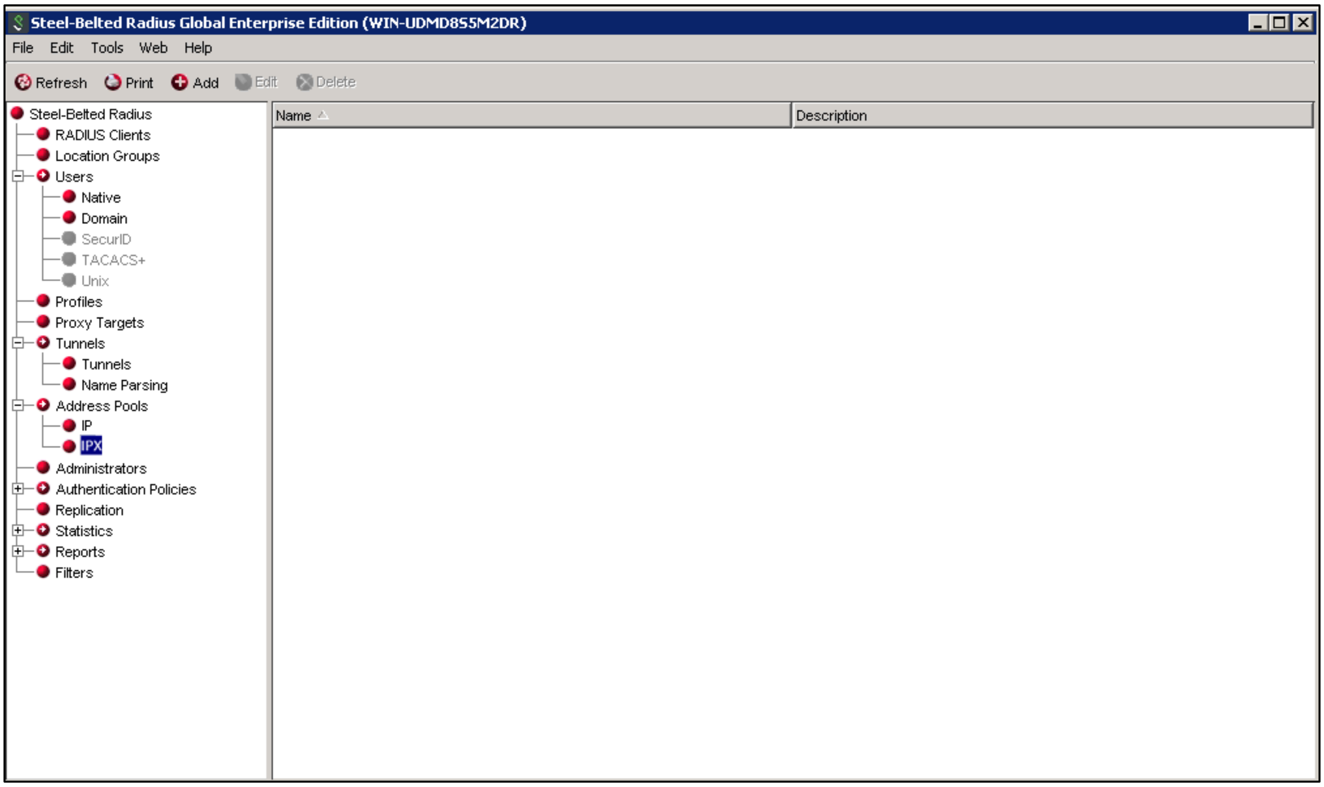
```
[CurrentSessions]

Enable = 0
```

Current sessions processing makes sense only when authentication and all accounting are directed to the same server. If current sessions processing is not disabled, the current session's database is incorrect and always growing. For example, DHCP addresses are prematurely released when phantoms expire.

Setting Up IPX Address Pools

The IPX Pools dialog (Figure 96: Address Pools Panel: IPX Pools) allows you to set up one or more pools out of which unique IPX network numbers are assigned as users require them. Each pool consists of a list of one or more ranges of IPX network numbers.

Figure 96: Address Pools Panel: IPX Pools



Adding an IPX Pool

An IPX pool consists of one or more ranges of IPX network numbers. You can add or delete ranges and set an optional description for each address pool.

To add an IPX address pool:

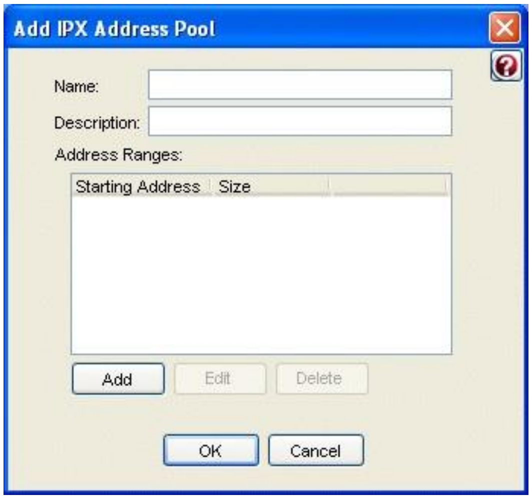
1. Choose Address Pools> IPX in the sidebar.

   The IPX Address Pools panel appears.

2. Click the Add button in the toolbar.

   The Add IPX Address Pool dialog appears.

Figure 97: Add IPX Address Pool Dialog



3. Enter the name of the IPX address pool in the **Name** field.

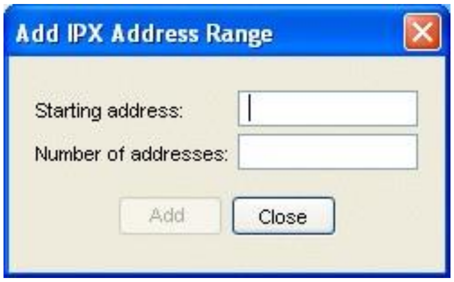
4. Optionally, enter a description of the address pool in the **Description** field.

5. Identify the address ranges in the IP address pool.

   a. Click the Add button below the Address Ranges list.

      The Add IP Address Range dialog opens.

      Figure 98: Add IP Address Range Dialog



   b. Enter the first address in the **Starting address** field.

   c. Enter the number of IPX addresses in the address range in the **Number of addresses** field.

   d. Click **Add**.

   e. Repeat steps a–d for each address range in the IPX address pool. When you are finished, click **Close.**

6. Click **OK.**

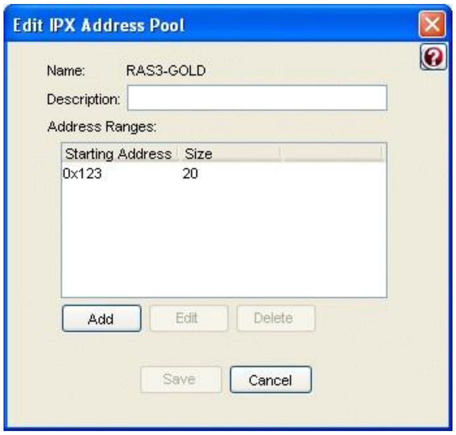
Editing an IPX Address Pool

To edit an IPX address pool:

1. Choose **Address Pools> IPX** in the sidebar.
2. Select the entry you want to modify and click the **Edit** button (or right-click the entry and choose **Edit**).

   The Edit IPX Address Pool dialog appears.

Figure 99: Edit IPX Address Pool Dialog



Name: RAS3-GOLD

Description:

Address Ranges:

Starting Address	Size
0x123	20

3. Modify the settings for the address pool as needed.
   - To add an address range to the address pool, click the **Add** button and specify the starting address and number of addresses in the range.
   - To modify an address range, select it and click the **Edit** button.
   - To delete an address ranges from the address pool, select it and click the **Delete** button.
4. When you are finished, click **Save**.

Removing an IPX Address Pool

To delete an IPX address pool:

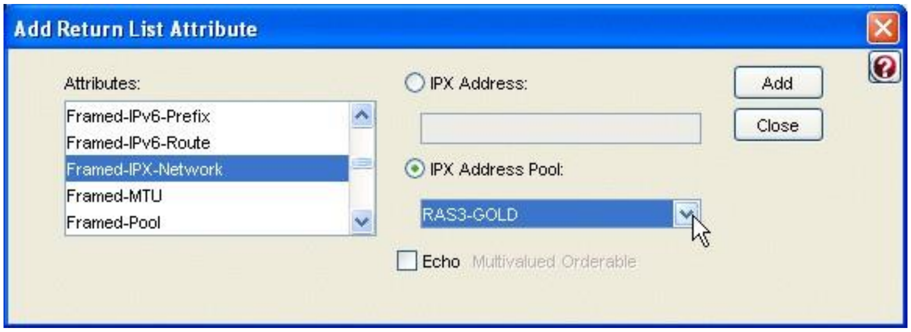
1. Click the Address Pools button to display the Address Pools panel.
2. Click the IPX Address Pools tab to display the list of IPX address pools that have been configured.
3. Select the entry you want to remove and click the Delete button (or right-click the entry and choose Delete).
4. When you are prompted to confirm the deletion, click Yes.

Specifying Pooled IPX Network Numbers in User/Profile Records

The Framed-IPX-Address return list attribute controls how Steel-Belted Radius assigns an IPX address to a user making a connection.

When you add or edit the Framed-IPX-Address attribute for a user or profile, the Framed-IPX-Address dialog appears. To select an IPX address assignment option, type an IPX address in the IPX address field or click the IPX Address Pool check box and select the name of the IPX address pool you want to use from the list.

Figure 100: Specifying an IPX Pool for the Framed-IPX-Address Attribute



Add Return List Attribute

Attributes:
- Framed-IPv6-Prefix
- Framed-IPv6-Route
- Framed-IPX-Network
- Framed-MTU
- Framed-Pool

○ IPX Address:

● IPX Address Pool: RAS3-GOLD

□ Echo Multivalued Orderable

Chapter 18
Administering Address Pools via WebGUI

This chapter describes how to set up IPv4 and IPX address pools via WebGUI. Steel-Belted Radius does not support IPv6 address pools.

Note: Please contact Pulse Secure Global Support Center If you need address pools larger than 65,535 (216) addresses.

Address Pool Files

The following files establish settings for IP and IPX address pools. For more information about these files, refer to the Steel-Belted Radius Reference Guide.

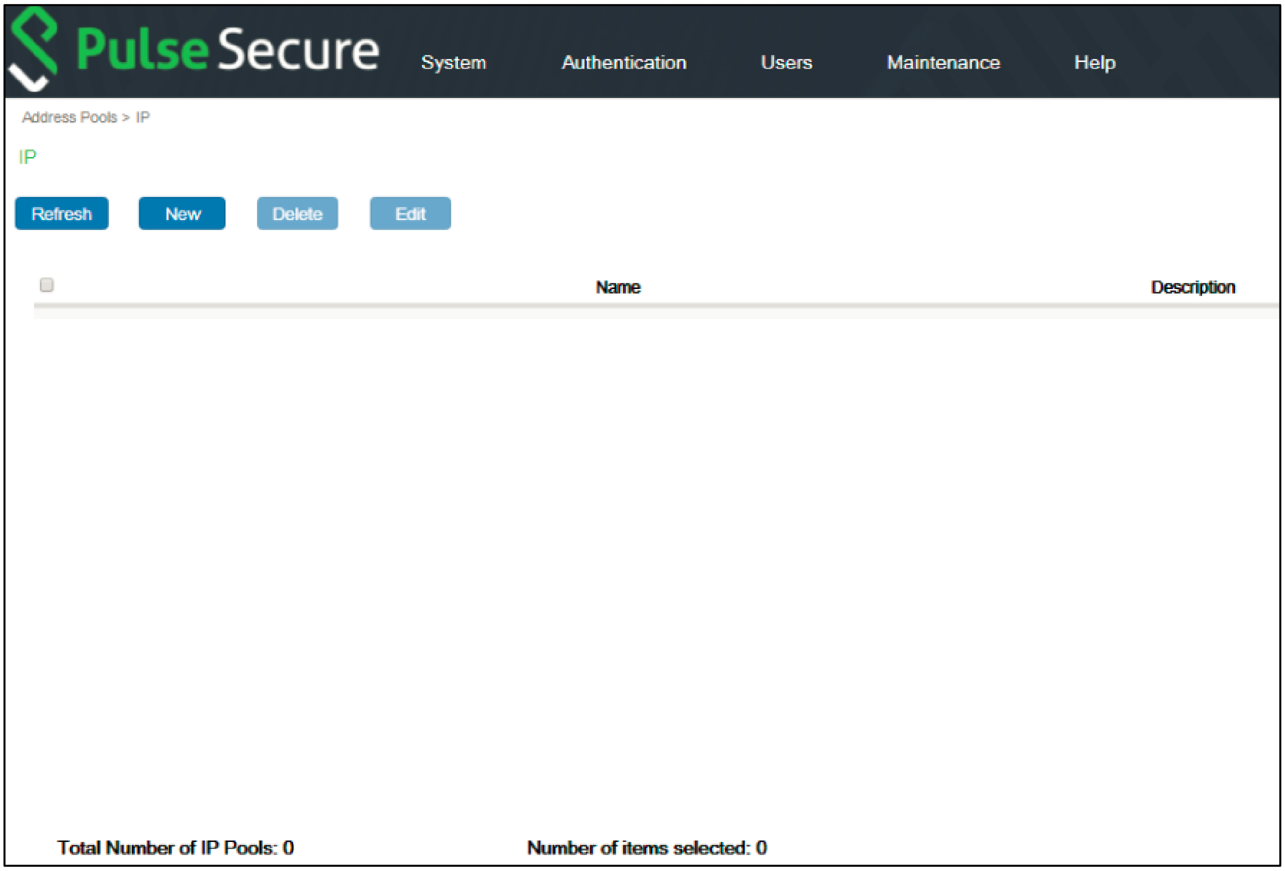
Table 24: Address Pool Files

File Name	Function
dhcp.ini	Configures DHCP address pools so that IP addresses can be assigned from a backend DHCP server.
pool.dhc	Configures specific DHCP address pools, where pool is the name of an address pool listed in dhcp.ini.
radius.ini	Specifies (among other things) the suffixes used to set up NAD-specific IP pools.

Setting Up IP Address Pools

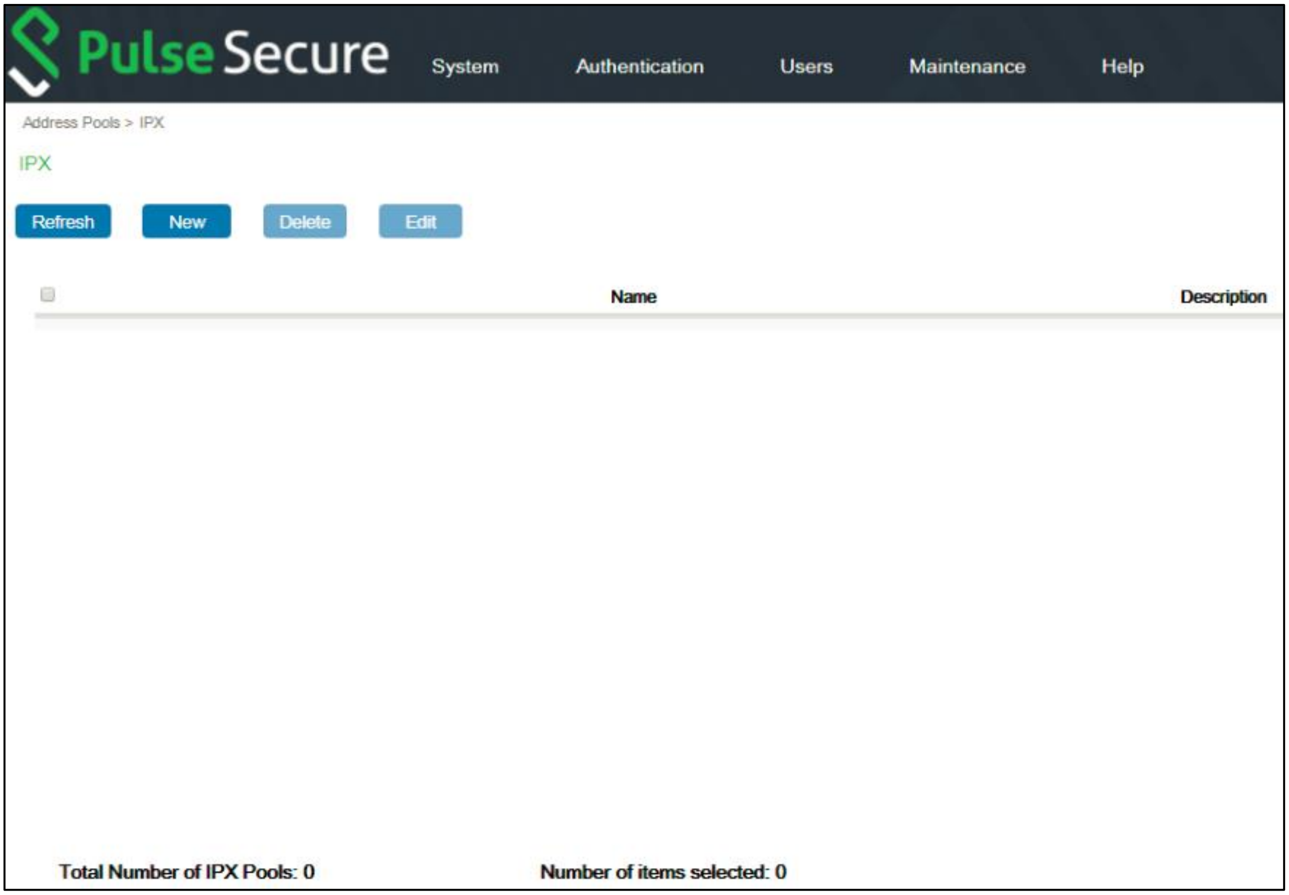
The IP Address Pool and IPX Address Pool page (Figure 101: IP Address Pools Page and IPX Address Pools Page Adding an) allow you to set up one or more pools out of which unique IPv4 or IPX addresses are assigned as users require them. Each address pool consists of a list of one or more ranges of addresses.

Figure 101: IP Address Pools Page and IPX Address Pools Page Adding an



Total Number of IP Pools: 0

Number of items selected: 0



Total Number of IPX Pools: 0

Number of items selected: 0

Adding an IPv4 Address Pool

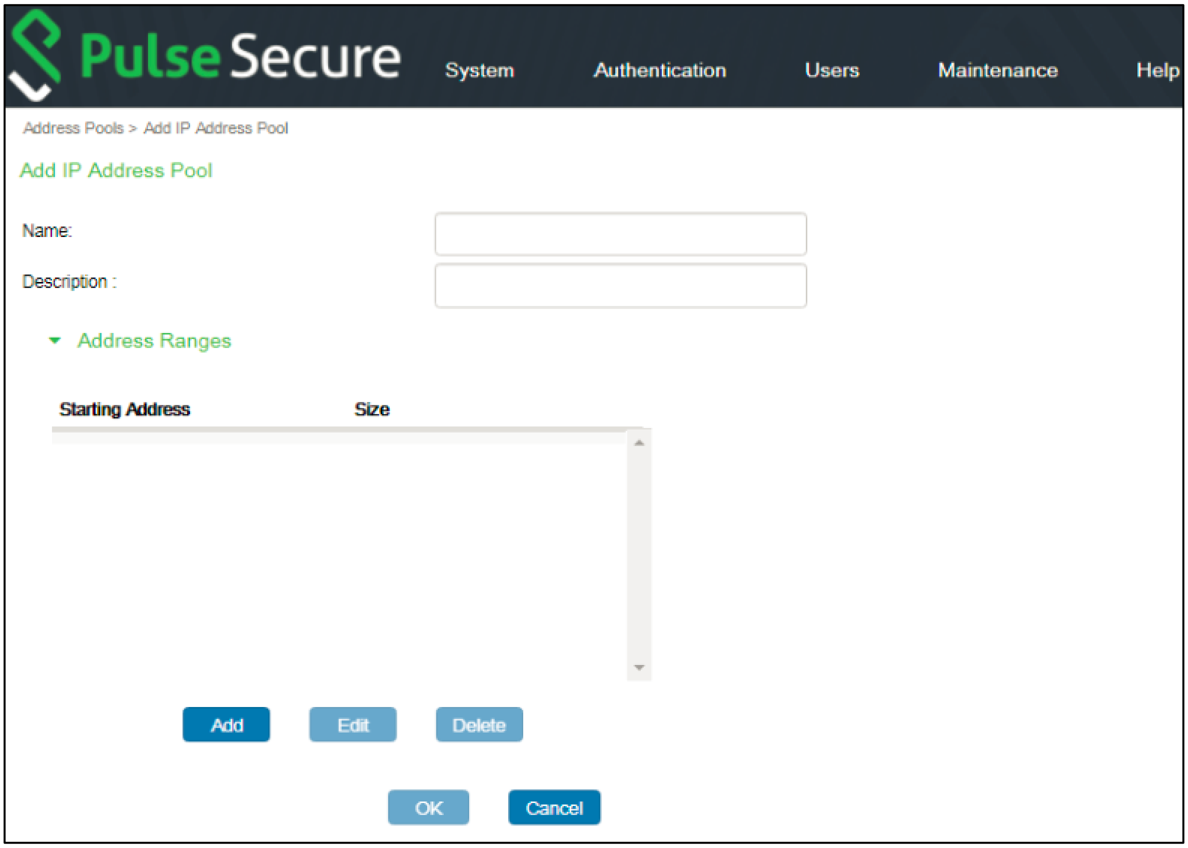
An IP address pool consists of one or more ranges of IPv4 addresses. You can add or delete ranges and set an optional description for each address pool.

To add an IP address pool:

1. Choose **System > Address Pools > IP**. The IP Address Pools page appears.
2. Click the **Add** button.

   The Add IP Address Pool page appears.

Figure 102: Add IP Address Pool Page



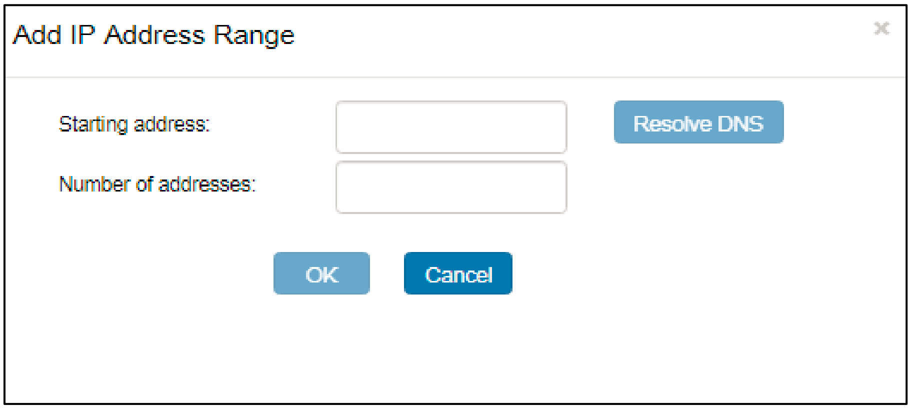
3. Enter the name of the IP address pool in the Name field.

4. Optionally, enter a description of the address pool in the Description field.

5. Identify the address range or ranges in the IP address pool.

   a. Click the Add button below the Address Ranges list.
   The Add IP Address Range page opens.

   Figure 103: Add IP Address Range Page

   b. Enter the first address in the **Starting address** field.

c. Enter the number of addresses in the address range in the **Number of addresses** field.

d. Click **Add**.

e. Repeat steps a–d for each address range in the IP address pool. When you are finished, click **Close.**

7. Click **OK.**

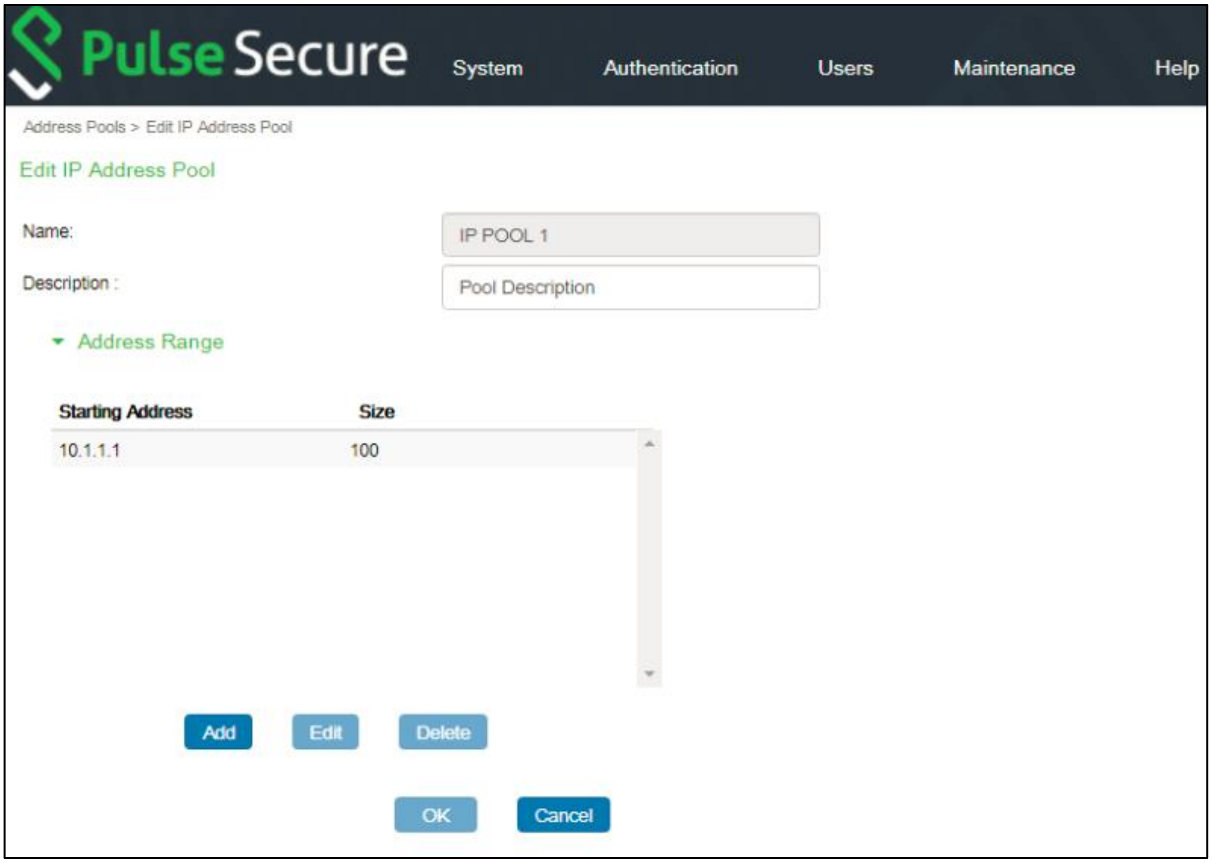
Editing an IP Address Pool

To edit an IP address pool:

1. Choose **System > Address Pools > IP.**
2. Select the entry you want to modify and click the **Edit** button.

   The Edit IP Address Pool page appears.

Figure 104: Edit IP Address Pool Page



Name: IP POOL 1

Description : Pool Description

Starting Address	Size
10.1.1.1	100

3. Modify the settings for the address pool as needed.
   - To add an address range to the address pool, click the **Add** button and specify the starting address and number of addresses in the range.
   - To modify an address range, select it and click the **Edit** button.
   - To delete an address range from the address pool, select it and click the **Delete** button.

4. When you are finished, click **Save**.

Removing an IP Address Pool

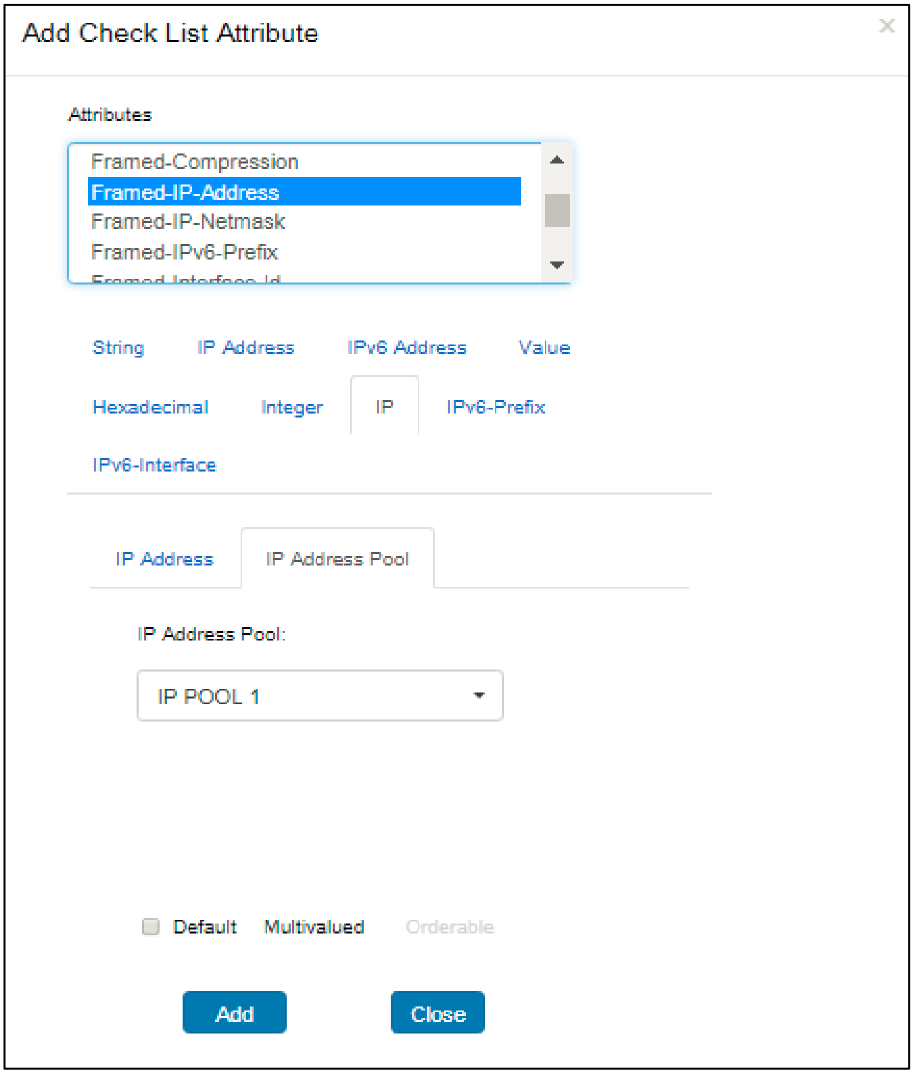
To delete an IP address pool:

1. Choose **System > Address Pools > IP.**
2. Select the entry you want to remove and click the **Delete** button.
3. When you are prompted to confirm the deletion, click Yes.

Specifying an IP Address Pool for User/Profile Records

The Framed-IP-Address return list attribute controls how the server assigns an IP address to a user making a connection. When you add or edit the Framed-IP-Address attribute in the Users or Profiles page, the Add Attribute page (Figure 105: Editing the Framed-IP-Address) allows you to choose an IP address pool instead of specifying an IP address.

Figure 105: Editing the Framed-IP-Address



Service-Level IP Address Pools

Steel-Belted Radius allows you to define a set of suffixes that define categories of IP address pools. For example, a pool category might correspond to the kinds of services available to users in that category. You might decide to define categories called Bronze, Silver, and Gold to identify different packet routing priorities.

To create a set of service-level address pools:

1. Define suffixes for the various service-level address pools in the [IPPoolSuffixes] section of radius.ini. For example:

   [IPPoolSuffixes]

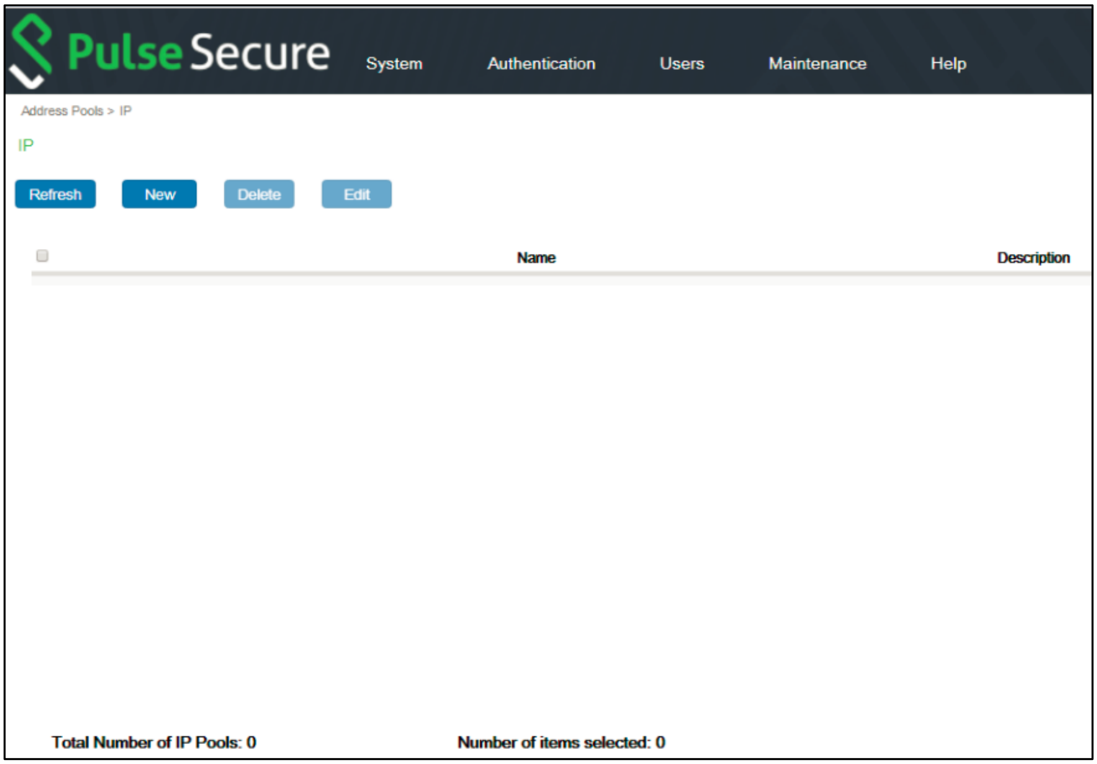
   -GOLD

   -SILVER

   -BRONZE

2. Define IP address pools using the suffixes configured in the [IPPoolSuffixes] section of radius.ini

Figure 106: Service Level Suffixes in the IP Address Pools Page



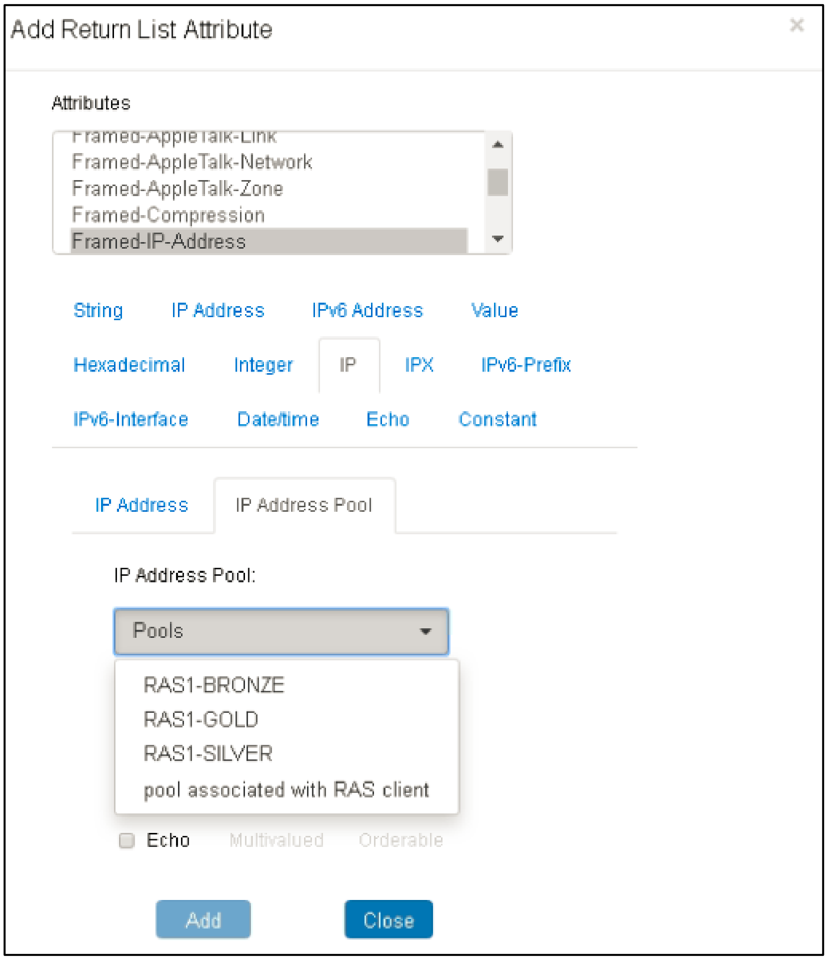
Total Number of IP Pools: 0

Number of items selected: 0

3. Associate the new IP address pool with the appropriate NAD by use of **IP Address** Pool field on the RADIUS Clients page.

4. Assign a user to a NAD-specific IP address pool and suffix in the Users page or the Profiles page.

Figure 107: Associating IP Address Pools with RADIUS Clients



If user EDISON CARTER, who has been assigned to <RAS>-GOLD, logs into RAS1, he receives an IP address from the RAS1-GOLD address pool. If he logs into RAS2, he receives an address from the RAS2-GOLD address pool. If, however, he logs into RAS3 but RAS3-GOLD has not been defined in the IP Pools page, he is not assigned an IP address.

Specifying IP Address Assignment from a DHCP Server

IP addresses can be assigned from a backend DHCP server, rather than from a standard IP address pool. DHCP address pools function like internal address pools—Framed-IP-Address can be allocated from any address pool, either internal or DHCP.

DHCP address pools are defined in the dhcp.ini file and initialization files with the extension .dhc.

In addition, each DHCP address pool must be enabled by adding a placeholder IP address pool in the SBR Administrator. This placeholder pool should have the same name as the DHCP pool, and should have an empty list of address ranges. The placeholder pool allows the DHCP pool to appear in lists presented by the SBR Administrator, so it can be selected into an attribute.

When an IP address must be assigned from a DHCP pool during an Access-Request, DHCP DISCOVER and REQUEST messages are issued to trigger the allocation of an address. When an accounting Stop ends the session, DHCP RELEASE is issued to the server that allocated the address. Upon receipt of an accounting INTERIM request, a DHCP REQUEST message is issued to the server that allocated the address, attempting to extend the lease. If the server is specified as a broadcast address, DHCP failover occurs if the primary DHCP server go down.

DHCP leases can be acquired, extended, and released by different servers. The server that acquires the lease adds all the information for extending and releasing the lease to the Class attribute.

Flexible configuration features allow RADIUS attributes to be mapped to DHCP options. Therefore, information from a RADIUS request can be provided to the DHCP server, and information returned from the DHCP server can be returned to the network access device.

During authentication, if an address is assigned from a pool, the pool name must refer to either a DHCP pool or an internal pool. If the pool name is not found, the request is rejected.

Address Allocation

During address allocation, a DISCOVER message is issued. If an OFFER is received from a DHCP server and the offered lease time meets the minimum lease time requirements, the server issues a REQUEST message. If an ACK message is received, the allocated address is returned in the Access-Accept.

In addition to the options required for normal DHCP operation, additional options in the DHCP DISCOVER and REQUEST messages are constructed based on the attributes in the RADIUS request and the literal values specified in the [Request] section for the pool. A Parameter Request List option is also constructed, listing all return options required for populating the RADIUS response, as specified in the [Reply] section for the pool.

If an address is assigned by means of DHCP, the DH= field is added to the Class attribute. This field includes:

- The unique client identifier for this lease.
- The address of the DHCP server.
- The lease time.

The unique client identifier for each user session is placed in the client hardware address field of the DHCP request as well as in the Client ID option. This information is used by the DHCP server to associate IP addresses with clients.

Address Renewal

If an INTERIM accounting message whose Class attribute includes both the IP= and the DH= fields is received, a REQUEST message is unicast to the DHCP server that allocated the address in an attempt to renew the lease. It requests the same lease time as was granted for the original request. If the server is specified as a broadcast address, DHCP failover occurs if the primary DHCP server go down.

Note: If a renewal request is rejected, the DHCP server does not inform the network access device that the user’s IP address is not renewed and might become invalid.

Address Release

If an accounting Stop message whose Class attribute includes both the IP= and the DH= fields is received, a RELEASE message is unicast to the DHCP server that allocated the address.

Note: The DHCP server does not reply to the RELEASE message.

An address to the DHCP server is also released when a session is deleted from its session database for reasons other than receiving an accounting Stop. For example, phantom session expiration or administrative deletion of a session result in the release of the temporary DHCP address.

DHCP Option Mapping

Options in a DHCP DISCOVER or REQUEST message can automatically be constructed based on attributes in the RADIUS request as well as pre-configured literal values. Also, options returned by the DHCP server in an OFFER message can be transmitted back to the network access device in RADIUS attributes.

The following applies to the mapping between RADIUS attributes and DHCP options:

- Both standard and vendor-specific DHCP options are supported. (Vendor-specific DHCP options must use standard encapsulation rules, as described in RFC 2132.)
- Format conversions between RADIUS attributes and DHCP options are performed. For example, a DHCP option containing an IP address is formatted into dotted notation when returned in a RADIUS string attribute.
- A single RADIUS request attribute can set more than one DHCP options in a request, and a single DHCP option can set more than one RADIUS response attribute.
- A single DHCP option containing multiple values can be mapped to multiple instances of a single RADIUS attribute.

  For example, a RADIUS attribute called IP-Router could appear multiple times in an Access-Accept. DHCP's Router option returns a list of IP addresses of routers. This single DHCP option can be configured to return multiple instances of the RADIUS IP-Router attribute -- one for each router address in the list.
- A single DHCP option containing multiple values can be mapped to multiple RADIUS attributes.

  For example, two RADIUS attributes exist, Primary-DNS-Server and Secondary-DNS-Server. DHCP's DNS Server option returns a list of IP addresses of DNS servers. This single DHCP option can be configured to set the first DNS server address in Primary-DNS-Server and the second in Secondary-DNS-Server.
- Only attributes appropriate to the dictionary are returned.

  Therefore, if network access devices from different vendors use different RADIUS attributes for the same information, each RADIUS attribute that might be required can be mapped to the same DHCP option. The correct attribute is returned to the network access device.

Using Multiple Servers

As the information required to renew or release a DHCP-assigned address is contained in the Class attribute, it is feasible to set up multiple servers, all utilizing a common DHCP server for address allocation. The network access device can issue requests to any of the servers, and addresses are assigned and released correctly even if different servers handle authentication and accounting requests for the same session.

This architecture requires that each server must be configured to be stateless—that is, the current sessions database must be turned off in the radius.ini file, as follows:

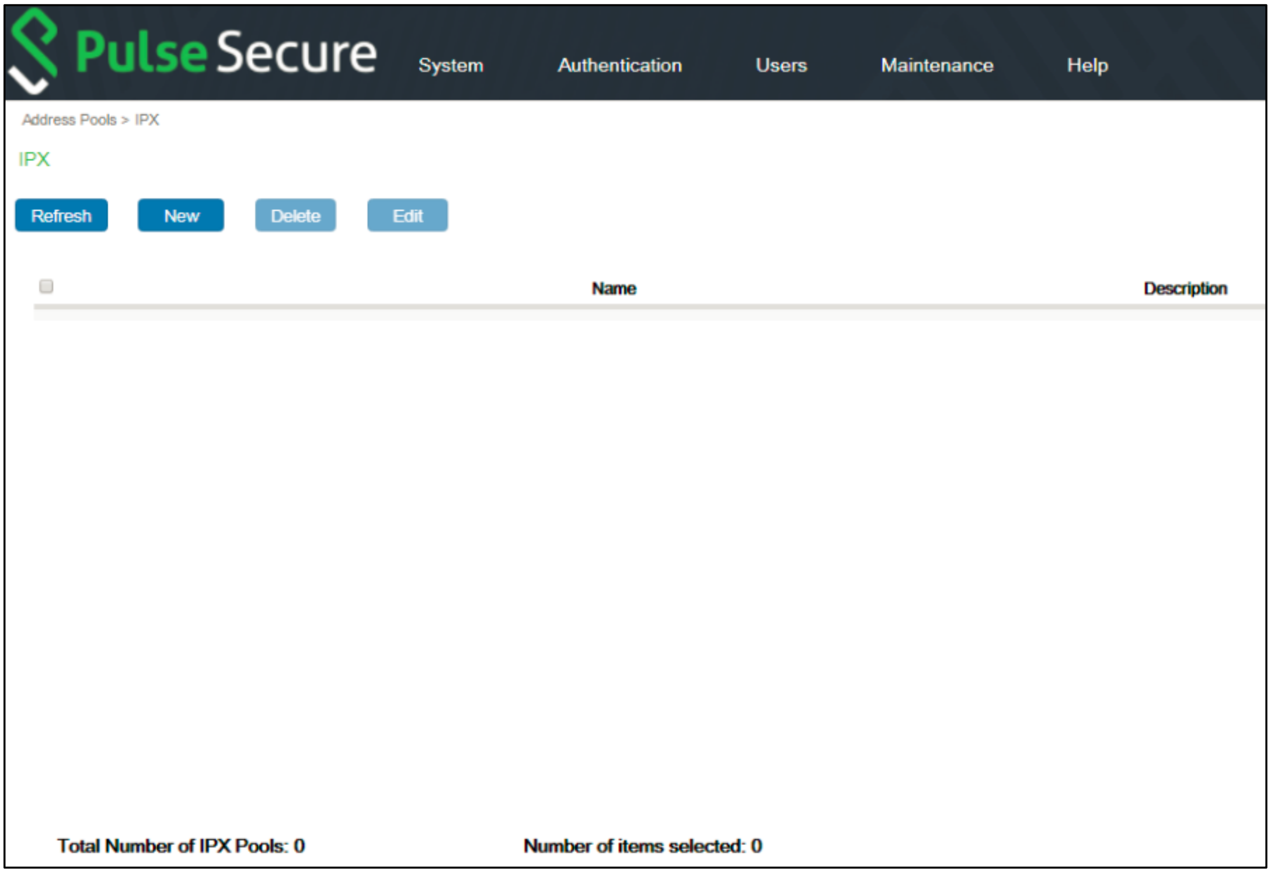
```
[CurrentSessions]

Enable = 0
```

Current sessions processing makes sense only when authentication and all accounting are directed to the same server. If current sessions processing is not disabled, the current session's database is incorrect and always growing. For example, DHCP addresses are prematurely released when phantoms expire.

Setting Up IPX Address Pools

The IPX Pools page (Figure 108: Address Pools Page: IPX Pools) allows you to set up one or more pools out of which unique IPX network numbers are assigned as users require them. Each pool consists of a list of one or more ranges of IPX network numbers.

Figure 108: Address Pools Page: IPX Pools



Adding an IPX Pool

An IPX pool consists of one or more ranges of IPX network numbers. You can add or delete ranges and set an optional description for each address pool.

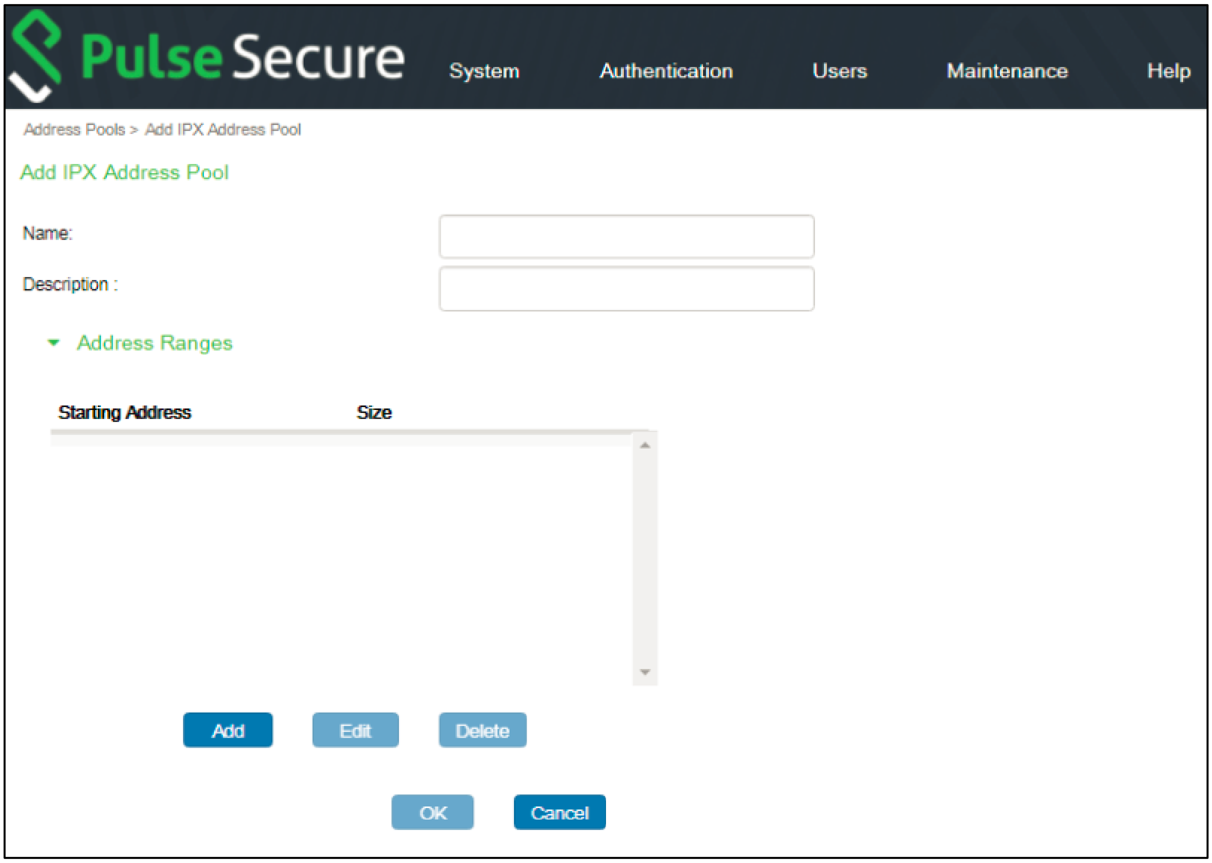
To add an IPX address pool:

1. Choose System > Address Pools > IPX .
   The IPX Address Pools page appears.

2. Click the New button.

   The Add IPX Address Pool page appears.

Figure 109: Add IPX Address Pool Page



3. Enter the name of the IPX address pool in the **Name** field.

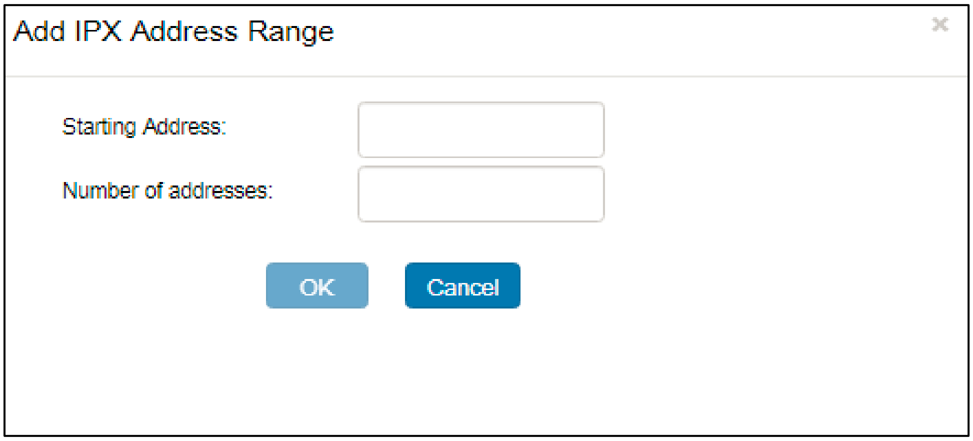
4. Optionally, enter a description of the address pool in the **Description** field.

5. Identify the address ranges in the IP address pool.

   a. Click the Add button below the Address Ranges list.

      The Add IP Address Range page opens.

Figure 110: Add IP Address Range Page



b. Enter the first address in the **Starting address** field.

c. Enter the number of IPX addresses in the address range in the **Number of addresses** field.

d. Click **Add**.

e. Repeat steps a–d for each address range in the IPX address pool. When you are finished, click **Close.**

6. Click **OK.**

Editing an IPX Address Pool

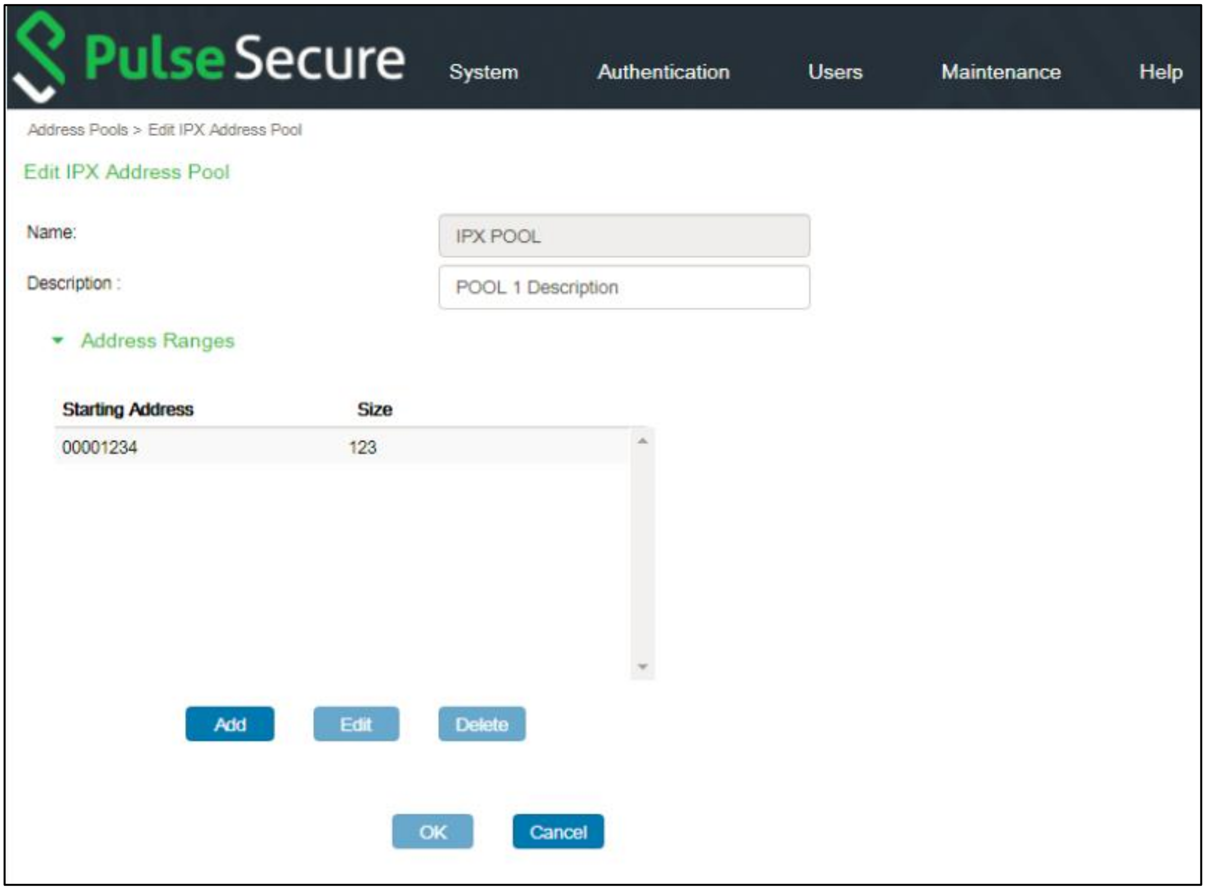
To edit an IPX address pool:

1. Choose **System > Address Pools > IPX**.
2. Select the entry you want to modify and click the **Edit** button

   .

   The Edit IPX Address Pool page appears.

Figure 111: Edit IPX Address Pool Page



Edit IPX Address Pool

Name: IPX POOL

Description : POOL 1 Description

Address Ranges

Starting Address	Size
00001234	123

3. Modify the settings for the address pool as needed.

    - To add an address range to the address pool, click the **Add** button and specify the starting address and number of addresses in the range.
    - To modify an address range, select it and click the **Edit** button.
    - To delete an address ranges from the address pool, select it and click the **Delete** button.

4. When you are finished, click **Save**.

Removing an IPX Address Pool

To delete an IPX address pool:

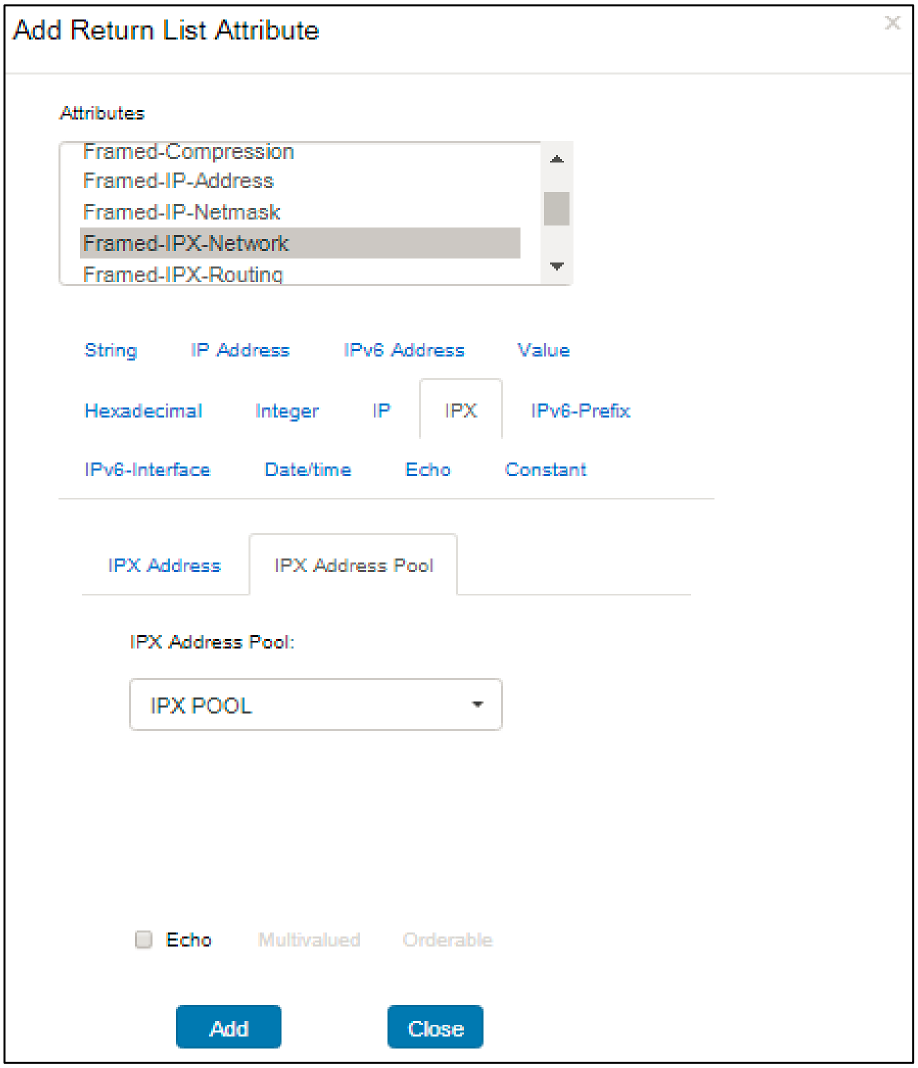
1. Choose **System > Address Pools > IPX** to display the list of IPX address pools that have been configured.
2. Select the entry you want to remove and click the Delete button.
3. When you are prompted to confirm the deletion, click Yes.

Specifying Pooled IPX Network Numbers in User/Profile Records

The Framed-IPX-Address return list attribute controls how Steel-Belted Radius assigns an IPX address to a user making a connection.

When you add or edit the Framed-IPX-Address attribute for a user or profile, the Framed-IPX-Address page appears. To select an IPX address assignment option, type an IPX address in the IPX address field or click the IPX Address Pool check box and select the name of the IPX address pool you want to use from the list.

Figure 112: Specifying an IPX Pool for the Framed-IPX-Address Attribute


Add Return List Attribute

Attributes

- Framed-Compression
- Framed-IP-Address
- Framed-IP-Netmask
- Framed-IPX-Network
- Framed-IPX-Routing

IPX Address Pool:

IPX POOL

□ Echo

Chapter 19

Setting Up Administrator Accounts via Legacy SBR Administrator

This chapter describes how to set up Steel-Belted Radius administrators via legacy SBR administrator and specify what permissions an administrator holds.

Administrator Files

The following files establish settings for administrative permissions. For more information about these files, refer to the Steel-Belted Radius Reference Guide.

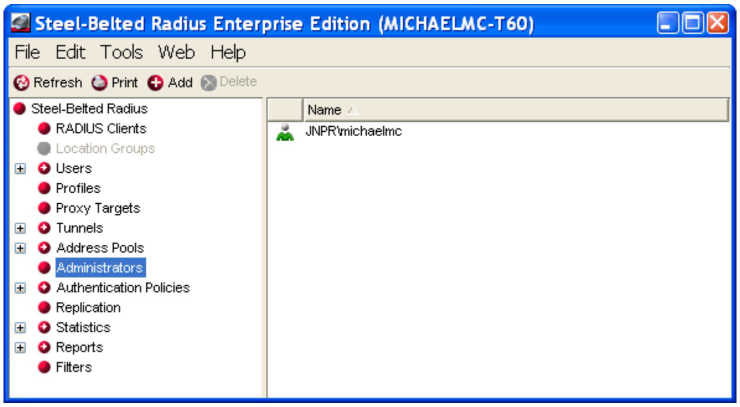
Table 25: Administrator Files

File Name	Function
access.ini	Maps operating system user or group account levels to administrative permissions.
admin.ini	Maps administrative access levels to sets of access rights.

Administrators Panel

The Administrators panel lets you grant and revoke the right to use the SBR Administrator to configure a Steel-Belted Radius server. Each time you log into a Steel-Belted Radius server, SBR Administrator prompts you to authenticate yourself by entering an account name and password.

Figure 113: Administrators Panel



When the Steel-Belted Radius software is installed, any user who is a member of the group Administrators on a Steel-Belted Radius server implicitly has the right to use the SBR Administrator at its default (full) level of access. The Administrators panel lets you modify these default permissions.

The Administrators panel lists the users and groups who have been explicitly granted the right to run the SBR Administrator. Local users or groups are shown with their normal name. Remote users or groups are shown with the name of the domain, followed by a backslash and then the name of the domain user or group.

If you want to control administrative access at a finer level of detail, Steel-Belted Radius allows you to designate other Windows user and/or group accounts as administrative accounts. You can also assign various levels of administrative privilege to these accounts. Refer to “access.ini File” and “admin.ini File” in the Steel-Belted Radius Reference Guide for more information.

Adding a Local Administrator

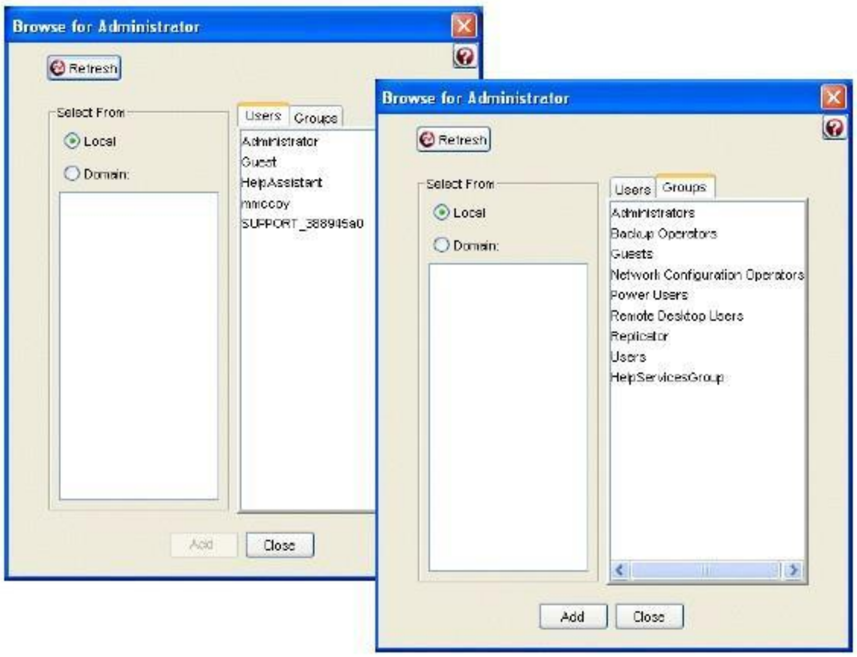
To add a local administrator to the Steel-Belted Radius database:

1. Choose **Administrators** in the sidebar.

   The Administrators panel appears.

2. Click the Add button to open the Browse for Administrator dialog.

Figure 114: Browse for Administrator Dialogs



3. Click the **Local** radio button to specify you are adding a local user.

4. Identify the local users or groups you want to add.

   - If you want to add a local user, click the **Users** tab and select the name of a user.

   - If you want to add a local group, click the **Groups** tab and select the name of a user group.

5. Click **Add.**

6. Continue adding local users and groups until you are done, then click **Close.**

Adding a Remote Administrator

Note: Browsing within a domain with large number of users or groups to select an administrator or group name can adversely affect Steel-Belted Radius performance. See “Adding a Remote Administrator Manually” for information on how to add a remote administrator manually.

To grant access to a remote administrator within a domain:

1. Choose Administrators in the sidebar.

   The Administrators panel appears.

2. Click the **Add** button to open the Browse for Administrator dialog Figure 114: Browse for Administrator Dialogs).

3. Click the **Domain** radio button to specify you are adding a remote (domain) user.

4. When the list of domains appears, select the domain within which you would like to grant access.

5. Identify the remote users or groups within that domain who you want to add.

   - If you want to add a remote user, click the **Users** tab and select the name of a user.

   - If you want to add a remote group, click the **Groups** tab and select the name of a user group.

6. Click **Add.**

7. Continue adding domain users and groups until you are done, then click **Close.**

Adding a Remote Administrator Manually

You can add a remote administrator manually by editing your admin.ini and access.ini configuration files. The access.ini file maps user or group account names to levels of administrative privilege. The admin.ini file maps administrative access levels to sets of access rights.

To add an administrator manually, create an entry in the [Users] section of the access.ini file in domain\user = access format. For example:

```
[Users]

_system.localhost=SnmpAgent

MyDomain\myuser=superadmin
```

If you want to add a group, create an entry in domain\group = access format in the [Groups] section of the access.ini file.

```
[Groups]

SBRDomain\Administrators=superadmin
```

After you modify and save the access.ini file, restart Steel-Belted Radius to make the new entries active.

If you want to specify which settings an administrator can view or change, you can create a new access level in the admin.ini file, and then associate that access level with one or more administrators. For example, the UserSetup section in admin.ini creates an access level called Users that would let an administrator add/modify/delete user entries and display existing profiles and IP address pools, Access to settings not referenced in this list is automatically denied to a user given this access level.

```
[UserSetup]

Users=rw

Profiles=r
```

```
IP-Pools=r
```

To assign the UserSetup access level to an administrator, edit the access.ini file, associate the Users access level with one or more users, and restart Steel-Belted Radius.

```
[Users]

_system.localhost=SnmpAgent

MyDomain\myuser=UserSetup
```

Deleting an Administrator

To revoke rights for a Steel-Belted Radius administrator:

1. Choose **Administrators** in the sidebar.

   The Administrators panel appears.

2. Select the user or group whose administration rights you want to revoke.
3. Click the **Delete** button on the SBR Administrator toolbar (or right-click the entry and choose **Delete**).
4. When you are prompted to confirm the deletion, click **Yes.**

ⓘ Note: Be careful not to revoke your own rights. If you do so, you will no longer have access to Steel-Belted Radius administrative functions.

Chapter 20
Setting Up Administrator Accounts via WebGUI

This chapter describes how to set up Steel-Belted Radius administrators and specify what permissions an administrator holds via WebGUI.

Administrator Files

The following files establish settings for administrative permissions. For more information about these files, refer to the Steel-Belted Radius Reference Guide.

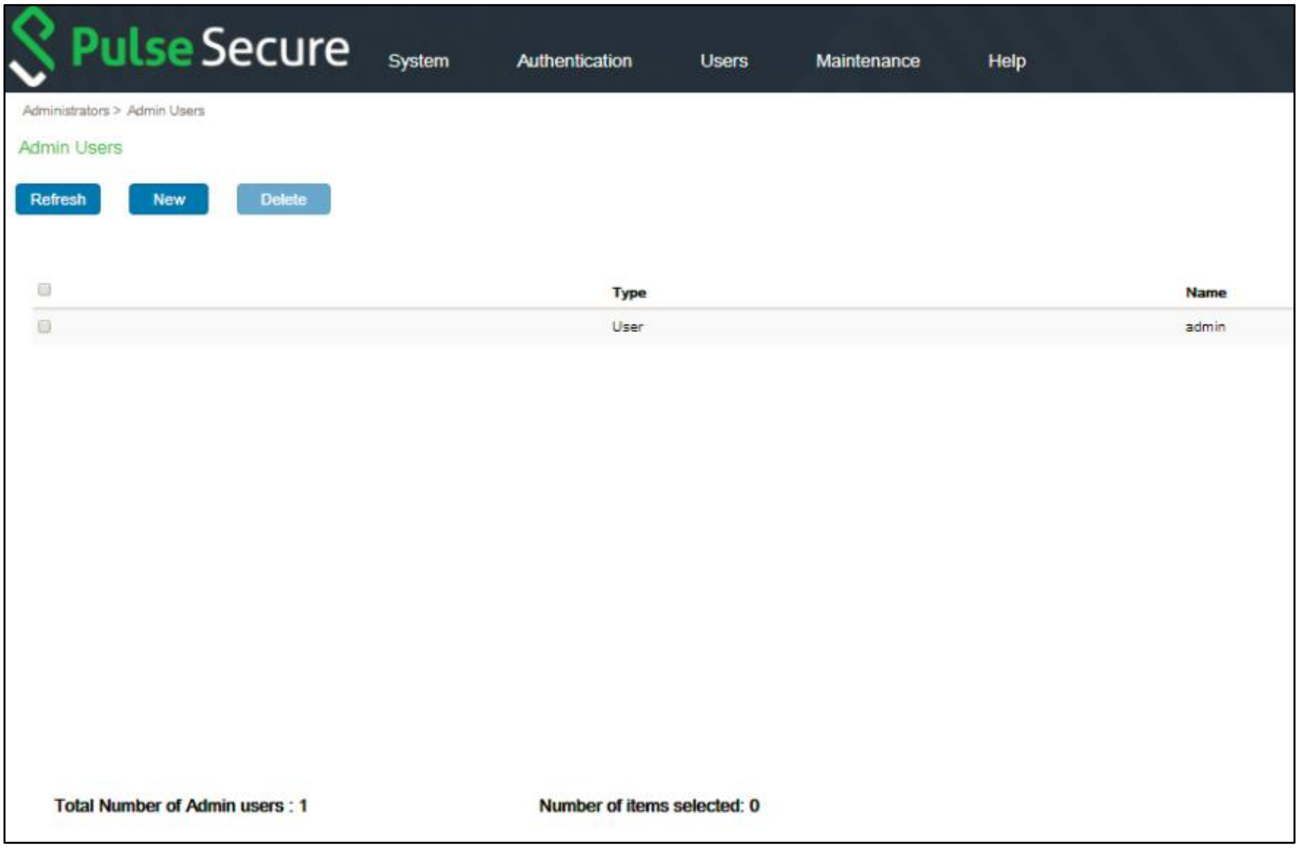
Table 25: Administrator Files

File Name	Function
access.ini	Maps operating system user or group account levels to administrative permissions.
admin.ini	Maps administrative access levels to sets of access rights.

Administrators Page

The Administrators page lets you grant and revoke the right to use the SBR Administrator to configure a Steel-Belted Radius server. Each time you log into a Steel-Belted Radius server, SBR Administrator prompts you to authenticate yourself by entering an account name and password.

Figure 115: Administrators Page



□	Type	Name
□	User	admin

Total Number of Admin users : 1

Number of items selected: 0

When the Steel-Belted Radius software is installed, any user who is a member of the group Administrators on a Steel-Belted Radius server implicitly has the right to use the SBR Administrator at its default (full) level of access. The Administrators page lets you modify these default permissions.

The Administrators page lists the users and groups who have been explicitly granted the right to run the SBR Administrator. Local users or groups are shown with their normal name. Remote users or groups are shown with the name of the domain, followed by a backslash and then the name of the domain user or group.

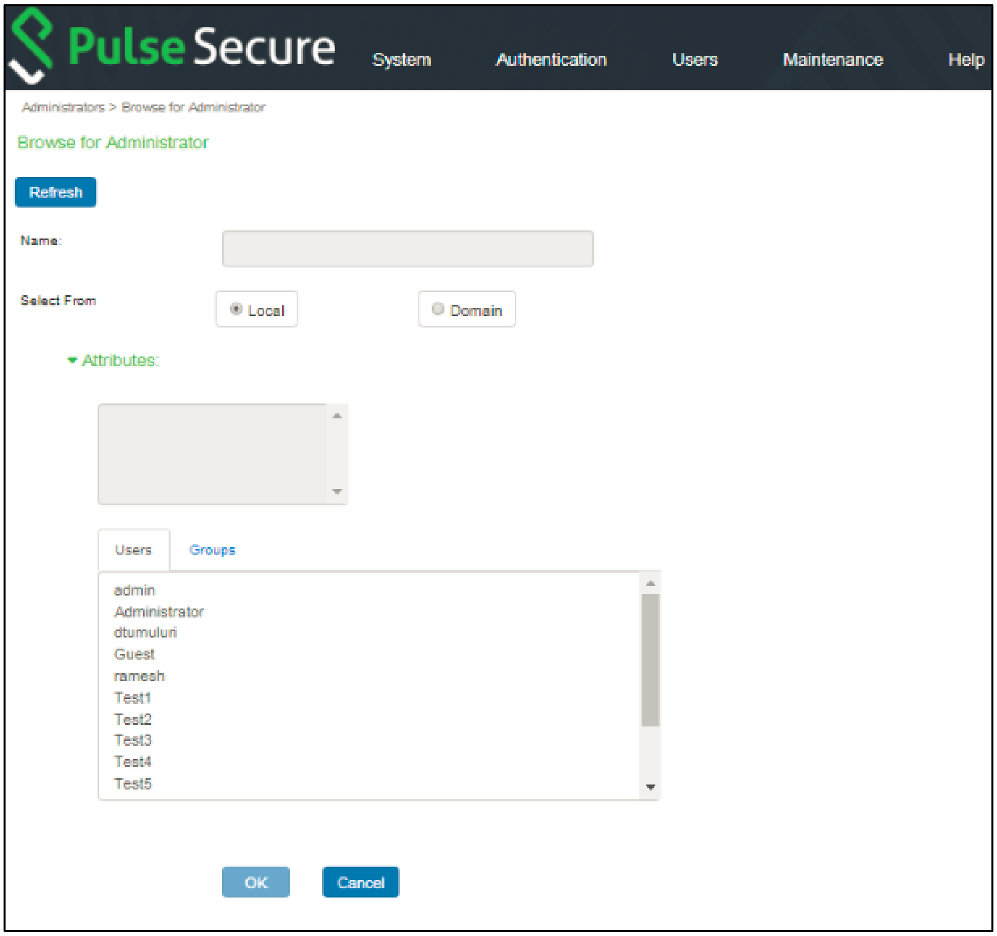
If you want to control administrative access at a finer level of detail, Steel-Belted Radius allows you to designate other Windows user and/or group accounts as administrative accounts. You can also assign various levels of administrative privilege to these accounts. Refer to “access.ini File” and “admin.ini File” in the Steel-Belted Radius Reference Guide for more information.

Adding a Local Administrator

To add a local administrator to the Steel-Belted Radius database:

1. Choose **Users > Administrator > Admin Users**. The Administrators page appears.
2. Click the New button to open the Browse for Administrator page.

Figure 116: Browse for Administrator Pages



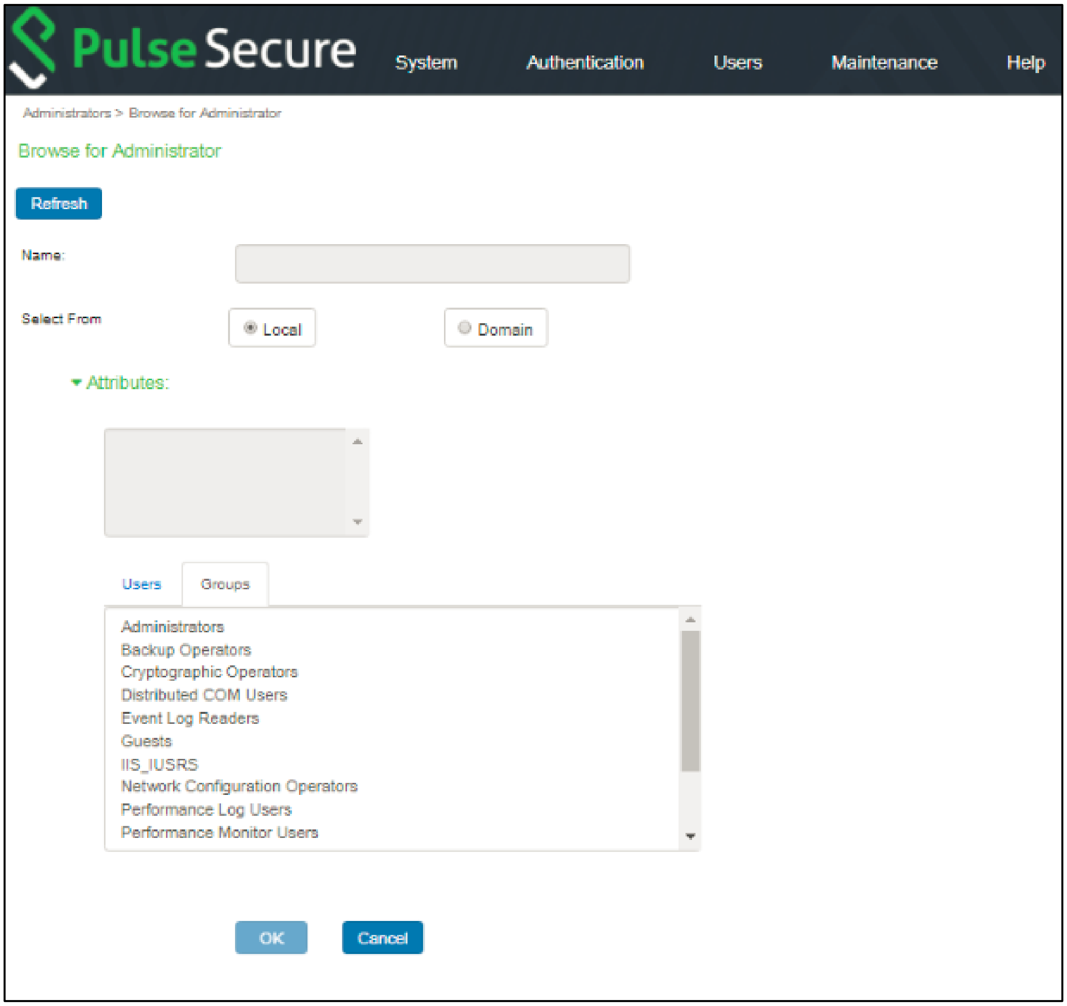
Administrators > Browse for Administrator

Browse for Administrator

Name:

Select From ● Local ○ Domain

admin
Administrator
dtumuluri
Guest
ramesh
Test1
Test2
Test3
Test4
Test5



3. Click the **Local** radio button to specify you are adding a local user.
4. Identify the local users or groups you want to add.
   - If you want to add a local user, click the **Users** tab and select the name of a user.
   - If you want to add a local group, click the **Groups** tab and select the name of a user group.
5. Click **Add.**
6. Continue adding local users and groups until you are done, then click **Close.**

Adding a Remote Administrator

Note: Browsing within a domain with large number of users or groups to select an administrator or group name can adversely affect Steel-Belted Radius performance. See "Adding a Remote Administrator Manually" for information on how to add a remote administrator manually.

To grant access to a remote administrator within a domain:

1. Choose Users > Administrators > Admin Users.

   The Administrators page appears.

2. Click the New button to open the Browse for Administrator page (Figure 116: Browse for Administrator Pages.
3. Click the **Domain** radio button to specify you are adding a remote (domain) user.
4. When the list of domains appears, select the domain within which you would like to grant access.
5. Identify the remote users or groups within that domain who you want to add.
   - If you want to add a remote user, click the **Users** tab and select the name of a user.
   - If you want to add a remote group, click the **Groups** tab and select the name of a user group.
6. Click **Add.**
7. Continue adding domain users and groups until you are done, then click **Close.**

Adding a Remote Administrator Manually

You can add a remote administrator manually by editing your admin.ini and access.ini configuration files. The access.ini file maps user or group account names to levels of administrative privilege. The admin.ini file maps administrative access levels to sets of access rights.

To add an administrator manually, create an entry in the [Users] section of the access.ini file in domain\user = access format. For example:

```
[Users]

_system.localhost=SnmpAgent

MyDomain\myuser=superadmin
```

If you want to add a group, create an entry in domain\group = access format in the [Groups] section of the access.ini file.

```
[Groups]

SBRDomain\Administrators=superadmin
```

After you modify and save the access.ini file, restart Steel-Belted Radius to make the new entries active.

If you want to specify which settings an administrator can view or change, you can create a new access level in the admin.ini file, and then associate that access level with one or more administrators. For example, the UserSetup section in admin.ini creates an access level called Users that would let an administrator add/modify/delete user entries and display existing profiles and IP address pools, Access to settings not referenced in this list is automatically denied to a user given this access level.

```
[UserSetup]

Users=rw

Profiles=r

IP-Pools=r
```

To assign the UserSetup access level to an administrator, edit the access.ini file, associate the Users access level with one or more users, and restart Steel-Belted Radius.

```
[Users]
_system.localhost=SnmpAgent
MyDomain\myuser=UserSetup
```

Deleting an Administrator

To revoke rights for a Steel-Belted Radius administrator:

1. Choose **Users > Administrators > Admin Users.**

   The Administrators page appears.

2. Select the user or group whose administration rights you want to revoke.
3. Click the **Delete** button.
4. When you are prompted to confirm the deletion, click **Yes.**

Note: Be careful not to revoke your own rights. If you do so, you will no longer have access to Steel-Belted Radius administrative functions.

Chapter 21
Configuring Realm Support

This chapter describes how to configure proxy and directed realms in Steel-Belted Radius.

Realm Configuration Files

The following files establish settings for setting up users. For more information about these files, refer to the Steel-Belted Radius Reference Guide.

Table 26: Realm Files

File Name	Function
filter.ini	Stores filters for RADIUS attributes. Filters might be referenced from the [Auth] or [Acct] section of a RealmName.pro or RealmName.dir file. Note: You use the SBR Administrator to configure the filter.ini file. Donot edit the filter.ini file manually.
proxy.ini	Specifies the order of realm selection methods, the realm selection rules, and other settings for all realms on the server.
proxyrl.ini	Configures settings for smart static accounting, which lets you specify that accounting packets for a proxy or directed realm should be forwarded to one or more proxy realms.
radius.ini	Enables proxy support and specifies (among other things) the realm names that indicate a server should handle a request and whether realm names and other username decorations should be stripped.
RealmName.dir	Specifies settings for each directed or accounting realm you set up, where RealmName is the realm name. Note that RealmName must be listed in the [Directed] section of the proxy.ini file.
RealmName.pro	Specifies settings for each proxy realm you set up, where RealmName is the realm name. Note that RealmName must be listed in the [Realms] section of the proxy.ini file.

Stage One of Realm Configuration

Perform the following steps to configure a realm of any type for Steel-Belted Radius. Some steps require that you edit parameters in Steel-Belted Radius configuration files.

1. Determine whether you want to provide proxy RADIUS or directed realms.

   Does the customer have its own RADIUS server(s), to which you'll direct requests? If so, you'll need to set up a proxy RADIUS realm for the customer.

   Does the customer need you to host its RADIUS server? If so, you'll need to set up a directed authentication and/or accounting realm for the customer.

2. If you have not done so already, configure Steel-Belted Radius to support realms. You must do this

for either type of realm.

```
radius.ini

[Configuration]

ExtendedProxy=1
```

3. You might also enable the attribute filtering feature for proxy RADIUS realms.

```
AttributeEdit=1
```

4. If you have not done so already, define delimiter conventions for realm name parsing.

   The delimiter conventions that you define in proxy.ini are used for all realms defined in Steel-Belted Radius. Be sure to inform the customer of the delimiter and prefix/suffix conventions you are using for realms.

```
proxy.ini

[Configuration]

RealmSuffix=

RealmPrefix=
```

5. Agree upon a realm name (or DNIS grouping) with the customer.

   Keep the realm name simple if the realm is not defined by DNIS, because end users must enter it in combination with their existing usernames (for example, User@RealmName). The realm name configured in Steel-Belted Radius does not need to match any names in use at the customer site. The realm name cannot match an existing target name, realm name, or tunnel name in your Steel-Belted Radius configuration.

   If the realm is defined as a DNIS grouping, the user is matched to a realm based on the Called-Station-Id.

Configuring a Proxy RADIUS Realm

A proxy RADIUS server treats a realm as a destination against which it performs authentication and accounting.

Table 27 outlines the process of configuring a new proxy RADIUS realm for Steel-Belted Radius. **Table 27** also lists the sections that you must edit in configuration files to accomplish each step. No step in this process might be omitted unless this table indicates that it is optional.

Table 27: Proxy Realm Configuration

Step	Proxy RADIUS Configuration Task	File and Section
1	Complete the preparatory steps outlined in the previous section.	—
2	Register the realm name with Steel-Belted Radius. Optionally, you can use wild cards to specify matching rules for realms, and you can specify the default realm for undecorated User-Name attributes	proxy.ini [Realms] Realm1

Step	Proxy RADIUS Configuration Task	File and Section

		Realm2 = *.msn.com
		Realm3 = <undecorated>

3	Create a realm configuration file for each realm you register.	Realm1.pro Realm2.pro Realm3.pro
4	Study the customer's current (or planned) RADIUS configuration. The customer's RADIUS servers are the target servers in the new realm. • Are authentication and accounting packets directed to different RADIUS servers? • What is their need for a fast-fail policy, primary-secondary server strategy, or round-robin load balancing? • Are some servers used for authentication and some for accounting? • What is the IP address of each RADIUS server? • What UDP port and shared secret does each server use for authentication and/or accounting?	—
5	Does the customer want its RADIUS servers to receive Accounting-On and Accounting-Off messages? If so, add the new realm to your static proxy accounting configuration. See "**Static Proxy Accounting**".	proxy.ini [StaticAcct] 7=name 8=name [name] realm=RealmName
6	Use the SBR Administrator program to create a proxy entry for each target in the new realm. For authentication targets, verify that the **Make available as an authentication method** check box is unchecked.	Add Proxy Target dialog Edit Proxy Target dialog
7	Give the customer the IP address of the Steel-Belted Radius server as well as the UDP port and shared secret it uses for authentication and accounting. Instruct the customer that for each target in the new realm, the Steel-Belted Radius server must be added to the target's database as a RADIUS client. Presumably, someone at the customer site performs this task by running the target server's RADIUS configuration utility.	—
8	Enable authentication in this realm.	RealmName.pro [Auth] Enable=1
9	(Optional) Indicate that any realm names and delimiters are to be stripped from the User-Name before it is sent to the target server for authentication. • A value of 0 indicates realm names should not be stripped. • A value of 1 indicates realm names should be stripped.	StripRealm=
10	Specify which target servers receive authentication packets.	

Step	Proxy RADIUS Configuration Task	File and Section
	Configure load balancing and other details of realm and target selection for authentication packets. This is a multi-step process: (1) In the [Auth] section of the RealmName.pro file, set Enable to 1 and assign a name to the TargetsSection parameter; (2) create a [name] section in the file; and (3) within this section list the targets for authentication. When listing a target, use the name you assigned to it in the Proxy dialog.	TargetsSection=name M [name] Server=
11	(Optional) Specify an attribute filter to apply to authentication requests going out to the realm from Steel-Belted Radius. This is a multi-step process: (1) In the [Auth] section of RealmName.pro, assign a name to the FilterOut parameter; (2) create a [name] section in the filter.ini file; and (3) within the filter.ini [name] section list the rules for editing the attributes in a RADIUS authentication request packet before forwarding the packet “out” to a proxy RADIUS realm.	RealmName.pro [Auth] FilterOut=name filter.ini [name] M
12	(Optional) Specify an attribute filter to apply to authentication responses returning into Steel-Belted Radius from the realm. This is a multi-step process: (1) In the [Auth] section of RealmName.pro, assign a name to the FilterIn parameter; (2) create a [name] section in the filter.ini file; and (3) within the filter.ini [name] section list the rules for editing the attributes in an authentication response packet as it returns “in” from the proxy RADIUS realm, before relaying the packet back to the RADIUS client.	RealmName.pro [Auth] FilterIn=name filter.ini [name] M
13	Enable proxy RADIUS accounting in this realm.	RealmName.pro [Acct] Enable=1
14	(Optional) Indicate that any realm names and delimiters are to be stripped from the User-Name before it is sent to the target server for accounting. • A value of 0 indicates realm names should not be stripped. • A value of 1 indicates realm names should be stripped.	StripRealm=1
15	(Optional) Indicate that accounting attributes should be logged locally on the Steel-Belted Radius server as well as being directed to the realm. • A value of 0 indicates accounting attributes should not be logged locally. • A value of 1 indicates accounting attributes should be logged locally.	RecordLocally=1
16	Specify which target servers receive accounting packets. Configure load balancing and other details of realm and target selection for accounting packets. This is a multi-step process: (1) In the [Acct] section of the RealmName.pro file, set Enable to 1 and assign a name to the TargetsSection parameter; (2) create a [name] section in the file; and (3) within this section list the targets for accounting. When listing a target, use the name you assigned to it in the	TargetsSection=name M [name] Server=

Step	Proxy RADIUS Configuration Task	File and Section
	Proxy dialog.	
17	(Optional) Specify an attribute filter to apply to accounting requests going out to the realm from Steel-Belted Radius. This is a multi-step process: (1) In the [Acct] section of **RealmName.pro,** assign a name to the FilterOut parameter; (2) create a [name] section in the **filter.ini** file; and (3) within the **filter.ini** [name] section list the rules for editing the attributes in a RADIUS accounting request packet before forwarding the packet "out" to a proxy RADIUS realm.	RealmName.pro [Acct] FilterOut=name filter.ini [name]
18	(Optional) Specify an attribute filter to apply to accounting responses returning into Steel-Belted Radius from the realm. This is a multi-step process: (1) In the [Acct] section of **RealmName.pro**, assign a name to the FilterIn parameter; (2) create a [name] section in the **filter.ini** file; and (3) within the **filter.ini** [name] section list the rules for editing the attributes in an accounting response packet as it returns "in" from the proxy RADIUS realm, before relaying the packet back to the RADIUS client.	RealmName.pro [Acct] FilterIn=name filter.ini [name]
19	(Optional) Provide DNIS information for this realm.	RealmName.pro [Called-Station-ID]
20	(Optional) Specify a proxy fast-fail policy for the realm.	[FastFail]
21	(Optional) Enable Steel-Belted Radius to map the presence or absence of certain attributes or values to this realm.	proxy.ini [AuthAttributeMap] RealmName [AcctAttributeMap] RealmName
22	It's possible to load your new realm configuration dynamically, without stopping and restarting the server. Under Linux: Issue the HUP signal to the Steel-Belted Radius process: kill -HUP ProcessID Under Windows: Run the RADHUP.EXE program from the command shell. (RADHUP.EXE is located in the server directory that you specified at installation time, usually C:\Program Files (x86)\Pulse Secure\Steel-Belted Radius\Service.) Steel-Belted Radius re-reads proxy.ini, filter.ini, and all .pro files in the server directory, and resets its realm configuration accordingly. ℹ Note: Rarely, you must edit radius.ini while configuring a realm. If you do edit radius.ini, you must stop and restart Steel-Belted Radius before your new configuration is fully loaded	—

Configuring a Directed Realm

Table 28 traces the process of configuring a directed authentication and/or accounting realm for Steel-Belted Radius. **Table 28** also lists the sections that you must edit in Steel-Belted Radius configuration files to accomplish each step. No step in this process should be omitted unless the table indicates that the step is optional.

Table 28: Configuring a Directed Realm

Step	Directed Realm Configuration Task	File and Section
1	Complete the steps outlined in "**Stage One of Realm Configuration**".	—
2	Register the RealmName with Steel-Belted Radius. Optionally, you can use wild cards to specify matching rules for realms, and you can specify the default realm for undecorated User-Name attributes.	proxy.ini [Directed] Realm1 Realm2 = *.msn.com Realm3 = <undecorated>
3	Create a realm configuration file.	RealmName.dir
4	Add the customer's user data to your database, which might be an external database (SQL, LDAP) or the Steel-Belted Radius database. For information on how to add a limited number of users, see "**Administering Users via Legacy SBR**". For information on adding users in batches, see **Appendix E".** See also "**LDAP Configuration Interface**".	—
5	Configure the authentication method in Steel-Belted Radius. See "**Setting Up EAP Authentication Policies**" See "**Configuring SQL Authentication**" and "**Configuring LDAP Authentication**".	—
6	Register the authentication method with the realm.	RealmName.dir [AuthMethods]
7	Enable directed authentication in the realm.	[Auth] Enable= 1
8	(Optional) Indicate that any realm names and delimiters are to be stripped from the User-Name before authentication is performed. • A value of 0 indicates realm names should not be stripped. • A value of 1 indicates realm names should be stripped.	StripRealm=
9	Understand the data that the customer uses (or plans to use) to store accounting and billing records. This indicates the accounting method(s) to use.	—
10	Configure the accounting method(s) in Steel-Belted Radius. For more information, refer to "proxy.ini File" in the Steel-Belted Radius Reference Guide.	
10a	You can set up unique accounting log files by copying	

Step	Directed Realm Configuration Task	File and Section
	account.ini from the server directory to another directory, renaming it (if desired, but keep the .ini extension), and editing it to record accounting attributes by each customer. Use account.ini file syntax. For more information, refer to "account.ini File" in the ***Steel-Belted Radius Reference Guide.***	
10b	You can log to external SQL databases by copying an .acc file from the server directory to another directory, renaming it (if desired, but keep the .acc extension), and editing it to record accounting attributes by each customer. Use .acc file syntax. See "**About SQL Accounting**".	.acc files
11	Name each accounting method.	proxy.ini [DirectedAcct Methods]
12	Register the accounting method with the realm.	RealmName.dir [AcctMethods]
13	Enable directed accounting in the realm.	[Acct] Enable= 1
14	(Optional) Indicate that any realm names and delimiters are to be stripped from the User-Name before accounting is performed. • A value of 0 indicates realm names should not be stripped. • A value of 1 indicates realm names should be stripped.	StripRealm=
15	(Optional) Indicate that accounting attributes should be logged locally on the Steel-Belted Radius server as well as being directed to the realm. • A value of 0 indicates accounting attributes should not be logged locally. • A value of 1 indicates accounting attributes should be logged locally.	RecordLocally=
16	(Optional) Provide DNIS information for this realm.	[Called-Station-ID]
17	Load your new configuration. If you've added or changed any directed accounting methods, you must stop and restart the server. If you've added or changed directed authentication methods in which external database (SQL or LDAP) authentication is used, you must stop and restart the server.	
18	If you've added or changed directed authentication methods in which local or pass-through (Local, UNIX, Domain, Host, SecurID, or TACACS+) authentication is used, it's possible to load your new realm configuration dynamically, without stopping and restarting the server. Under Linux: Issue the HUP signal to the Steel-Belted Radius process: kill -HUP ProcessID Under Windows: Run the RADHUP.EXE program from the	

Step	Directed Realm Configuration Task	File and Section
	command shell. (RADHUP.EXE is located in the server directory that you specified at installation time, usually C:\ Program Files\Pulse Secure\Steel-Belted Radius\Service.) Steel-Belted Radius re-reads proxy.ini and all .dir files in the server directory, and resets its realm configuration accordingly. Note: Rarely, you must edit radius.ini while configuring a realm. If you do edit radius.ini, you must stop and restart the Steel-Belted Radius before your new configuration is fully loaded.	

Editing the radius.ini Realm Settings

1. Use the [Configuration] section of radius.ini to enable or disable realm features for Steel-Belted Radius: ExtendedProxy and AttributeEdit. Both fields are enabled (set to 1) by default. You can disable either feature by setting the corresponding field to 0.

2. Use the [Self] section of radius.ini to list all of the realm names that should be handled by this Steel-Belted Radius server, rather than being proxied to other targets.

3. As with all changes to radius.ini, if you edit radius.ini while configuring a realm, you must stop and restart the Steel-Belted Radius before your new realm configuration is fully loaded.

Editing the proxy.ini File

The proxy.ini file specifies the order of realm selection methods, the realm selection rules, and other settings for all realms on the server. Settings for a realm are provided in its RealmName.pro or RealmName.dir file.

After you edit proxy.ini, you must apply your changes as follows:

- If you have configured any proxy RADIUS realms, it's possible to load your new realm configuration dynamically, without stopping and restarting the server.

  - Linux: Issue the HUP signal to the Steel-Belted Radius process: kill -HUP ProcessID

  - Windows: Run the RADHUP.EXE program from the command shell. (RADHUP.EXE is located in the server directory that you specified at installation time, usually C:\Program Files (x86)\Pulse Secure\Steel-BeltedRadius\Service.)

  Steel-Belted Radius re-reads proxy.ini, filter.ini, and all *.pro and *.dir files in the server directory, and resets its realm configuration accordingly.

- If you have configured any directed realms and if you have added or changed:

  - Any directed accounting methods at all, you must stop and restart the server to load your new configuration.

  - Directed authentication methods in which external database (SQL or LDAP) authentication is used, you must stop and restart the server to load your new configuration.

  - Directed authentication methods in which local or pass-through (Local, UNIX, Domain, Host, SecurID, or TACACS+) authentication is used, it's possible to load your new realm configuration by using a HUP signal.

Note: Refer to the Steel-Belted Radius Reference Guide for syntax information for the proxy.ini and proxyrl.ini files.

Note: If you edit radius.ini while configuring a realm, you must stop and restart the server before your new configuration is fully loaded.

Setting Up Smart Static Accounting

The proxyrl.ini file supports a feature called smart static accounting, which allows you to specify that the accounting packets for a proxy or directed realm should be forwarded to a list of one or more proxy or directed realms. These groups of realms can also be used for static accounting configured in proxy.ini.

This file consists of a number of sections that you name. Each section name is referenced in the StaticAcctRealms field in the [Acct] section of a .pro or .dir file. Following the section name, you can list a number of proxy realm names, in the following format:

```
[realm-list-name-1]
proxy-realm-1
proxy-realm-2
:
[realm-list-name-2]
:
```

For example:

```
[StaticAcctTargets1]
AcctSrvr1
AcctSrvr4
```

Note: To avoid an infinite loop, the list of static accounting servers must not include realms that use the list. If you include a realm in a list of static accounting servers and specified the same realm in proxy.ini as doing static accounting, the realm receives duplicate accounting packets.

Setting Up Proxy RADIUS Realms

For each proxy RADIUS realm that you want to configure in Steel-Belted Radius, you must create a file called RealmName.pro, where RealmName is the name of the realm, and you must add this RealmName to the [Realms] section of the proxy.ini file.

If you create or edit a RealmName.pro file, you can apply your configuration changes dynamically, without stopping the server. Depending on your operating system:

- Under **Linux**: Issue the HUP signal to the Steel-Belted Radius process: kill -HUP ProcessID
- Under **Windows:** Run the **RADHUP.EXE** program from the command shell. (RADHUP.EXE is located in the server directory that you specified at installation time, usually **C:\Program Files (x86)\Pulse Secure\Steel-Belted Radius\Service.)**

After you do this, Steel-Belted Radius re-reads proxy.ini, filter.ini, and all .pro and .dir files in the server directory, and resets its realm configuration accordingly.

Note: Rarely, you must edit radius.ini while configuring a realm. If you do edit radius.ini, you must stop and restart Steel-Belted Radius before your new configuration is fully loaded.

Refer to the Steel-Belted Radius Reference Guide for field information for a *.pro file.

Configuration Tasks

To set up a Realmname.pro file:

1. Specify proxy RADIUS target selection rules.

   Each [name] section of a RealmName.pro file specifies a set of rules that Steel-Belted Radius can use to select a target for proxy-forwarding within the proxy RADIUS realm. Each [name] section consists of a list of target servers. For any particular request, if the first listed server fails to respond (or is presumed down), then the other servers are tried in the order listed. A [name] section is activated by referencing it from the [Auth] and/or [Acct] sections.

2. Optionally, configure round-robin load balancing.

   If you have multiple target servers in a realm, you can select whether to use them in round-robin fashion (load balancing), primary/backup fashion, or a combination of both. The value of the RoundRobin entry in the [Auth] or [Acct] section indicates the number of targets that are to be used in round-robin fashion. Refer to the Steel-Belted Radius Reference Guide for information on configuring round-robin options.

3. Configure proxy RADIUS fast-fail options.

   You can use the [FastFail] section of a realm configuration file to fine-tune retry policies for individual realms, and for specific targets within a realm. If you provide a [FastFail] section, the ProxyFastFail parameter in the radius.ini [Configuration] section is ignored.

4. Specify username decoration options.

   You can use the [ModifyUser] section of a realm configuration file to decorate a realm, where the realm is determined by other means, such as DNIS or attribute mapping. For example, if george@ gm and george@ford. are both in the RADIUS database, either user could log in as george, as Steel-Belted Radius would determine the realm, for example, by DNIS. Based on the realm, Steel-Belted Radius would append either @gm or @ford to the username, and then use the Local User directed method to authenticate.

Setting Up Directed Realms

A directed realm specifies target methods for directed authentication and/or directed accounting. Its realm configuration file is called RealmName.dir.

The directed authentication feature permits the server to bypass its authentication methods list and map an incoming RADIUS request to one or more specific authentication methods. Steel-Belted Radius chooses the destination method based on routing information found in the request packet. The destination methods might be any authentication methods already configured on the local Steel-Belted Radius server, regardless of how they were configured; for example, a method might have been configured using the SBR Administrator dialogs, the LDAP configuration interface, or an .aut configuration file.

If no directed authentication method is configured, every request percolates through the same authentication methods list, as defined in the Authentication Policies panel in SBR Administrator. Directed authentication allows you to tailor an authentication methods list to a customer's needs.

Directed accounting is also possible. The destination accounting method might be the Steel-Belted Radius accounting log, an external database configured using an .acc file, or a distinct accounting log file that contains entries only for this customer.

To activate these features, you must create RealmName.dir files, place them in the Steel-Belted Radius directory, and list them in the [Directed] section of proxy.ini. Subsequently, any requests that arrive addressed to one of these realm names are processed on the local server using the instructions you have provided in proxy.ini and in the corresponding RealmName.dir file.

After you edit a RealmName.dir file, you must apply your changes as follows. If you have added or changed:

- Any directed accounting methods at all, you must stop and restart the server to load your new configuration.
- Directed authentication methods in which external database (SQL or LDAP) authentication is used, you must stop and restart the server to load your new configuration.
- Directed authentication methods in which local or pass-through (Local, UNIX, Domain, Host, SecurID, or TACACS+) authentication is used, you can apply your configuration changes dynamically, without stopping the server. Depending on your operating system:
  - Under Linux: Issue the HUP signal to the Steel-Belted Radius process: kill -HUP ProcessID
  - Under Windows: Run the RADHUP.EXE program from the command shell. (RADHUP.EXE is located in the server directory that you specified at installation time, usually C:\Program Files (x86)\Pulse Secure\Steel-Belted Radius\Service.)

  Steel-Belted Radius re-reads proxy.ini, filter.ini, and all .pro and .dir files in the server directory, and resets its realm configuration accordingly.

  Note: Rarely, you must edit radius.ini while configuring a realm. If you edit radius.ini, you must stop and restart Steel-Belted Radius before your new configuration is fully loaded.

How to Update Realm Configuration

The following information explains when a HUP signal (RADHUP.EXE under Windows) is sufficient, or insufficient, for updating realm configuration:

- A HUP signal is sufficient to load any changes that you make to proxy.ini, filter.ini, or *.pro files for the purpose of configuring proxy RADIUS realms.
- However, when you configure directed realms (proxy.ini, *.dir files, and possibly *.acc, *.aut, and accounting *.ini files as well) you must load configuration changes as follows. If you have added or changed:
  - Any directed accounting methods at all, you must stop and restart the server.
  - Directed authentication methods in which external database (SQL or LDAP) authentication is used, you must stop and restart the server.
  - Directed authentication methods in which local or pass-through (Local, UNIX, Domain, Host,

SecurID, or TACACS+) authentication is used, a HUP signal is sufficient.

Chapter 22
Setting Up Filters via Legacy SBR Administrator

This chapter describes how to set up and use filters in Steel-Belted Radius via legacy SBR administrator.

Note: If you are running the Global Enterprise Edition (GEE) of Steel-Belted Radius, you can use a separately licensed add-on module to use Javascript to select and create filters. For more
Information, refer to the Steel-Belted Radius Scripting Guide.

Overview

A filter is a collection of rules for adding, modifying, or removing attributes or attribute values in RADIUS requests and responses. You define filters and their rules by means of the Adding a Filter. You enable filters by referring to them by name when using the SBR Administrator or when editing certain .ini file sections.

A filter consists of one or more rules, which are processed in sequential order.

- Add rules specify that an attribute-value pair (AVP) is added to a RADIUS packet during processing. The AVP is added after all other rules are processed. An attribute is added to a packet only if it is legal to do so.

  Some attributes can appear only once in a RADIUS packet; others can appear multiple times. If an attribute that is the subject of an Add rule is already present in the packet (after processing Allow and Exclude rules) and the attribute can only appear once, the Add rule is not processed and the second instance of the attribute is not added.

- Allow rules to specify whether an attribute (or AVP) is allowed in a RADIUS packet.
  - If an Allow rule specifies an attribute name and an attribute value, then only attributes of the specified type and value are allowed in the RADIUS packet.
  - If an Allow rule specifies an attribute name without an attribute value, then all attributes of the specified type, regardless of value, are allowed in the RADIUS packet.
  - If an Allow rule does not specify an attribute name, then all attributes, regardless of value, are allowed in the RADIUS packet.
- Exclude rules specify an attribute (or AVP) is excluded from a RADIUS packet.
  - If an Exclude rule specifies an attribute name and an attribute value, then only attributes of the specified type and value are excluded from the RADIUS packet.
  - If an Exclude rule specifies an attribute name without an attribute value, then all attributes of the specified type, regardless of value, are excluded from the RADIUS packet.
  - If an Exclude rule does not specify an attribute name, then all attributes, regardless of value, are excluded from the RADIUS packet.
- Replace rules specify the conditions whereby one attribute (or attribute value) is replaced with another.

  - If a Replace rule specifies that one named attribute of a specified value (attr1 v1) should be replaced with a different attribute of a specified value (attr2 v2) , then any occurrence of the first AVP is replaced with the second AVP. Result: attr2 v2.
  - If a Replace rule specifies that a named attribute without a specified value (attr1) should be replaced with a different attribute of a specified value (attr2 v2) , then any occurrence of the first attribute (regardless of value) is replaced with the second AVP. Result: attr2 v2.
  - If a Replace rule specifies that one named attribute of a specified value (attr1 v1) should be replaced with a different attribute without a specified value (attr2), then any occurrence of the first attribute is replaced with the second attribute, which retains the value of the original attribute. Result: attr2 v1.
  - If a Replace rule specifies that one named attribute (without a specified value) should be replaced with a different attribute without a specified value, then any occurrence of the first attribute is replaced with the second attribute, which retains the value of the original attribute. Result: attr2 v1.
- Script rules specify when to run attribute filter scripts. For information on attribute filter scripts, refer to the Steel-Belted Radius Scripting Guide.

The Steel-Belted Radius dictionary file radius.dct provides string aliases for certain integer values defined in the RADIUS standard. You can use these strings in attribute filter rules.

Note: Filter rules provide you with tremendous flexibility. However, Steel-Belted Radius does not prevent you from creating an invalid RADIUS packet. Some attributes are not appropriate for certain types of requests. For example, adding a pooled Framed-Ip-Address attribute to an accounting request could cause a loss of available IP addresses.

Order of Filter Rules

The order of rules within a filter is important. General default rules that take no parameters, such as Allow (allow all attributes unless otherwise specified) or Exclude (exclude all attributes unless otherwise specified) must appear as the first rule in the filter. Later rules supersede earlier rules; the last applicable rule “wins.” Add and Replace rules are applied after the Allow and Exclude rules.

More specific rules with more parameters (Add attribute value) act as exceptions to less specific rules with fewer parameters (Allow attribute, EXCLUDE). For example, you might want to allow a certain attribute and exclude one or more specific values for that attribute. Or you might exclude all attributes, allow specific attributes, and add specific attribute/value pairs.

Note: Script rules are not subject to rule ordering.

You can use two basic approaches to designing a filter:

- Start the rule list with a default Exclude rule (no parameters) and add Allow rules for any attributes or attribute/value pairs that you want to insert into the packet. Add and Replace rules might be used.
- Start the rule list with a default Allow rule (no parameters) and add Exclude rules for any attributes or attribute/value pairs that you want to remove from the packet. Add and Replace rules might be used.

The default action for Steel-Belted Radius is Exclude. If a filter does not contain any rules, the filter removes all attributes from a packet when the filter is applied.

Values in Filter Rules

The value of an attribute is interpreted based on the type of the attribute in its attribute dictionary. **Table 29** lists the meaning of each attribute type.

Table 29: Filter Rule Values

Attribute Type	Function
hexadecimal	A hexadecimal value is specified as a string. Special characters might be included using escape codes.
int1, int4, integer	1- or 4-byte unsigned decimal number (integer is equivalent to int4). Note: The Steel-Belted Radius dictionary file radius.dct provides string aliases for certain integer values defined in the RADIUS standard. You can use these strings in attribute filter rules.
ipaddr, ipaddr-pool	An IP address in dotted notation; for example: EXCLUDE NAS-IP-Address 127.0.0.1
ipxaddr-pool	A sequence of hex digits; for example: ALLOW Framed-IPX-Network 0042A36B
string	String attribute (includes null terminator). A string is specified as text. The text can be enclosed in double-quotes (“). The text is interpreted as a regular expression. Backslash (\) is the escape character. Escape codes are interpreted as follows: Code Meaning \a 7 \b 8 \f 12 \n 10 \r 13 \t 9 \v 11 \nnn nnn is a decimal value between 0 and 255 \xnn nn is a hexadecimal value between 00 and FF \c c is a single character, interpreted literally Literal backslashes (\) within a string and double-quotes (“) within quoted strings should be prefixed with an escape character. For example: ADD Reply-Message Session limit is one hour ADD Reply-Message “Session limit is one hour” ADD Reply-Message “Your user name is \”George\””
time	A time value is specified with a string indicating date and time: yyyy/mm/dd hh:mm:ss The date portion is mandatory; the time portion can be specified to whatever degree of precision is required, or can be omitted entirely. For example:

Attribute Type	Function
	2006/4/3 14:00:00 and 2006/4/3 14 both refer to April 3, 2006 at 2:00 p.m. For example: ADD Ascend-PW-Expiration 2006/4/3

Referencing Attribute Filters

Steel-Belted Radius attribute filtering provides flexibility in packet processing. You reference filters by name in SBR Administrator dialogs, in various .ini and .aut configuration files, and in the FilterOut and FilterIn sections of your .pro and .dir files. You can use the same filter for all packets in all realms. You can apply filtering to some realms, and not others. To disable filtering for a realm, omit filtering parameters from the *.pro or *.dir files and from the EAP-PEAP/EAP-TTLS configurations. Filtering is often used only for packets that are routed “out” to realms (the FilterOut parameter).

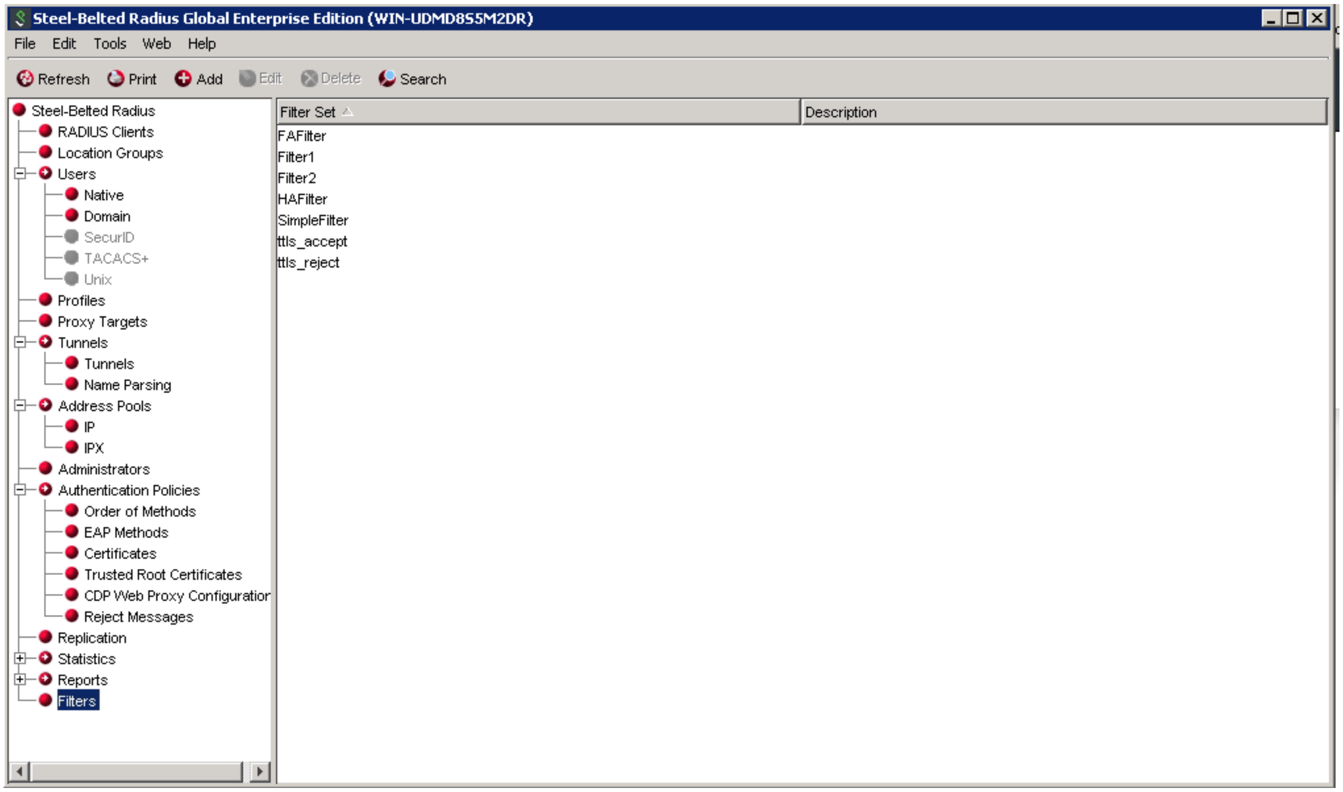
To reference filtering rules in proxy or directed realm configurations, you must use the FilterOut and FilterIn parameters in the [Auth] and [Acct] sections of a realm configuration file. For more information, refer to the Steel-Belted Radius Reference Guide.

ℹ Note: Do not allocate IP addresses from Steel-Belted Radius IP address pools in accounting filters. These addresses will be allocated but never released.

Filters Panel

You can use the Filters panel (Figure 117: Filters Panel) to display the filters configured for Steel-Belted Radius. To open the Filters panel, click Filters in the SBR Administrator sidebar.

Figure 117: Filters Panel



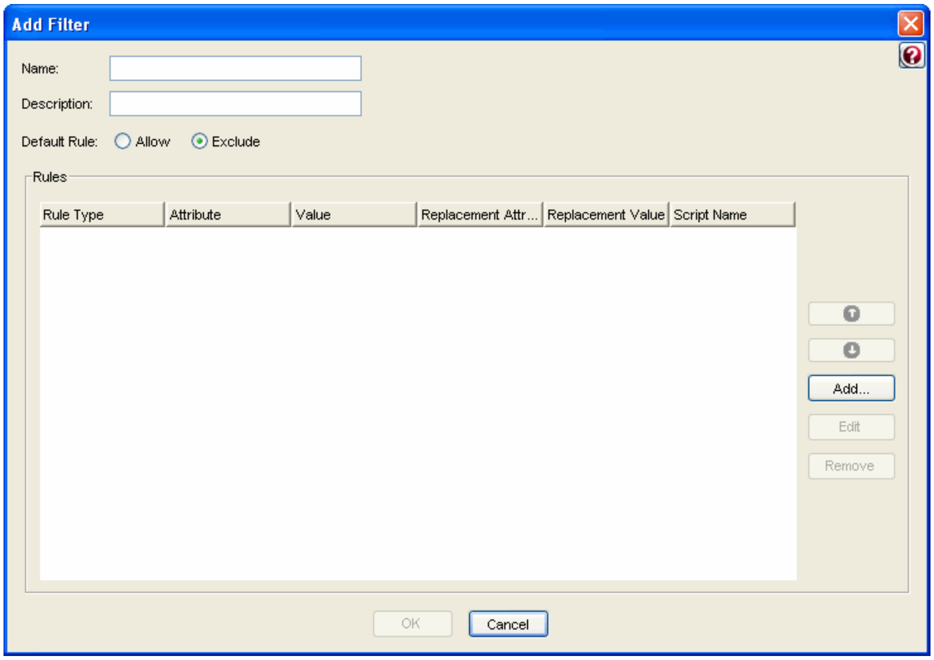
Filter Set	Description
FAFilter	
Filter1	
Filter2	
HAFilter	
SimpleFilter	
ttls_accept	
ttls_reject	

Adding a Filter

To add a filter:

1. Click Filters in the SBR Administrator sidebar to display the Filter panel.
2. Click Add in the toolbar to display the Add Filter dialog (Figure 118: Add Filter Dialog).

Figure 118: Add Filter Dialog



Add Filter

Name:

Description:

Default Rule: ○ Allow ● Exclude

Rules

Rule Type	Attribute	Value	Replacement Attr...	Replacement Value	Script Name

3. Enter the name of the filter in the **Name** field.

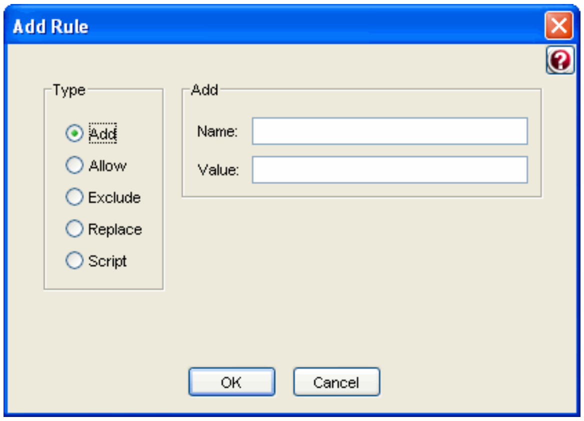
4. Optionally, enter a description of the filter in the **Description** field.

   You can enter as many as 4095 characters for a filter description.

5. Use the **Default Rule** radio buttons to specify whether attributes should be allowed or excluded if no other rule applies to a RADIUS packet.

6. Click the **Add** button to display the Add Rule dialog.

Figure 119: Add Rule Dialog



Add Rule

Type

● Add
○ Allow
○ Exclude
○ Replace
○ Script

Add

Name:

Value:

7. Select the type of rule you want to add to the filter.

   Options are Add, Allow, Exclude, Replace, and Script. The fields in the Add Rule dialog depend on the option you select.

8. Specify the attribute name and value settings you want to use for the rule.

(GEE): Ifyou edit the filters for a Steel-Belted Radius server, you can apply your configuration changes without stopping the server:

- Linux: Issue the HUP signal to the Steel-Belted Radius process: kill -HUP ProcessID
- Windows: Run the radhup.exe program from the command shell. (radhup.exe is located in the server directory that you specified at installation time, usually C:\Radius\Service.)

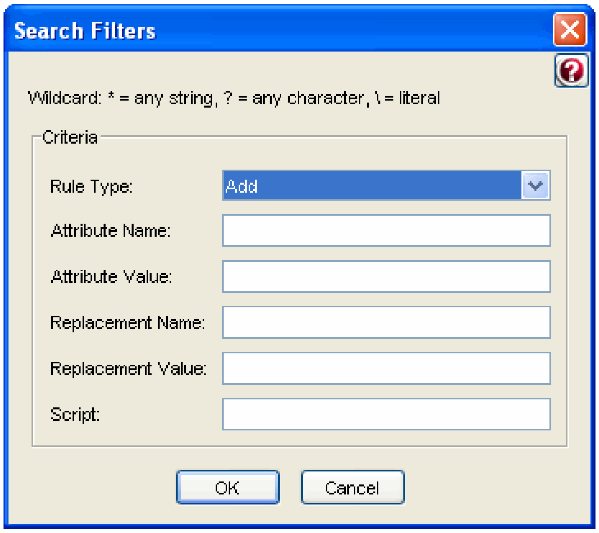
Searching the Filter List

You can search your list of filters to identify those of a specific type or that use a specific rule. To search your filter list:

1. Open the Filters panel.
2. Click the **Search** toolbar button.
3. When the Search Filters dialog (Figure 120: Search Filters Dialog) appears, enter one or more search criteria in the fields provided.

   You can use the question mark (?) wildcard to represent one character in a string. You can use the asterisk (*) wildcard to represent any number of characters. For example, entering the search string MS-MPPE-* would identify any filter that looks for the MS-MPPE-Recv-Key or MS-MPPE-Send-Key attribute.

Figure 120: Search Filters Dialog



Wildcard: * = any string, ? = any character, \ = literal

Criteria

Rule Type: Add

Attribute Name:

Attribute Value:

Replacement Name:

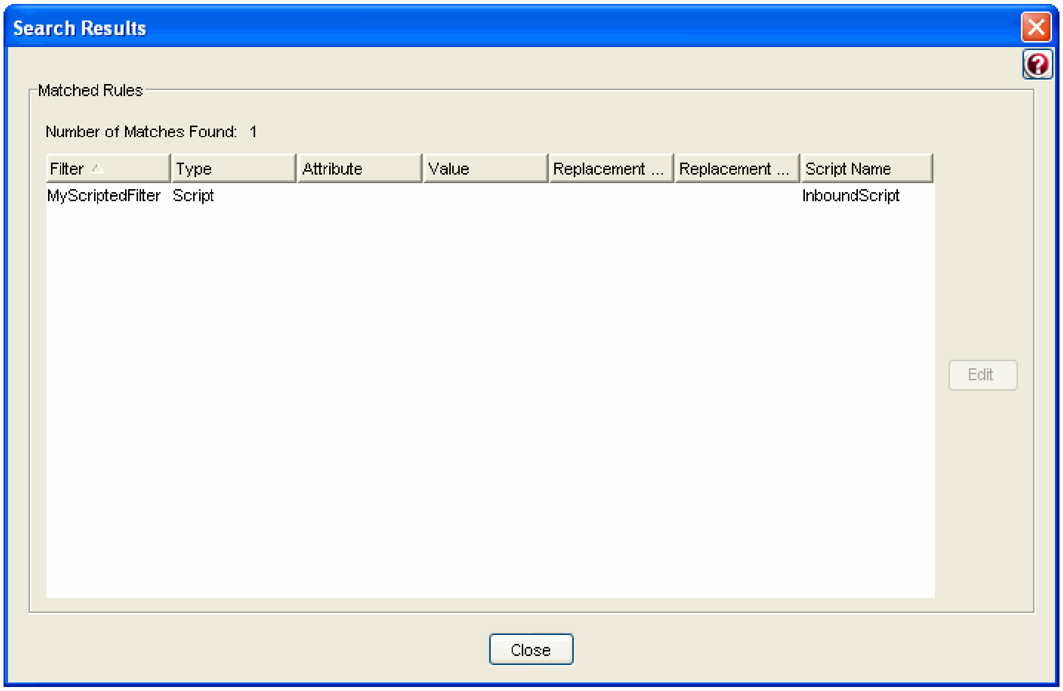
Replacement Value:

Script:

4. Click **OK.**

   The Search Results dialog (Figure 121: Search Results Dialog) displays the list of filters that satisfy your search criteria.

Figure 121: Search Results Dialog



Number of Matches Found: 1

Filter	Type	Attribute	Value	Replacement ...	Replacement ...	Script Name
MyScriptedFilter	Script					InboundScript

Chapter 23
Setting Up Filters via WebGUI

This chapter describes how to set up and use filters in Steel-Belted Radius via WebGUI.

Note: If you are running the Global Enterprise Edition (GEE) of Steel-Belted Radius, you can use a separately licensed add-on module to use Javascript to select and create filters. For more information, refer to the Steel-Belted Radius Scripting Guide.

Overview

A filter is a collection of rules for adding, modifying, or removing attributes or attribute values in RADIUS requests and responses. You define filters and their rules by means of the Adding a Filter. You   enable filters by referring to them by name when using the SBR Administrator or when editing certain .ini file sections.

A filter consists of one or more rules, which are processed in sequential order.

- Add rules specify that an attribute-value pair (AVP) is added to a RADIUS packet during processing. The AVP is added after all other rules are processed. An attribute is added to a packet only if it is legal to do so.

  Some attributes can appear only once in a RADIUS packet; others can appear multiple times. If an attribute that is the subject of an Add rule is already present in the packet (after processing Allow and Exclude rules) and the attribute can only appear once, the Add rule is not processed and the second instance of the attribute is not added.

- Allow rules to specify whether an attribute (or AVP) is allowed in a RADIUS packet.
  - If an Allow rule specifies an attribute name and an attribute value, then only attributes of the specified type and value are allowed in the RADIUS packet.
  - If an Allow rule specifies an attribute name without an attribute value, then all attributes of the specified type, regardless of value, are allowed in the RADIUS packet.
  - If an Allow rule does not specify an attribute name, then all attributes, regardless of value, are allowed in the RADIUS packet.
- Exclude rules specify an attribute (or AVP) is excluded from a RADIUS packet.
  - If an Exclude rule specifies an attribute name and an attribute value, then only attributes of the specified type and value are excluded from the RADIUS packet.
  - If an Exclude rule specifies an attribute name without an attribute value, then all attributes of the specified type, regardless of value, are excluded from the RADIUS packet.
  - If an Exclude rule does not specify an attribute name, then all attributes, regardless of value, are excluded from the RADIUS packet.
- Replace rules specify the conditions whereby one attribute (or attribute value) is replaced with another.

  - If a Replace rule specifies that one named attribute of a specified value (attr1 v1) should be replaced with a different attribute of a specified value (attr2 v2) , then any occurrence of the first AVP is replaced with the second AVP. Result: attr2 v2.
  - If a Replace rule specifies that a named attribute without a specified value (attr1) should be replaced with a different attribute of a specified value (attr2 v2) , then any occurrence of the first attribute (regardless of value) is replaced with the second AVP. Result: attr2 v2.
  - If a Replace rule specifies that one named attribute of a specified value (attr1 v1) should be replaced with a different attribute without a specified value (attr2), then any occurrence of the first attribute is replaced with the second attribute, which retains the value of the original attribute. Result: attr2 v1.
  - If a Replace rule specifies that one named attribute (without a specified value) should be replaced with a different attribute without a specified value, then any occurrence of the first attribute is replaced with the second attribute, which retains the value of the original attribute. Result: attr2 v1.
- Script rules specify when to run attribute filter scripts. For information on attribute filter scripts, refer to the Steel-Belted Radius Scripting Guide.

The Steel-Belted Radius dictionary file radius.dct provides string aliases for certain integer values defined in the RADIUS standard. You can use these strings in attribute filter rules.

ℹ Note: Filter rules provide you with tremendous flexibility. However, Steel-Belted Radius does not prevent you from creating an invalid RADIUS packet. Some attributes are not appropriate for certain types of requests. For example, adding a pooled Framed-Ip-Address attribute to an accounting request could cause a loss of available IP addresses.

Order of Filter Rules

The order of rules within a filter is important. General default rules that take no parameters, such as Allow (allow all attributes unless otherwise specified) or Exclude (exclude all attributes unless otherwise specified) must appear as the first rule in the filter. Later rules supersede earlier rules; the last applicable rule “wins.” Add and Replace rules are applied after the Allow and Exclude rules.

More specific rules with more parameters (Add attribute value) act as exceptions to less specific rules with fewer parameters (Allow attribute, EXCLUDE). For example, you might want to allow a certain attribute and exclude one or more specific values for that attribute. Or you might exclude all attributes, allow specific attributes, and add specific attribute/value pairs.

ℹ Note: Script rules are not subject to rule ordering.

You can use two basic approaches to designing a filter:

- Start the rule list with a default Exclude rule (no parameters) and add Allow rules for any attributes or attribute/value pairs that you want to insert into the packet. Add and Replace rules might be used.
- Start the rule list with a default Allow rule (no parameters) and add Exclude rules for any attributes or attribute/value pairs that you want to remove from the packet. Add and Replace rules might be used.

The default action for Steel-Belted Radius is Exclude. If a filter does not contain any rules, the filter removes all attributes from a packet when the filter is applied.

Values in Filter Rules

The value of an attribute is interpreted based on the type of the attribute in its attribute dictionary. **Table 29** lists the meaning of each attribute type.

Table 29: Filter Rule Values

Attribute Type	Function
hexadecimal	A hexadecimal value is specified as a string. Special characters might be included using escape codes.
int1, int4, integer	2- or 4-byte unsigned decimal number (integer is equivalent to int4). Note: The Steel-Belted Radius dictionary file radius.dct provides string aliases for certain integer values defined in the RADIUS standard. You can use these strings in attribute filter rules.
ipaddr, ipaddr-pool	An IP address in dotted notation; for example: EXCLUDE NAS-IP-Address 127.0.0.1
ipxaddr-pool	A sequence of hex digits; for example: ALLOW Framed-IPX-Network 0042A36B
string	String attribute (includes null terminator). A string is specified as text. The text can be enclosed in double-quotes (“). The text is interpreted as a regular expression. Backslash (\) is the escape character. Escape codes are interpreted as follows: Code Meaning \a 7 \b 8 \f 12 \n 10 \r 13 \t 9 \v 11 \nnn nnn is a decimal value between 0 and 255 \xnn nn is a hexadecimal value between 00 and FF \c c is a single character, interpreted literally Literal backslashes (\) within a string and double-quotes (“) within quoted strings should be prefixed with an escape character. For example: ADD Reply-Message Session limit is one hour ADD Reply-Message “Session limit is one hour” ADD Reply-Message “Your user name is \”George\””
time	A time value is specified with a string indicating date and time: yyyy/mm/dd hh:mm:ss The date portion is mandatory; the time portion can be specified to whatever degree of precision is required, or can be omitted entirely. For example:

Attribute Type	Function
	2006/4/3 14:00:00 and 2006/4/3 14 both refer to April 3, 2006 at 2:00 p.m. For example: ADD Ascend-PW-Expiration 2006/4/3

Referencing Attribute Filters

Steel-Belted Radius attribute filtering provides flexibility in packet processing. You reference filters by name in SBR Administrator pages, in various .ini and .aut configuration files, and in the FilterOut and FilterIn sections of your .pro and .dir files. You can use the same filter for all packets in all realms. You can apply filtering to some realms, and not others. To disable filtering for a realm, omit filtering parameters from the *.pro or *.dir files and from the EAP-PEAP/EAP-TTLS configurations. Filtering is often used only for packets that are routed "out" to realms (the FilterOut parameter).

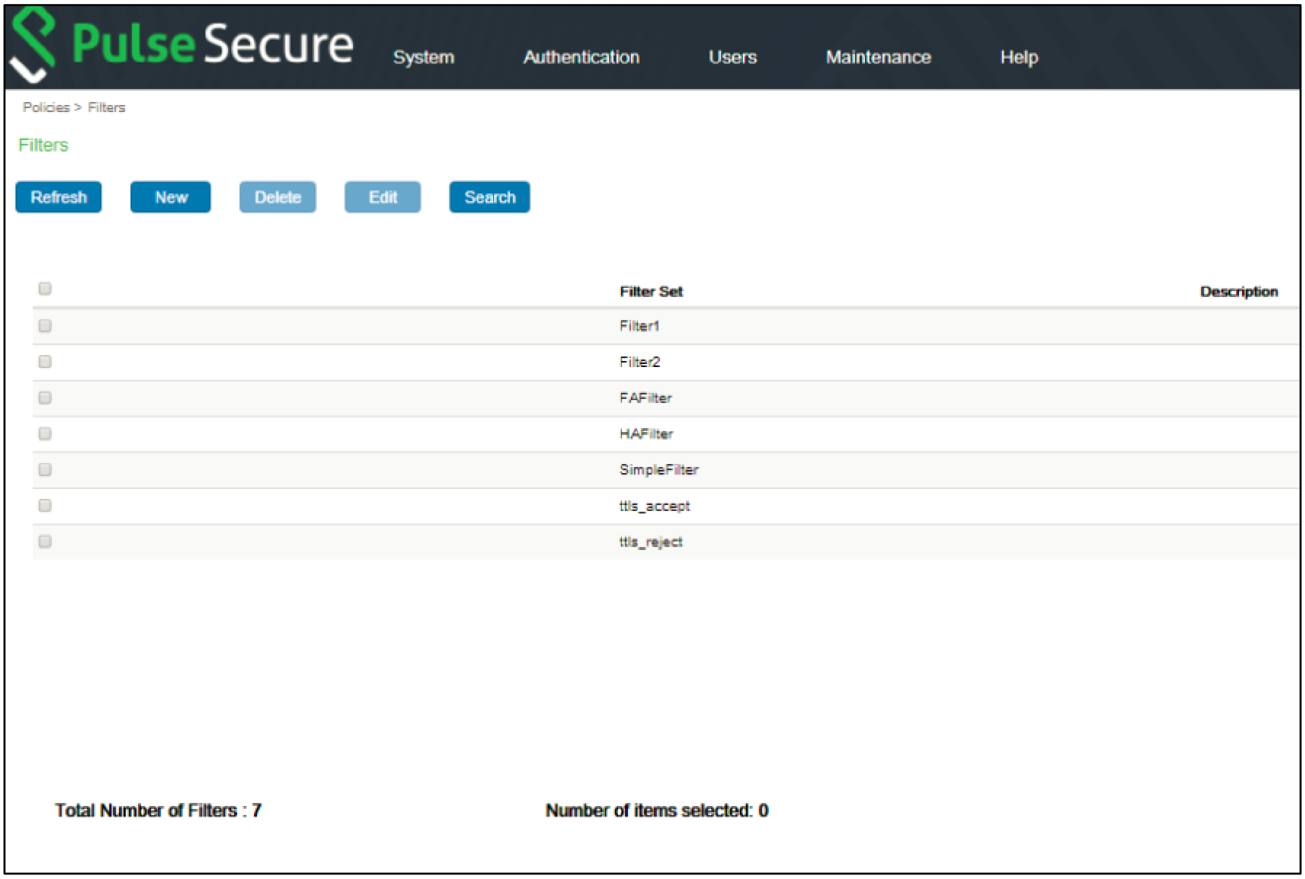
To reference filtering rules in proxy or directed realm configurations, you must use the FilterOut and FilterIn parameters in the [Auth] and [Acct] sections of a realm configuration file. For more information, refer to the Steel-Belted Radius Reference Guide.

Note: Do not allocate IP addresses from Steel-Belted Radius IP address pools in accounting filters. These addresses will be allocated but never released.

Filters Page

You can use the Filters page (Figure 122: Filters Page) to display the filters configured for Steel-Belted Radius. To open the Filters page, click Filters in the SBR Administrator sidebar.

Figure 122: Filters Page



□	Filter Set	Description
□	Filter1	
□	Filter2	
□	FAFilter	
□	HAFilter	
□	SimpleFilter	
□	ttls_accept	
□	ttls_reject	

Total Number of Filters : 7

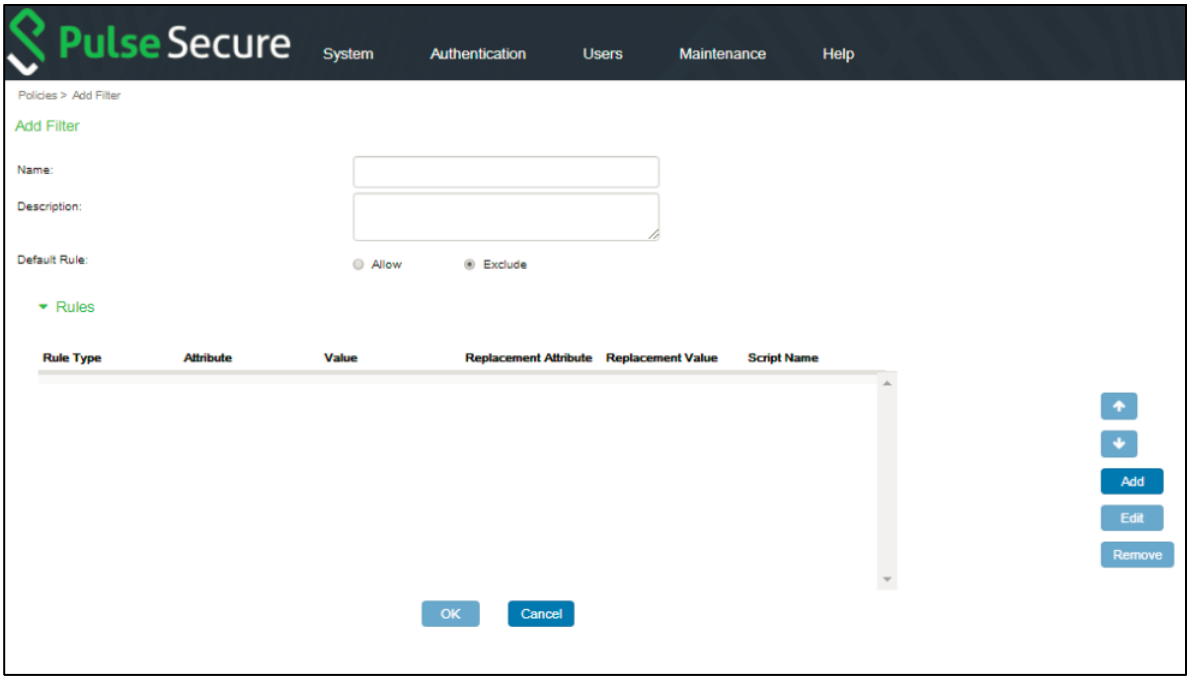
Number of items selected: 0

Adding a Filter

To add a filter:

1. Choose System > Policies > Filters in the menu bar.
2. Click New to display the Add Filter page (Figure 123: Add Filter Page).

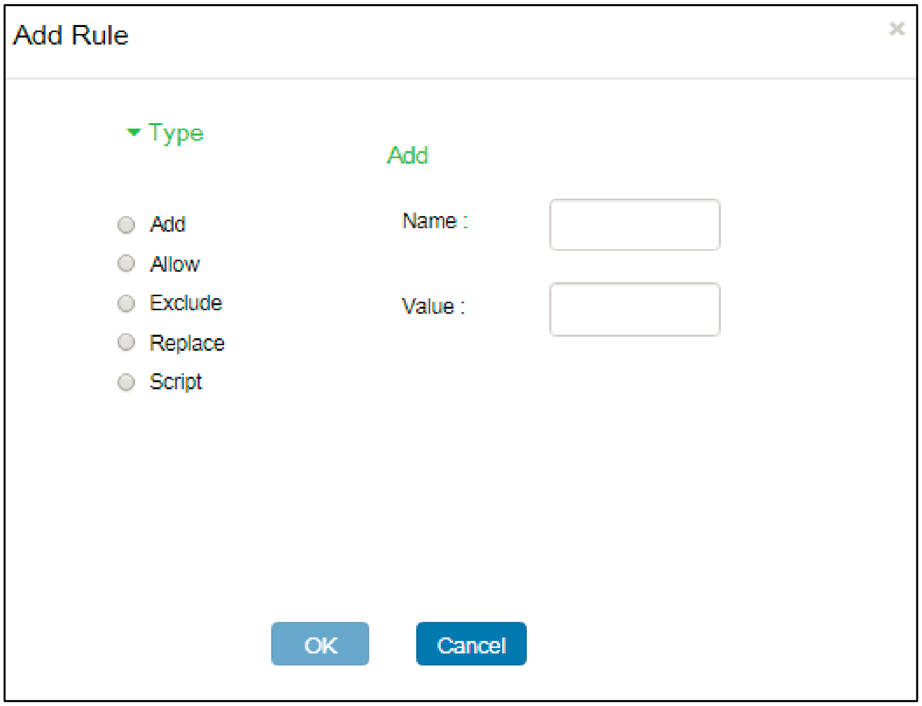
Figure 123: Add Filter Page



3. Enter the name of the filter in the **Name** field.
4. Optionally, enter a description of the filter in the **Description** field.

   You can enter as many as 4095 characters for a filter description.
5. Use the **Default Rule** radio buttons to specify whether attributes should be allowed or excluded if no other rule applies to a RADIUS packet.
6. Click the **Add** button to display the Add Rule page.

Figure 124: Add Rule Page



7. Select the type of rule you want to add to the filter.

   Options are Add, Allow, Exclude, Replace, and Script. The fields in the Add Rule page depend on the option you select.
8. Specify the attribute name and value settings you want to use for the rule.

   (GEE): Ifyou edit the filters for a Steel-Belted Radius server, you can apply your configuration changes without stopping the server:

   - Linux: Issue the HUP signal to the Steel-Belted Radius process: kill -HUP ProcessID
   - Windows: Run the radhup.exe program from the command shell. (radhup.exe is located in the server directory that you specified at installation time, usually C:\Radius\Service.)

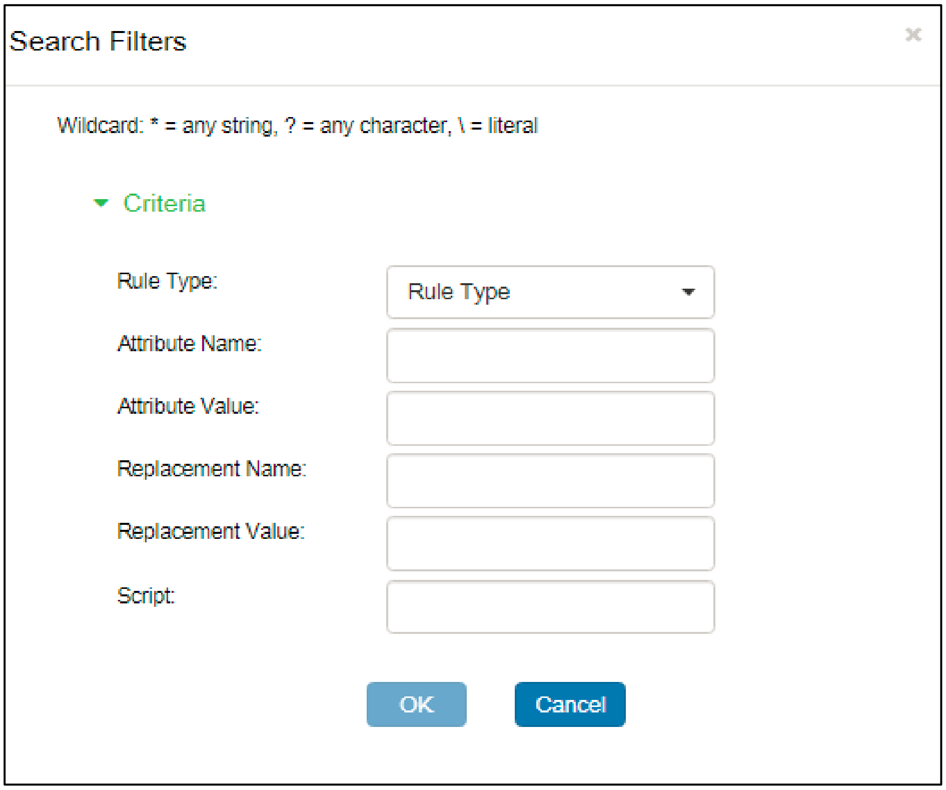
Searching the Filter List

You can search your list of filters to identify those of a specific type or that use a specific rule. To search your filter list:

1. Open the Filters page.
2. Click the **Search** button to go Search Filters page.
3. On the click of **Search** button, Search Filters Modal Window appears (Figure 125: Search Filters Modal Window) appears, enter one or more search criteria in the fields provided.

   You can use the question mark (?) wildcard to represent one character in a string. You can use the asterisk (*) wildcard to represent any number of characters. For example, entering the search string MS-MPPE-* would identify any filter that looks for the MS-MPPE-Recv-Key or MS-MPPE-Send-Key attribute.

Figure 125: Search Filters Modal Window



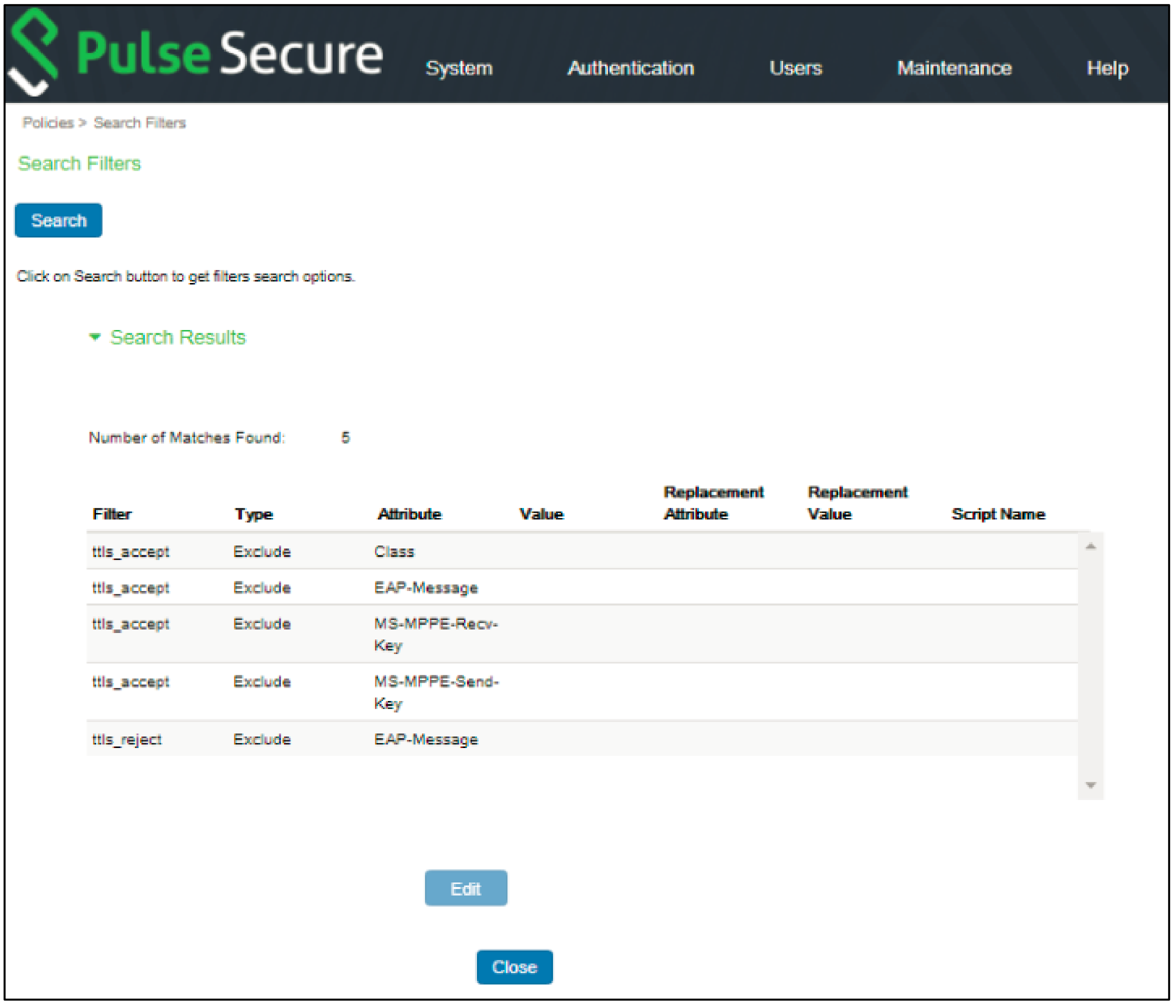
Search Filters

Wildcard: * = any string, ? = any character, \ = literal

4. Click **OK.**

   The Search Results page (Figure 126: Search Results Page) displays the list of filters that satisfy your search criteria.

Figure 126: Search Results Page



Click on Search button to get filters search options.

Number of Matches Found: 5

Filter	Type	Attribute	Value	Replacement Attribute	Replacement Value	Script Name
ttls_accept	Exclude	Class				
ttls_accept	Exclude	EAP-Message				
ttls_accept	Exclude	MS-MPPE-Recv-Key				
ttls_accept	Exclude	MS-MPPE-Send-Key				
ttls_reject	Exclude	EAP-Message				

Chapter 24

Setting Up EAP Authentication Policies via Legacy SBR Administrator

This chapter presents an overview of concepts relating to the Extensible Authentication Protocol (EAP) and describes how to configure Steel-Belted Radius to use EAP authentication methods and plug-ins via legacy SBR administrator.

About the Extensible Authentication Protocol

Steel-Belted Radius supports the Extensible Authentication Protocol (EAP), a standard for communication between network access devices and servers that provides for the future extensibility of authentication protocols.

EAP allows specialized knowledge about authentication protocols to be taken out of a NAD so that it acts solely as a conduit between authentication server and client. This means that new types of authentication can be supported by adding the appropriate functionality to server and client, without any changes to PPP or network access devices. When the authentication process is complete, the RADIUS server simply informs the NAD of the result.

Steel-Belted Radius supports several EAP authentication mechanisms, such as TTLS, TLS, PEAP, LEAP, MD5-Challenge, and Generic Token. Support for EAP has been designed to anticipate other authentication types as they become available.

TLS v1.2 (latest version) protocol provides improved flexibility and enhanced security. It supports modern encryption algorithms like SHA-256, AES cipher suites to communicate with RADIUS clients and avoid any weak cipher suite negotiations. OpenSSL 1.0.2d is used to support TLS v1.2 and address various security vulnerabilities.

The [EapSettings] section of radius.ini initialization file contains two parameters: AllowTLSFallback and MinimumProtocolVersion.

- Parameter AllowTLSFallback enables fallback to support SSL/TLS protocol versions
- Parameter MinimumProtocolVersion specifies the protocol version (TLSv10/TLSv11/TLSv12) to be used for EAP

Note: AllowTLSFallback option in radius.ini is applicable only for certificate based EAP plugins. The TLS fallback is a unique feature, aimed to provide backward compatibility to deprecated EAP client.

When AllowTLSFallback option is enabled in radius.ini, SBR EAP component will be initialized with protocol independent SSL structure. Depending on EAP client’s hello message, SBR will either negotiate in TLS v1.2 or it can degrade itself till SSL v3 compliant server. This option is very specific to EAP plugins only.

For technical details about EAP, see RFC 2284, “PPP Extensible Authentication Protocol (EAP),” and RFC 2869, “RADIUS Extensions.”

Handling EAP Requests

The flow of RADIUS packets in an EAP scenario is quite different from the transactions using standard user credentials (for example, PAP or CHAP). Standard user credentials involve the transmission of a RADIUS request from the NAD to Steel-Belted Radius and a response (either an Accept or Reject) from the server back to the NAD.

With EAP, the first packet sent from the NAD to Steel-Belted Radius contains an EAP-Message attribute containing an EAP Identity Response. This is a signal sent by the system being authenticated that it wants to be authenticated by means of EAP. It is now up to Steel-Belted Radius to select the EAP protocol with which it is to authenticate the end-user.

The contents of the User-Name attribute is the only guideline available to Steel-Belted Radius in selecting the appropriate EAP protocol. Should Steel-Belted Radius select an EAP protocol that is not supported by the client, the client has the opportunity to send an EAP-NAK and to request a specific alternate protocol.

ℹ Note: Given this general flow, a RADIUS request with EAP credentials must incur a minimum of two network round-trips between the RAS (or Access Point) and the Steel-Belted Radius before reaching a successful conclusion.

Automatic EAP Helpers

Automatic EAP helpers serve as intermediaries between EAP and traditional authentication methods. These helper modules can be configured (using an associated .eap file) to work with existing authentication methods to shield the authentication methods from the particulars of the selected EAP protocol.

Table 30 indicates whether each EAP type is implemented as an EAP helper or stand-alone module in Steel-Belted Radius.

Table 30: EAP Implementations

EAP-Type	Implemented As
EAP-TTLS	Standalone Authentication Method Module
EAP-TLS	Standalone Authentication Method Module
EAP-TLS	Automatic EAP helper
LEAP	Automatic EAP helper for MS-CHAP-v1
EAP Generic-Token	Standalone Authentication Method Module (SecurID)
EAP MD5-Challenge	Automatic EAP helper for CHAP
EAP MS-CHAP-v2	Automatic EAP helper for MS-CHAP-v2 (needed for PEAP)

Whether an automatic EAP helper can be used in conjunction with a specific authentication method depends on what types of credentials the authentication method supports.

The automatic EAP helper that implements EAP MD5-Challenge generates CHAP credentials, while the helper that implements LEAP generates MS-CHAP-v1 credentials. As such, EAP MD5-Challenge can be used only with authentication methods that support CHAP, and LEAP can be used only with authentication methods that support MS-CHAP-v1.

Table 31 summarizes the support for MS-CHAP-v1 and CHAP in the Steel-Belted Radius authentication methods.

Table 31: MS-CHAP-v1 and CHAP Support

Authentication Method	MS-CHAP-V1	CHAP
LDAP	Yes for BindName (password must be stored in the clear or encrypted using enc-md5 in LDAP server), No for Bind	Yes for BindName (password must be stored in the clear or encrypted using enc-md5 in LDAP server), No for Bind
Local	Yes	Yes
Proxy RADIUS	Yes	Yes
SecurID	No	No
SQL	Yes if password is in clear or encrypted using enc-md5 in SQL database	Yes if password is in clear or encrypted using enc-md5 in SQL database
TACACS+	No	Yes
UNIX User	No	No
UNIX Group	No	No
Windows Domain User	Yes (server must be running under SYSTEM account)	No
Windows Domain Group	Yes (server must be running under SYSTEM account)	No

Authentication Request Routing

The order in which authentication methods and automatic EAP helpers are called to handle an authentication request depends on two factors:

- The ordered list of enabled authentication methods (viewable in the Authentication Policies panel in SBR Administrator). Refer to "Activating EAP Methods" for information on using the Authentication Policies panel.

- The EAP-related configuration for each of the enabled authentication methods in the eap.ini file, which you configure from the Authentication Policies panel.

When Steel-Belted Radius receives an authentication request that does not contain EAP credentials, it passes the request to each enabled authentication method until one of the methods claims the request. The EAP settings in the eap.ini file come into play only when a request with EAP credentials is received. An authentication request contains EAP credentials if it includes one or more EAP-Message attributes and contains no other form of user credentials (for example, User-Password).

EAP-Only Setting

When an authentication method's EAP-Only setting is 1, Steel-Belted Radius prevents the authentication method from being called for any request that does not contain EAP credentials. Under this setting, the authentication method is also bypassed if an authentication request specifically requests an EAP protocol that is not listed in the authentication method's EAP-Type list in the eap.ini file.

ℹ Note: The PEAP authentication method plug-in converts the inner EAP/Generic Token credentials to PAP for security reasons. If you are using SecurID with PEAP, you should set the EAP-Only setting to 0.

First-Handle-Via-Auto-EAP Setting

If your configuration involves clients using more than one EAP protocol, Steel-Belted Radius must select an initial EAP protocol with which to proceed when receiving an authentication request with EAP credentials.

Selecting the incorrect EAP protocol is not fatal; the client simply sends an EAP NAK in response to the server's selected protocol and suggests an alternate one. After one additional network round-trip, the correct EAP protocol becomes active.

Depending on the capabilities of the authentication methods being used, you might be able to cut out this additional network round-trip that affects a portion of your EAP-based authentication requests.

If an authentication method can check for the existence of a user and can retrieve the user's password information with only the information available in the authentication request (for example, the username), it is said to be prefetch-capable. A prefetch-capable authentication method could be consulted first to see if a user exists in its database before committing to a specific EAP protocol.

If your authentication method is prefetch-capable, you would set First-Handle-Via-Auto-EAP to 0, indicating that the authentication method should have the first chance to handle the request. You would also set First-Handle-Via-Auto-EAP to 0 if the authentication method is capable of handling EAP credentials all on its own (clearly, it would not expect an automatic helper EAP method to do work on its behalf in this case).

By configuring the authentication method to be called first, Steel-Belted Radius can delay selection of an EAP protocol until it has ascertained whether the user exists in a particular authentication method's database. This is a useful technique when you plan to use more than one EAP protocol, but you do not know which one the client will want. Even in this scenario, automatic EAP helpers can still end up performing the EAP protocol processing; they will take over after the authentication method has retrieved a user's password information, rather than before.

The goal of an automatic EAP helper is to generate credentials against which traditional authentication methods (ones that do not understand EAP) can operate. Once an automatic EAP helper has generated these credentials, the authentication method that triggered the use of the helper is checked first for a password/credential match. Should this match not be present, the same traditional credentials are passed to all remaining enabled authentication methods in the master list (in the order in which they appear in the list).

Table 32: Authentication Method Prefetch Capability

Authentication Method	Prefetch Capable?
LDAP	Yes, if using BindName (rather than the Bind option)
Native User	Yes
SQL	Yes, if password does not need to be used as an input parameter in the SQL statement
UNIX User	No
Windows Domain	No

Note: If you enable the lockout facility in Steel-Belted Radius and you use a tunneled authentication method (TTLS or PEAP) with a prefetch-capable method (native user, SQL, or LDAP) and an enabled EAP protocol (MS-CHAPv2, MD5-Challenge, LEAP, TLS), then you must enable First Handle via Auto- EAP in that prefetch-capable method to prevent the outer username (anonymous) from being added to the lockout list.

Otherwise, when Steel-Belted Radius receives an authentication request that uses an unconfigured EAP method, Steel-Belted Radius will reject the user (because the EAP method is not configured) and add the outer username (anonymous) to its lockout list. This will result in all users with an outer authentication name of anonymous being rejected until the lockout period expires.

EAP-NAK Notifications

If you are supporting only one type of client or only one EAP protocol, Steel-Belted Radius selects that EAP protocol for all EAP-based authentication requests it receives. If you are planning to support multiple EAP protocols and do not intend to maintain databases that track the appropriate EAP protocol on a user-by-user basis, Steel-Belted Radius automatically selects the appropriate EAP protocol for you.

When multiple EAP protocols are in play, you should configure each authentication method you plan to use with all the EAP protocols that can be used with it. In this configuration, when Steel-Belted Radius receives an authentication request containing EAP information, it chooses the first EAP protocol listed for the first authentication method that claims the request. Should the client require a different EAP protocol, it sends back an EAP-NAK that specifies the EAP protocol it would prefer to use.

After receiving an EAP-NAK, Steel-Belted Radius performs a scan of the authentication methods to find the first authentication method that has the requested EAP protocol listed (the authentication method might support this EAP protocol directly or with the help of an automatic EAP helper).

If the requested EAP protocol does not appear in any of the authentication methods' lists of supported EAP protocols, Steel-Belted Radius rejects the authentication request.

Reauthenticating Connections

Most Access Points understand only a limited number of attributes that can be included in a RADIUS response to signal that the user has been accepted. The Session-Timeout attribute is of particular significance in a WLAN realm as it instructs the Access Point how long to allow the user to remain connected to a WLAN before having to re-authenticate to Steel-Belted Radius.

You can configure your choice of Session-Timeout settings using standard Steel-Belted Radius reply-list items on a user-by-user basis. If you are using EAP-TLS or EAP-TTLS to authenticate users, you can also have these modules automatically generate Session-Timeout attributes based on policies set in their configuration files. This level of control is necessary for EAP-TLS and EAP-TTLS as these modules also support session resumption, a quicker method of re-authenticating users. The value in the Session-Timeout attribute might need to be dynamically calculated in these cases.

Note: Not all Access Points support the Session-Timeout attribute. You should check your Access Points' specifications to determine whether this configuration must be performed in a fixed manner on the Access Point or if the Access Point should defer to the server.

Certificates

A certificate is an electronic data structure used to identify an individual, a server, a company, or some other entity, and to associate that identity with a public key and an associated private key. Like a passport, a certificate provides generally recognized proof of an entity's identity. Certificates bind public key values to entities, so that remote users of an entity's public key can be certain the associated private key is owned by the correct person or system. Certificates help prevent the use of fake public keys for impersonation. Only the public key certified by the certificate will work with the corresponding private key possessed by the entity identified by the certificate. The most widely accepted format for certificates is defined by the ITU-T X.509 international standard, which is described in RFC 3280, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile."

Certificate authorities (CAs) are entities that validate identities and issue certificates. An organization that wants to serve as its own CA can issue its own certificates, or an organization can purchase certificates from a trusted third-party CA. The methods used to validate an identity vary depending on the policies of a given CA. In general, before issuing a certificate, a CA must verify the identity of the entity and must digitally sign the certificate to ensure it cannot be modified. This ensures that a certificate issued by a CA binds a particular public key to the name of the entity the certificate identifies (such as the name of an employee).

In addition to a public key, a certificate includes the name of the entity it identifies, an expiration date, the name and URI of the CA that issued the certificate, a serial number, and the digital signature of the issuing CA, which creates a mathematical relationship between the signing CA certificate's public key and the public key of the certificate it signs. The CA's digital signature allows the certificate to function as a "letter of introduction" for users who know and trust the CA but don't know the entity identified by the certificate.

Because a certificate's expiration date is part of its signed contents, remote entities can verify that a certificate is valid and current.

Common types of certificates include the following:

- Certificate Authority certificates can sign other certificates.
- Server certificates are used on a server to enable a software client to verify the validity of the connection to a machine ("Am I really connecting to www.pulsesecure.net?") and to create an encrypted channel between a client and a server.
- Client certificates are used to allow a server to verify a client's identity (certificate based authentication) and to allow a user to digitally sign or encrypt data. Client certificates, which is digitally signed by a trusted certificate authority, is stronger proof of a client's identity than username/ password credentials alone.

Certificate Chains

A certificate chain is a sequence of certificates, where each certificate in the chain is signed by the certificate above it in the chain. At the top of the chain is a self-signed certificate. Each CA in the chain vouches for the identity in the entity to which it issues a signed digital certificate. Certificate chains establish a chain of trust; if you trust the CA at the top of the chain, this implies you can trust the signed certificates below it in the chain.

Certificate Revocation Lists

Under normal circumstances, a certificate remains valid until it reaches its expiration date. However, a certificate might become invalid before it expires. For example, if an employee whose identity is bound to a certificate terminates employment or if an enterprise suspects the confidentiality of the private key associated with a certificate's public key has been compromised, the certificate might be declared invalid and revoked before its expiration date.

When a CA revokes a certificate, it must let other entities know the certificate is no longer valid and should not be accepted. A certificate revocation list (CRL) is a signed data structure that identifies the serial numbers of certificates that have been issued and subsequently revoked by the CA. When a remote entity is asked to use a certificate to verify a remote user's identity, it can download a current copy of the applicable CRL from a CRL distribution point (CDP) and confirm that the certificate's serial number is not present. If a CRL has expired, the entity must connect to the CDP to download a new revocation list.

CRLs can be issued by a CA periodically (hourly, daily, or weekly) or as needed. When a certificate is revoked, its serial number is listed in the CRL, and that serial number remains in the CRL at least one period after the certificate's expiration date. CRLs, like certificates, can be distributed by untrusted servers and untrusted communications.

Under some circumstances, latency (the time between when a certificate is revoked and when the certificate's serial number appears on the CRL of the issuing CA) might be a concern. For example, if a revocation is reported today, that revocation will not be reliably notified to certificate-using systems until all currently issued CRLs are updated, which might take hours, days, or even weeks. Online revocation checking can reduce the latency between a revocation report and the distribution of the information to relying parties.

If CRL checking is enabled, Steel-Belted Radius uses the URI information contained in a client certificate to connect to the certificate's CDP. Steel-Belted Radius then uses HTTP, LDAP, or a network file system to retrieve the appropriate CRLs. Steel-Belted Radius stores these retrieved CRLs in the CRLCache directory under the radiusdir server directory.

When a client certificate is presented during EAP-TLS or EAP-TTLS authentication, Steel-Belted Radius can evaluate the client's certificate chain against its set of stored CRLs to verify none of the certificates in the chain have been revoked.

You can configure the following settings for CRL checking:

- Static CDPs—A static CDP is a CDP whose address (URI) is specified in the [Static_CDPs] section of a TLS or TTLS initialization file.
- CRL expiration—The CRL checking feature can be configured to operate in strict or lax mode.
    - In strict mode, a cached CRL that has expired will be immediately discarded; if Steel-Belted Radius cannot acquire a new CRL in the allotted time during a CRL check on a chain, the user is rejected.
    - In lax mode, you can configure Steel-Belted Radius to accept an expired CRL for a a period past its expiration. Note that Steel-Belted Radius attempts to obtain a current CRL whether it is running in strict or lax mode.
- Missing CDP attribute—When a CRL check is performed on a certificate chain, Steel-Belted Radius reads the contents of the CDP attribute for each certificate past the root certificate and uses the CDP information to retrieve the appropriate CRL. If a non-root certificate in the chain does not contain a CDP attribute, no CRL checking will be performed for that certificate. You can configure EAP-TLS to reject the user if it encounters a non-root certificate that is missing a CDP attribute.
- Incomplete LDAP CDP—Some CAs can create certificates that contain an LDAP-style CDP (//ldap:\\...) that does not specify the identity of the LDAP server to be queried. You can designate a default LDAP server that will be used when such CDPs are encountered. If you do not designate a default LDAP server and an LDAP-style CDP is encountered, the CRL retrieval will fail.

- HTTP proxies for CRL checking—Network security policy may prevent Steel-Belted Radius from making a direct HTTP connection to a CDP. In such cases, you can configure Steel-Belted Radius to download CRLs through an HTTP proxy server on its local network. Optionally, you can specify the hosts or domains that do not require an HTTP proxy.
- CRL cache flushing—You can flush the CRL caches used for EAP-TLS and EAP-TTLS authentication at any time.

EAP-TLS

The EAP-TLS (Transport Layer Security) protocol requires that both user and authentication server have certificates for mutual authentication. While the mechanism is very strong, it requires that the corporation that deploys it maintain a certificate infrastructure for all of its users.

EAP-TLS can be deployed as an authentication method or as an automatic EAP helper.

- When EAP-TLS is deployed as an authentication method, EAP-TLS appears in the Authentication Policies panel in SBR Administrator after it is deployed as an authentication method. You can use the Authentication Policies panel to enable the EAP-TLS method and specify its sequence relative to other authentication methods Steel-Belted Radius uses.

  When EAP-TLS is deployed as an authentication method, you can configure it to perform certificate revocation list (CRL) checking. When CRL checking is enabled, EAP-TLS confirms that the client’s certificate chain traces back to one of the trusted root certificates installed at initialization and checks the serial number of each certificate in the chain against the contents of CRLs to verify that none of the certificates in the chain have been revoked.

  You can configure the tlsauth.aut file to call a fixed profile when TLS-EAP is used. This profile specifies the attributes that are sent back in response to a successful authentication.

  You cannot use secondary authorization when EAP-TLS is deployed as an authentication method.

- When EAP-TLS is deployed as an automatic EAP helper, you must list TLS in the EAP-Type list of an authentication method. When EAP-TLS is triggered, the tlsauth authentication goes through the TLS handshake required by the EAP-TLS specification. Assuming the user provides a certificate that the server can verify against a list of trusted root certificates, the EAP-TLS part of the exchange concludes successfully.

  You might not want to grant access to your network to every user with a trusted certificate. By enabling the optional secondary authorization feature of the tlsauth plugin, you can have Steel-Belted Radius authorize users with valid certificates on a case-by-case basis. Secondary authorization also allows you to include user-specific attributes in an Access-Accept response; these attributes can be used to communicate options that are to be active for a user’s connection to the NAD. Without secondary authorization, the only attributes returned on an Access-Accept are those generated by the tlsauth plug-in itself (termination-action and session-limit).

  If you enable the TLS authentication method, secondary authorizations must be performed by local authentication methods (they cannot be proxied). The authentication method you select for secondary authorizations must be able to authenticate users in a single pass; it cannot challenge the authorization request and request additional information. The username employed during secondary authorization is derived from a field in the user’s certificate. Since a user’s certificate does not include a password, you must configure tlsauth to make the secondary authorization request with no password or with a fixed password.

If you configure secondary authorization with no password, your selected authentication method must be capable of handling requests that do not include passwords; the only authentication methods that support this style of authentication and ship with Steel-Belted Radius are Native User, LDAP and SQL. If you configure secondary authorization with a fixed password, you can use any authentication method that supports PAP authentication. In this configuration all user records must have the same fixed password.

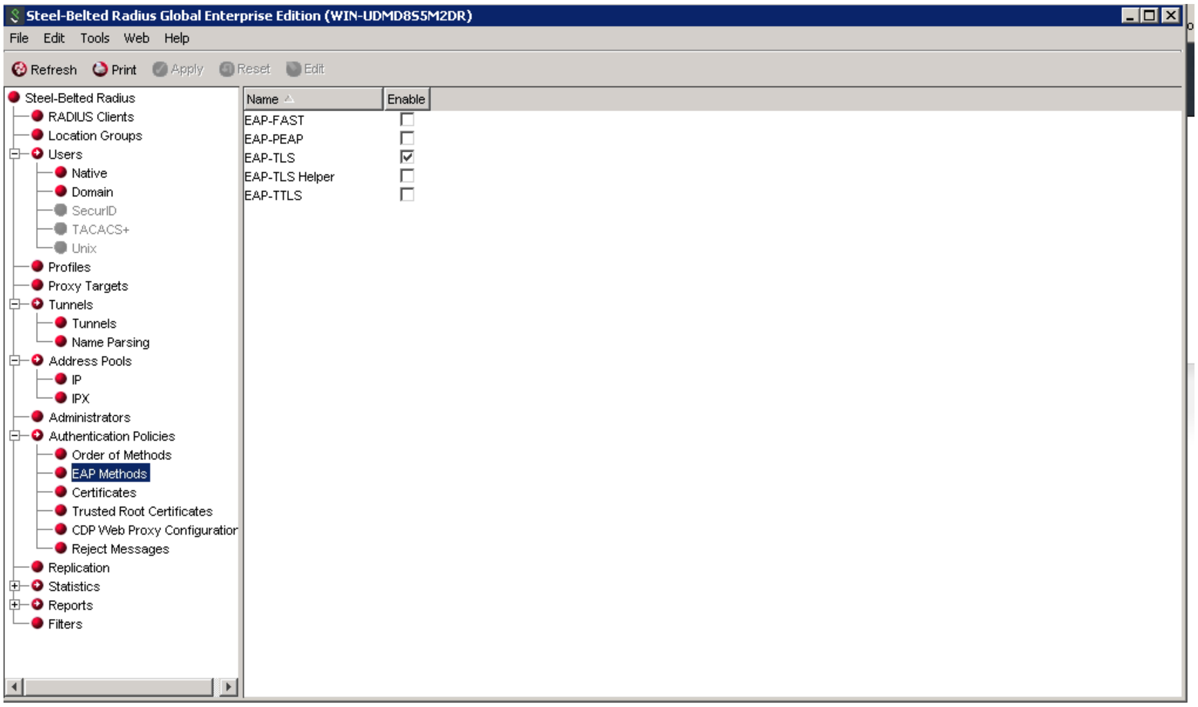
Configuring EAP-TLS as an Authentication Method

Note: You must configure the server certificate for the Steel-Belted Radius server before you use the EAP-TLS authentication method. For information on configuring your server certificate, see "Configuring Server Certificates".

To configure EAP-TLS as an authentication method:

1. Select Authentication Policies > EAP Methods to open the EAP Methods panel.

Figure 127: EAP Methods Panel



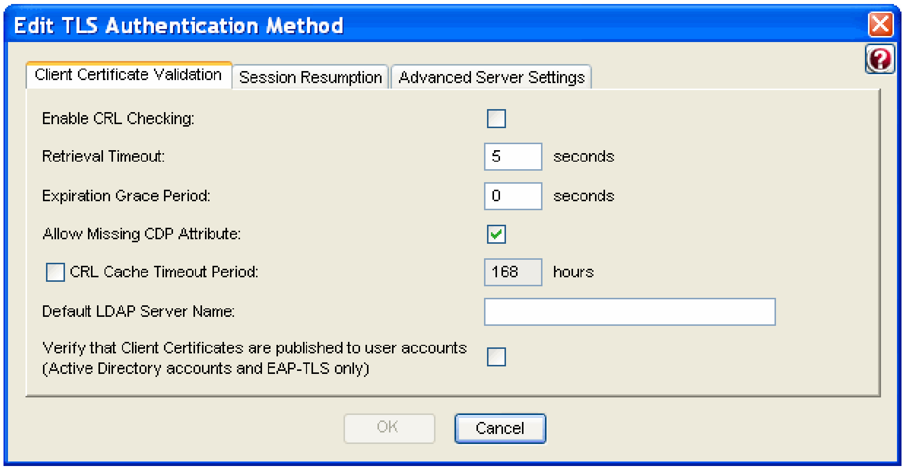
Name	Enable
EAP-FAST	□
EAP-PEAP	□
EAP-TLS	☑
EAP-TLS Helper	□
EAP-TTLS	□

2. Click the **Enable** check box for the EAP-TLS authentication method.

3. Select the **EAP-TLS** entry and click the **Edit** button on the toolbar (or double-click the **EAP-TLS** entry).

   The Edit TLS Authentication Method dialog opens.

Figure 128: Edit TLS Authentication Method dialog



Enable CRL Checking: ☐

Retrieval Timeout: 5 seconds

Expiration Grace Period: 0 seconds

Allow Missing CDP Attribute: ☑

☐ CRL Cache Timeout Period: 168 hours

Default LDAP Server Name:

Verify that Client Certificates are published to user accounts (Active Directory accounts and EAP-TLS only) ☐

4. Use the tabs in the Edit TLS Authentication Method dialog to configure the following settings:

   - Client certificate validation
   - Session resumption
   - Advanced server settings

   Each configuration task is described separately below.

Configuring Client Certificate Validation

Client certificate validation settings let you specify how Steel-Belted Radius performs certificate revocation list (CRL) checking.

To configure session resumption for the EAP-TLS protocol:

1. Click the **Client Certificate Validation** tab in the Edit TLS Authentication Method dialog.
2. Click the **Enable CRL Checking** check box to enable CRL checking.
3. Enter the number of seconds that EAP-TLS will wait for a CRL checking transaction to complete when the CRL check involves a CRL retrieval in the **Retrieval Timeout** field.

   When CRL retrieval takes longer than the specified time, the user's authentication request results in a reject.
4. Enter the number of seconds during which a CRL is still considered acceptable after it has expired in the **Expiration Grace Period** field.

   EAP-TLS always attempts to retrieve a new CRL when it is presented with a certificate chain and it finds an expired CRL in its cache.

   - If you enter 0 (strict expiration mode), EAP-TLS does not accept a CRL that has expired.
   - If you enter a value greater than 0 (lax expiration mode), EAP-TLS considers the expired CRL as an acceptable stand-in from the time the CRL expires to the time the grace period ends.
5. Click the Allow Missing CDP Attribute check box if you want Steel-Belted Radius to accept a non-root certificate that does not have a CDP attribute.

Without a CDP attribute, EAP-TLS will not know how to retrieve a CRL and will not be able to perform a revocation check on the certificate.

If you click the Allow Missing CDP Attribute check box, EAP-TLS allows such certificates and skips CRL checking for them.

If you clear the Allow Missing CDP Attribute check box, EAP-TLS does not accept a CRL with a missing CDP attribute.

6. If you want to specify a CRL cache timeout period, click the CRL Cache Timeout Period check box and enter the number of hours in the timeout period in the hours field.

   - If you do not enable this setting, the CRL will be refreshed whenever it expires.

   - If you enable this setting and enter 0, Steel-Belted Radius always regards the CRL in the cache as expired and downloads a new CRL every time it receives a client certificate request.

   - If you enable this setting and enter a number greater than 0, the CRL begins to expire when the age of the CRL in the cache exceeds the number of hours specified in this field or when the scheduled CRL expiration time occurs, whichever comes first.

   After a CRL has expired (because its scheduled expiration time has passed or because the CRL cache has timed out), Steel-Belted Radius uses the expiration grace period to determine whether it should use the current CRL.

7. Enter the name of the LDAP server to use if the CDP contains a value that begins with the string // ldap:\\\ in the Default LDAP Server Name field.

   CDPs generated by some CAs do not include the identity of the LDAP server. If you expect to encounter certificates with this style CDP, specify the name of the LDAP server that contains the CRLs.

   If you don’t specify a server name and such certificates are encountered, the CRL retrieval fails.

8. If your enterprise uses Active Directory and you want client certificates published to user accounts, click the Verify that Client Certificates are published to user accounts check box.

Configuring Session Resumption

Session resumption settings let you specify whether session resumption is permitted and under what circumstances session resumption is performed.

Note: For session resumption to work, the network access device must be configured to handle the Session-Timeout return list attribute, because the network access device must be able to tell the client to reauthenticate after the session timer has expired.

To configure session resumption for the EAP-TLS protocol:

1. Click the **Session Resumption** tab in the Edit TLS Authentication Method dialog.

2. Enter the maximum number of seconds you want the client to remain connected to the network access device before having to re-authenticate in the **Session TImeout** field.

   If you enter a number greater than 0, the lesser of this value and the remaining resumption limit is sent in a Session-Limit attribute to the RADIUS client on the RADIUS Access-Accept response.

If you enter 0, no Session-Limit attribute is generated. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

Entering a value such as 600 (10 minutes) does not necessarily cause a full re-authentication to occur every 10 minutes. You can configure the resumption limit to make most re-authentications fast and computationallycheap.

3. Enter the value that you want returned in a Termination-Action attribute in the Termination Action field.

   The Termination-Action attribute is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached. Valid values are:

   - -1: Do not send the attribute.
   - 0: Send the Termination-Action attribute with a value of 0.
   - 1: Send the Termination-Action attribute with a value of 1.

   Default value is -1. Note that this does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

4. Enter the maximum number of seconds you want the client to be able to re-authenticate using the TLS session resumption feature in the Resumption Limit field.

   This type of re-authentication is fast and computationally cheap. It does, however, depend on previous authentications and might not be considered as secure as a complete (computationally expensive) authentication. Specifying a value of 0 disables the session resumption feature.

ConfiguringAdvancedServerSettings

Advanced server settings specify the manner in which the inner authentication step operates. To configure advanced server settings for the EAP-TLS protocol:

1. Click the Advanced Server Settings tab in the Edit TLS Authentication Method dialog.

2. Enter the maximum length of the TLS message that might be generated during each iteration of the TLS exchange. in the **TLS Message Fragment Length** field.

   Enter a number in the range 500–4096. This value affects the number of RADIUS challenge/response round trips required to conclude the TLS exchange. A value of 1400 might result in 6 round-trips, while a value of 500 might result in 15 round trips.

   Some Access Points might have problems with RADIUS responses or EAP messages that exceed the size of one Ethernet frame (1500 bytes including IP/UDP headers).

   The default length for TLS messages is 1020 bytes, which prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame.

3. Enable the **Return MPPE Keys** check box to specify whether the TLS authentication method includes RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Access-Accept response sent to the Access Point.

   You should enable this option if the Access Point needs to key the WEP encryption. If the Access Point is authenticating only end-users and WEP is not being used, you can clear this check box.

4. Use the **DH Prime Bits** list to specify the number of bits in the prime number that the module uses for Diffie-Hellman exponentiation.

   Selecting a longer prime number makes the system less susceptible to certain types of attacks but requires more CPU processing to compute the Diffie-Hellman key agreement operation.

   Valid values are 512, 1024, 1536, 2048, 3072, and 4096 bits.

5. Enter the TLS cipher suites (in order of preference) that the server is to use in the **Cipher Suites** field.

   These cipher suites are documented in RFC 2246, "The TLS Protocol Version 1.2," and other TLS-related RFCs and draft RFCs.

   Default value is 0x3C, 0x3D, 0x67, 0x6B, 0x40, 0x6A, 0x9C, 0x9D, 0x9E, 0x9F, 0xA2, 0xA3.

6. Optionally, enable the **Verify User Name is Principal** check box if you want Steel-Belted Radius to verify that the contents of the RADIUS User-Name attribute match the Principal Name of the certificate used to authenticate the user.

   Certificates issued by Microsoft's Windows 2000 Certificate Server typically include a Subject Alternative Name/Other Name attribute, where Principal Name set to something like **user@certtest.acme.com.**

   The Windows XP client that supports EAP-TLS in conjunction with 802.1X extracts this attribute value from the client's certificate and uses it to respond to the Access Point's EAP Identity Request. The Access Point, in turn, packages up this value as the RADIUS User-Name attribute in requests it sends to a RADIUS server.

   You should clear (disable) the check box if the certificates used do not include a Principal Name or if the client being used does not report the contents of Principal Name as the user's identity in response to an EAP Identity Request.

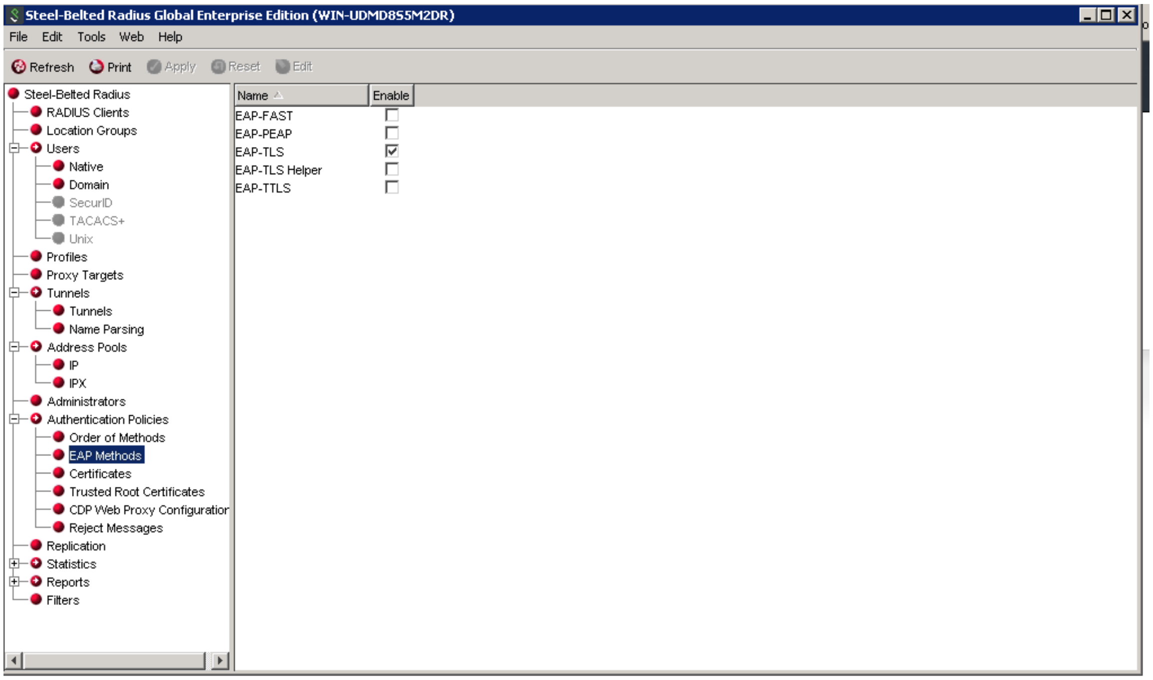
Configuring EAP-TLS as an Automatic EAP Helper

Note: You must configure the server certificate for the Steel-Belted Radius server before you use the TLS EAP helper. For information on configuring your server certificate, see "Configuring Server Certificates".

To configure EAP-TLS as an EAP helper:

1. Select Authentication Policies > EAP Methods to open the EAP Methods panel (Figure 129: EAP Methods Panel).

Figure 129: EAP Methods Panel



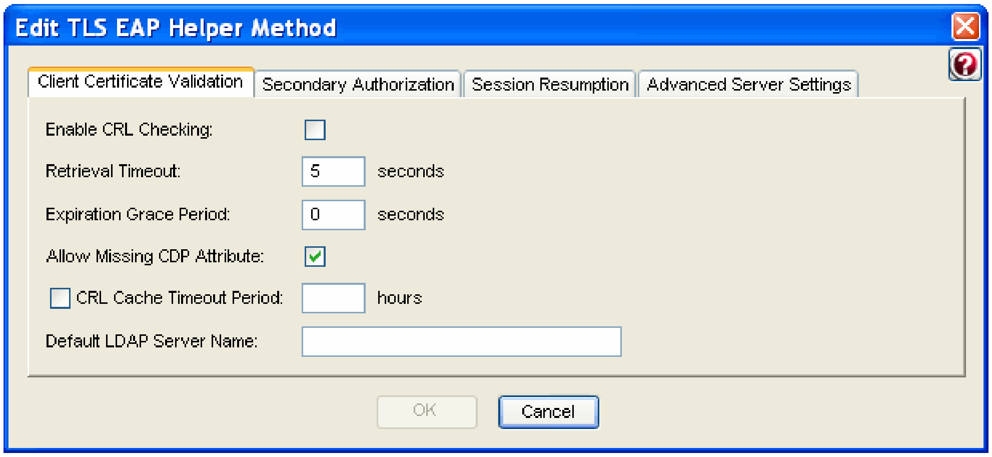
Name	Enable
EAP-FAST	☐
EAP-PEAP	☐
EAP-TLS	☑
EAP-TLS Helper	☐
EAP-TTLS	☐

2. Click the Enable check box for the EAP-TLS Helper method.

3. Select the EAP-TLS Helper entry and click the Edit button on the toolbar (or double-click the EAP-TLS Helper entry).

   The Edit TLS EAP Helper Method dialog opens.

Figure 130: Edit TLS EAP Helper Method dialog



Enable CRL Checking: ☐

Retrieval Timeout: 5 seconds

Expiration Grace Period: 0 seconds

Allow Missing CDP Attribute: ☑

☐ CRL Cache Timeout Period: hours

Default LDAP Server Name:

4. Use the tabs in the Edit TLS EAP Helper Method dialog to configure the following settings:

   - Client certificate validation
   - Secondary authorization
   - Session resumption
   - Advanced server settings

Each configuration task is described separately below.

Configuring Client Certificate Validation

Client certificate validation settings let you specify how Steel-Belted Radius performs certificate revocation list (CRL) checking.

To configure session resumption for the the TLS EAP helper protocol:

1. Click the **Client Certificate Validation** tab in the Edit TLS EAP Helper Method dialog.
2. Click the **Enable CRL Checking** check box to enable CRL checking.
3. Enter the number of seconds that the TLS EAP helper will wait for a CRL checking transaction to complete when the CRL check involves a CRL retrieval in the **Retrieval Timeout** field.

   When CRL retrieval takes longer than the specified time, the user's authentication request results in a reject.
4. Enter the number of seconds during which a CRL is still considered acceptable after it has expired in the **Expiration Grace Period** field.

   The TLS EAP helper always attempts to retrieve a new CRL when it is presented with a certificate chain and it finds an expired CRL in its cache.

   - If you enter 0 (strict expiration mode), the TLS EAP helper does not accept a CRL that has expired.
   - If you enter a value greater than 0 (lax expiration mode), the TLS EAP helper considers the expired CRL as an acceptable stand-in from the time the CRL expires to the time the grace period ends.
5. Click the **Allow Missing CDP Attribute** check box if you want Steel-Belted Radius to accept a non-root certificate that does not have a CDP attribute.

   Without a CDP attribute, the TLS EAP helper will not know how to retrieve a CRL and will not be able to perform a revocation check on the certificate.

   If you click the **Allow Missing CDP Attribute** check box, the TLS EAP helper allows such certificates and skips CRL checking for them.

   If you clear the **Allow Missing CDP Attribute** check box, the TLS EAP helper does not accept a CRL with a missing CDP attribute.
6. If you want to specify a CRL cache timeout period, click the **CRL Cache Timeout Period** check box and enter the number of hours in the timeout period in the **hours** field.

   - If you do not enable this setting, the CRL will be refreshed whenever it expires.
   - If you enable this setting and enter 0, Steel-Belted Radius always regards the CRL in the cache as expired and downloads a new CRL every time it receives a client certificate request.
   - If you enable this setting and enter a number greater than 0, the CRL begins to expire when the age of the CRL in the cache exceeds the number of hours specified in this field or when the scheduled CRL expiration time occurs, whichever comes first.

After a CRL has expired (because its scheduled expiration time has passed or because the CRL cache has timed out), Steel-Belted Radius uses the expiration grace period to determine whether it should use the current CRL.

7. Enter the name of the LDAP server to use if the CDP contains a value that begins with the string //ldap:\\\ in the **Default LDAP Server Name** field.

   CDPs generated by some CAs do not include the identity of the LDAP server. If you expect to encounter certificates with this style CDP, specify the name of the LDAP server that contains the CRLs.

   If you don’t specify a server name and such certificates are encountered, the CRL retrieval fails.

Configuring Secondary Authentication

Secondary authorization settings let you specify whether secondary authorization is performed and, if it is, what information is used in the secondary authorization request.

To configure session resumption for the TLS EAP helper protocol:

1. Click the **Secondary Authorization** tab in the Edit TLS EAP Helper Method dialog.
2. Click the **Enable Secondary Authorization** check box to enable secondary authorization checking.

   If secondary authorization is disabled, the EAP-TLS plug-in accepts the user upon proof of ownership of a private key that matches a valid certificate.

   If secondary authorization is enabled, a secondary authorization check against a traditional authentication method such as an SQL plug-in is performed.

3. Specify whether you want user names to be converted to Subject CN names or principal names.

   After the EAP-TLS module has concluded its processing, it might still defer to a traditional authentication method (core or plug-in) for final authorization. To do so, it must provide a user name and password to the traditional authentication method.

   - If you click the **Subject CN** option button, the EAP-TLS module parses the Subject attribute of the client’s certificate for the least significant ‘CN=’ and takes the value of this attribute (for example, ‘George Washington’) as the user name being passed to the traditional authentication method.
   - If you click the **Principal Name** option button, the EAP-TLS module uses the principal name (Subject Alternate Name or Other Name) from the client certificate (for example, joe@acme.com) as the user name being passed to the traditional authentication method.

4. If you plan to use secondary authorization against an authentication method (for example, LDAP) that cannot be configured to ignore the lack of user credentials, specify a fixed password that the plug-in uses on all secondary authorization checks in the **Fixed Password** field.

   By default, the secondary authorization check includes a user name but no other user credentials, because no password or similar credential for the client is available at the conclusion of the TLS handshake. Some authentication methods (Native User, LDAP, and SQL) can be configured to not require user credentials.

5. If you want the EAP-TLS plug-in to add four attributes to the request before the secondary authorization check is performed, click the Include Certificate Info check box.

   When the Include Certificate Info check box is clicked, Steel-Belted Radius adds the following attributes to the request:

   - The Funk-Peer-Cert-Subject attribute contains the value of the Subject attribute in the client certificate.
   - The Funk-Peer-Cert-Principal attribute contains the value of the principal name (Subject Alternate Name or Other Name) attribute of the client certificate.
   - The Funk-Peer-Cert-Issuer attribute contains the value of the Issuer attribute in the client certificate.
   - The Funk-Peer-Cert-Hash attribute contains a hexadecimal ASCII representation of the SHA1 hash of the client certificate.

   These attributes are ignored if the authentication method that performs the authentication check does not use them.

Configuring Session Resumption

Session resumption settings let you specify whether session resumption is permitted and under what circumstances session resumption is performed.

Note: For session resumption to work, the network access device must be configured to handle the Session-Timeout return list attribute, because the network access device must be able to tell the client to reauthenticate after the session timer has expired.

To configure session resumption for the TLS EAP helper protocol:

1. Click the **Session Resumption** tab in the Edit TLS EAP Helper Method dialog.
2. Enter the maximum number of seconds you want the client to remain connected to the network access device before having to re-authenticate in the **Session TImeout** field.

   If you enter a number greater than 0, the lesser of this value and the remaining resumption limit is sent in a Session-Limit attribute to the RADIUS client on the RADIUS Access-Accept response.

   If you enter 0, no Session-Limit attribute is generated. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

   Entering a value such as 600 (10 minutes) does not necessarily cause a full re-authentication to occur every 10 minutes. You can configure the resumption limit to make most re-authentications fast and computationallycheap.

3. Enter the value that you want returned in a Termination-Action attribute in the Termination Action field.

   The Termination-Action attribute is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached. Valid values are:

   - -1: Do not send the attribute.

- 0: Send the Termination-Action attribute with a value of 0.
- 1: Send the Termination-Action attribute with a value of 1.

Default value is -1. Note that this does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

4. Enter the maximum number of seconds you want the client to be able to re-authenticate using the TLS session resumption feature in the **Resumption Limit** field.

   This type of re-authentication is fast and computationally cheap. It does, however, depend on previous authentications and may not be considered as secure as a complete (computationally expensive) authentication. Specifying a value of 0 disables the session resumption feature.

Configuring Advanced Server Settings

Advanced server settings specify the manner in which the inner authentication step operates. To configure advanced server settings for the TLS EAP helper protocol:

1. Click the **Advanced Server Settings** tab in the Edit TLS EAP Helper Method dialog.
2. Enter the maximum length of the TLS message that may be generated during each iteration of the TLS exchange in the **TLS Message Fragment Length** field.

   Enter a number in the range 500–4096. This value affects the number of RADIUS challenge/response round-trips required to conclude the TLS exchange. A value of 1400 may result in 6 round-trips, while a value of 500 may result in 15 round-trips.

   Some Access Points may have problems with RADIUS responses or EAP messages that exceed the size of one Ethernet frame (1500 bytes including IP/UDP headers).

   The default length for TLS messages is 1020 bytes, which prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame.
3. Enter the maximum number of seconds you want to allow for the EAP authentication sequence in the **Max Transaction Time** field.

   If the EAP authentication sequence takes longer than the number of seconds specified in this field, Steel-Belted Radius terminates the user authentication.
4. Enable the **Return MPPE Keys** check box to specify whether the TLS EAP helper includes RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Access-Accept response sent to the Access Point.

   You should enable this option if the Access Point needs to key the WEP encryption. If the Access Point is authenticating only end-users and WEP is not being used, you can clear this check box.
5. Use the **DH Prime Bits** list to specify the number of bits in the prime number that the module uses for Diffie-Hellman exponentiation.

   Selecting a longer prime number makes the system less susceptible to certain types of attacks but requires more CPU processing to compute the Diffie-Hellman key agreement operation.

   Valid values are 512, 1024, 1536, 2048, 3072, and 4096 bits.

6. Enter the TLS cipher suites (in order of preference) that the server is to use in the **Cipher Suites** field.

   These cipher suites are documented in RFC 2246, "The TLS Protocol Version 1.2," and other TLS-related RFCs and draft RFCs.

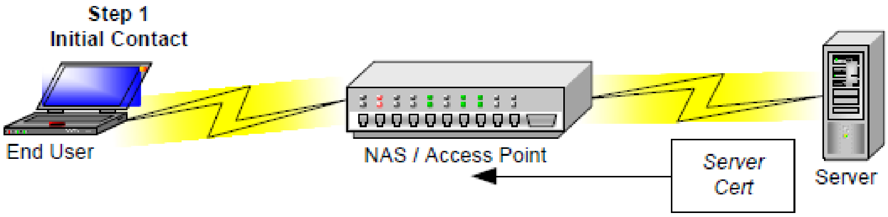
   Default value is 0x3C, 0x3D, 0x67, 0x6B, 0x40, 0x6A, 0x9C, 0x9D, 0x9E, 0x9F, 0xA2, 0xA3.

EAP-TTLS

EAP-TTLS (Tunneled Transport Layer Security) is designed to provide authentication that is as strong as EAP-TLS, but it does not require that each user be issued a certificate. Instead, only the authentication servers are issued certificates. User authentication is performed by password; but the password credentials are transported in a securely encrypted "tunnel" established based upon the server certificates.

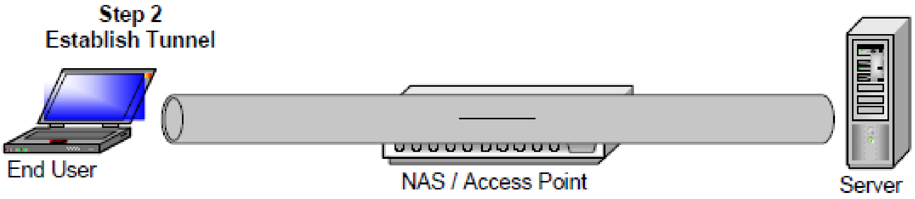
1. After the authentication server determines that the user has made an authentication request, it sends its certificate to the user's system.

   Figure 131: Server Certificate Sent to RAS



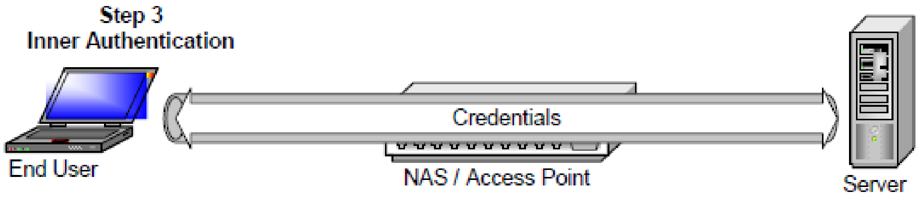
2. The authentication server's certificate is used to establish a tunnel between the user and the server.

   Figure 132: Tunnel Established



3. Once the tunnel is established, credentials can be exchanged safely between the server and the user since tunnels encrypt all data in a secure fashion. This stage is called ***inner authentication.***

   Figure 133: Inner Authentication



With EAP-TTLS, it is not necessary to create a new infrastructure of user certificates. User authentication is performed against the same security database that is already in use on the corporate LAN; for example, Windows Domain Controllers, SQL or LDAP databases, or token systems.

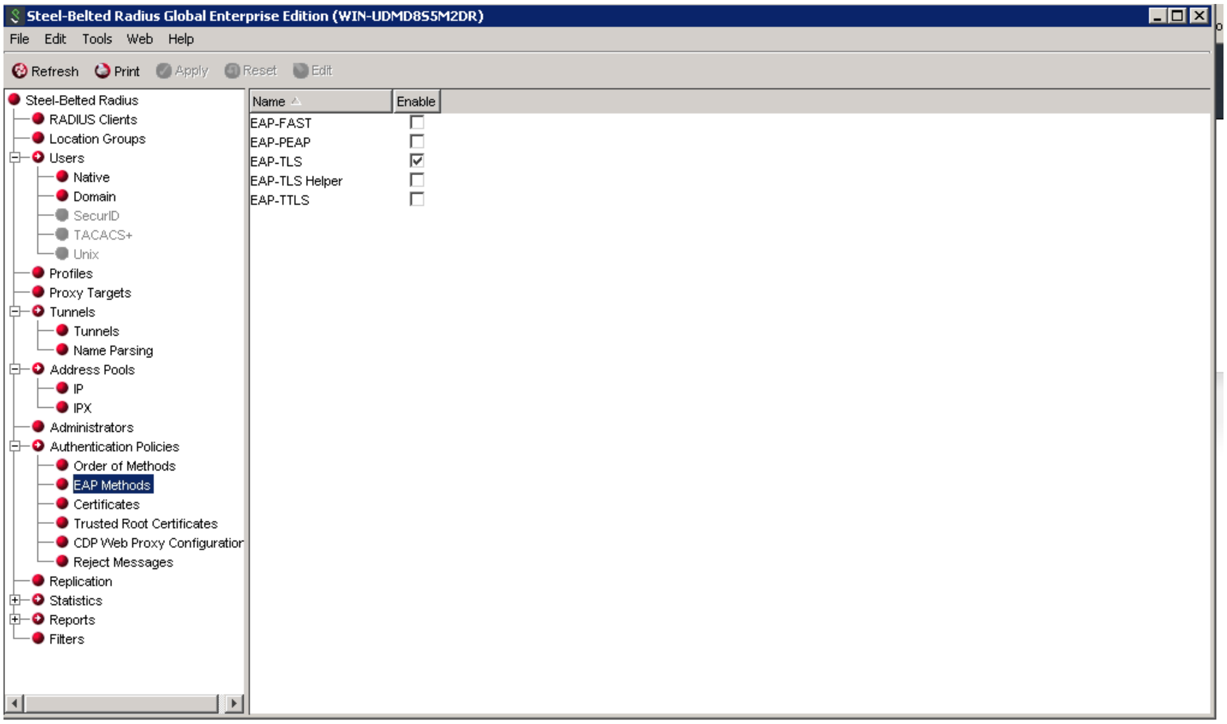
The routing of the inner authentication request can be handled either by means of standard Steel-Belted Radius authentication request routing or by means of a directed realm. If your EAP-TTLS tunnel ends at a dedicated server and all the inner authentication requests are to be performed by other servers, you should use standard request routing so the proxy realm target can be determined in a standard fashion (that is, the decoration of the username revealed by inner authentication). If your EAP-TTLS tunnel and inner authentication are handled by the same server, you can use a directed realm to specify which authentication method(s) handle the inner authentication.

Configuring EAP-TTLS

To configure EAP-TTLS as an authentication method:

1. Select Authentication Policies > EAP Methods to open the EAP Methods panel.

Figure 134: EAP Methods Panel



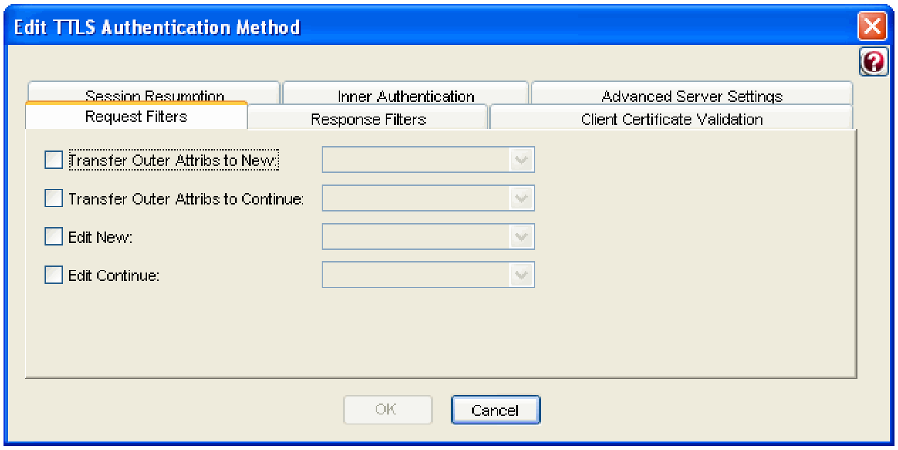
Name	Enable
EAP-FAST	☐
EAP-PEAP	☐
EAP-TLS	☑
EAP-TLS Helper	☐
EAP-TTLS	☐

2. Click the Enable check box for the EAP-TTLS method.

3. Select the EAP-TTLS entry and click the Edit button on the toolbar (or double-click the EAP-TTLS entry).

   The Edit TTLS Authentication Method dialog opens.

Figure 135: Edit TTLS Authentication Method dialog


Edit TTLS Authentication Method
Session Resumption
Inner Authentication
Advanced Server Settings
Request Filters
Response Filters
Client Certificate Validation
☐ Transfer Outer Attribs to New:
☐ Transfer Outer Attribs to Continue:
☐ Edit New:
☐ Edit Continue:
OK
Cancel

4. Use the tabs in the Edit TTLS Authentication Method dialog to configure the following settings:

    - Request filters
    - Response filters
    - Client certificate validation
    - Session resumption
    - Inner authentication
    - Advanced server settings

    Each configuration task is described separately below.

Configuring Request Filters

Request filters affect the attributes of inner authentication requests. By default, Steel-Belted Radius does not use request filters.
To configure request filtering for the EAP-TTLS protocol:

1. Click the **Request Filters** tab in the Edit TTLS Authentication method dialog.
2. Optionally, click the **Transfer Outer Attribs to New** check box and select the filter you want to use from the dropdown list.

    This filter affects only a new inner authentication request (rather than continuations of previous requests).

    - If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.
    - If this filter is not specified, no attributes from the outer request are transferred to the inner request.

3. Optionally, click the **Transfer Outer Attribs to Continue** check box and select the filter you want to use from the dropdown list.

This filter affects only a continued inner authentication request (rather than the first inner authentication request). If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.

If this filter is not specified, no attributes from the outer request are transferred to the inner request.

4. Optionally, click the **Edit New** check box and select the filter you want to use from the dropdown list.

   This filter affects only a new inner authentication request (rather than continuations of previous requests). If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (by the filter specified in Step 2) and attributes included in the inner authentication request sent through the tunnel by the client.

   If this filter is not specified, the request remains unaltered.

5. Optionally, click the **Edit Continue** check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than a new inner authentication request). If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (by the filter specified in Step 3) and attributes included in the inner authentication request sent through the tunnel by the client.

   If this filter is not specified, the request remains unaltered.

Configuring Response Filters

Response filters affect the attributes in the final response (Access-Accept or Access-Reject) returned to the originating NAD. By default, Steel-Belted Radius does not use response filters.

To configure response filtering for the EAP-TTLS protocol:

1. Click the **Response Filters** tab in the Edit TTLS Authentication method dialog.

2. Optionally, click the **Transfer Inner Attribs to Accept** check box and select the filter you want to use from the dropdown list.

   This filter affects only an outer Access-Accept response that is sent back to a network access device.

   - If this filter is specified, the filter is applied to the inner authentication response and all resulting attributes are transferred to the outer authentication response.
   - If this filter is not specified, no inner authentication response attributes are transferred to the outer authentication response.

3. Optionally, click the **Transfer Inner Attribs to Reject** check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than the first inner authentication request). If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.

If this filter is not specified, no attributes from the outer request are transferred to the inner request.

Configuring Client Certificate Validation

Client certificate validation settings let you specify how Steel-Belted Radius performs certificate revocation list (CRL) checking.

To configure session resumption for the EAP-TTLS protocol:

1. Click the **Client Certificate Validation** tab in the Edit TTLS Authentication Method dialog.
2. Click the **Enable CRL Checking** check box to enable CRL checking.
3. If you want to require that the client must provide a certificate as part of the TTLS exchange, click the **Require Client Certificate** check box.
4. Enter the number of seconds that EAP-TTLS will wait for a CRL checking transaction to complete when the CRL check involves a CRL retrieval in the **Retrieval Timeout** field.

   When CRL retrieval takes longer than the specified time, the user’s authentication request results in a reject.
5. Enter the number of seconds during which a CRL is still considered acceptable after it has expired in the **Expiration Grace Period** field.

   EAP-TTLS always attempts to retrieve a new CRL when it is presented with a certificate chain and it finds an expired CRL in its cache.

   - If you enter 0 (strict expiration mode), EAP-TTLS does not accept a CRL that has expired.
   - If you enter a value greater than 0 (lax expiration mode), EAP-TTLS considers the expired CRL as an acceptable stand-in from the time the CRL expires to the time the grace period ends.
6. Click the **Allow Missing CDP Attribute** check box if you want Steel-Belted Radius to accept a non-root certificate that does not have a CDP attribute.

   Without a CDP attribute, EAP-TTLS will not know how to retrieve a CRL and will not be able to perform a revocation check on the certificate.

   - If you click the **Allow Missing CDP Attribute** check box, EAP-TTLS allows such certificates and skips CRL checking for them.
   - If you clear the **Allow Missing CDP Attribute** check box, EAP-TTLS does not accept a CRL with a missing CDP attribute.
7. If you want to specify a CRL cache timeout period, click the **CRL Cache Timeout** Period check box and enter the number of hours in the timeout period in the **hours** field.

   - If you do not enable this setting, the CRL will be refreshed whenever it expires.
   - If you enable this setting and enter 0, Steel-Belted Radius always regards the CRL in the cache as expired and downloads a new CRL every time it receives a client certificate request.
   - If you enable this setting and enter a number greater than 0, the CRL begins to expire when the age of the CRL in the cache exceeds the number of hours specified in this field or when the

scheduled CRL expiration time occurs, whichever comes first.

After a CRL has expired (because its scheduled expiration time has passed or because the CRL cache has timed out), Steel-Belted Radius uses the expiration grace period to determine whether it should use the current CRL.

8. Enter the name of the LDAP server to use if the CDP contains a value that begins with the string //ldap:\\\ in the Default LDAP Server Name field.

   CDPs generated by some CAs do not include the identity of the LDAP server. If you expect to encounter certificates with this style CDP, specify the name of the LDAP server that contains the CRLs.

   If you don't specify a server name and such certificates are encountered, the CRL retrieval fails.

Configuring Session Resumption

Session resumption settings let you specify whether session resumption is permitted and under what circumstances session resumption is performed.

Note: For session resumption to work, the network access device must be configured to handle the Session-Timeout return list attribute, because the network access device must be able to tell the sclient to reauthenticate after the session timer has expired.

To configure session resumption for the EAP-TTLS protocol:

1. Click the Session Resumption tab in the Edit TTLS Authentication Method dialog.

2. Enter the maximum number of seconds you want the client to remain connected to the network access device before having to re-authenticate in the **Session TImeout** field.

   - If you enter 0, no Session-Limit attribute is generated. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

   - If you enter a number greater than 0, the lesser of this value and the remaining resumption limit is sent in a Session-Limit attribute to the RADIUS client on the RADIUS Access-Accept response.

   Entering a value such as 600 (10 minutes) does not necessarily cause a full re-authentication to occur every 10 minutes. You can configure the resumption limit to make most re-authentications fast and computationallycheap.

3. Enter the value that you want returned in a Termination-Action attribute in the **Termination Action** field.

   The Termination-Action attribute is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached. Valid values are:

   - -1: Do not send the attribute.

   - 0: Send the Termination-Action attribute with a value of 0.

   - 1: Send the Termination-Action attribute with a value of 1.

   Default value is -1. Note that this does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

4. Enter the maximum number of seconds you want the client to be able to re-authenticate using the TLS session resumption feature in the **Resumption Limit** field.

   This type of re-authentication is fast and computationally cheap. It does, however, depend on previous authentications and may not be considered as secure as a complete (computationally expensive) authentication. Specifying a value of 0 disables the session resumption feature.

Configuring Inner Authentication Settings

The inner authentication settings let you specify the way in which the inner authentication step is to operate.

To configure inner authentication settings for the EAP-TTLS protocol:

1. Click the **Inner Authentication** tab in the Edit TTLS Authentication Method dialog.
2. If you want requests to be routed based on the methods listed in the directed realm, enter the name of a directed realm in the **Directed Realm** field.

   Omitting this setting causes the inner authentication request to be handled like any other request received from a network access device.
3. If you want requests to be processed by means of a realm selection script, enter the name of a script in the **Realm Selection Script** field.

Configuring Advanced Server Settings

Advanced server settings specify the manner in which the inner authentication step operates. To configure advanced server settings for the EAP-TTLS protocol:

1. Click the Advanced Server Settings tab in the Edit TTLS Authentication Method dialog.
2. Enter the maximum length of the TLS message that may be generated during each iteration of the TLS exchange. in the **TLS Message Fragment Length** field.

   Enter a number in the range 500–4096. This value affects the number of RADIUS challenge/response round-trips required to conclude the TLS exchange. A value of 1400 may result in 6 round-trips, while a value of 500 may result in 15 round-trips.

   Some Access Points may have problems with RADIUS responses or EAP messages that exceed the size of one Ethernet frame (1500 bytes including IP/UDP headers).

   The default length for TLS messages is 1020 bytes, which prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame.
3. Enter the maximum number of seconds you want to allow for the EAP authentication sequence in the **Max Transaction Time** field.

   If the EAP authentication sequence takes longer than the number of seconds specified in this field, Steel-Belted Radius terminates the user authentication.
4. Enable the **Return MPPE Keys** check box to specify whether the TTLS authentication method includes RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Access-Accept response sent to the Access Point.

   You should enable this option if the Access Point needs to key the WEP encryption. If the Access Point is authenticating only end-users and WEP is not being used, you can clear this check box.

5. Use the DH Prime Bits list to specify the number of bits in the prime number that the module uses for Diffie-Hellman exponentiation.

   Selecting a longer prime number makes the system less susceptible to certain types of attacks but requires more CPU processing to compute the Diffie-Hellman key agreement operation.

   Valid values are 512, 1024, 1536, 2048, 3072, and 4096 bits.

6. Enter the TLS cipher suites (in order of preference) that the server is to use in the Cipher Suites field.

   These cipher suites are documented in RFC 2246, "The TLS Protocol Version 1.2," and other TLS-related RFCs and draft RFCs.

   Default value is 0x3C, 0x3D, 0x67, 0x6B, 0x40, 0x6A, 0x9C, 0x9D, 0x9E, 0x9F, 0xA2, 0xA3.

EAP-PEAP

The EAP-PEAP (Protected EAP) protocol is similar to EAP-TTLS. Unlike EAP-TTLS, which can tunnel any kind of authentication request (such as PAP or CHAP) and extended attributes, PEAP can tunnel only other EAP protocols inside its connection.

EAP-PEAP works in two phases:

- In Phase 1, the client authenticates the server and uses a TLS handshake to create an encrypted tunnel.
- In Phase 2, the server authenticates the user or machine credentials using an EAP authentication protocol. The EAP authentication is protected by the encrypted tunnel created in Phase 1. The authentication type negotiated during Phase 2 can be any valid EAP type, such as GTC (Generic Token Card) or MS-CHAPv2.

Microsoft's implementation of PEAP and Cisco's implementation of PEAP supports different methods of client authentication through the TLS tunnel.

- The Microsoft PEAP implementation requires MS-CHAP-V2 for client authentication.
- The Cisco PEAP implementation supports client authentication by EAP-Generic Token, which Cisco uses both for authenticating token cards and for authenticating users against Windows domain/ Active Directory accounts.

The Cisco PEAP implementation supports the ability to hide username identities until the TLS encrypted tunnel is established and authentication phase is complete. The Microsoft PEAP implementation sends the username in clear text in Phase 1 of PEAP authentication.

Steel-Belted Radius supports both Microsoft PEAP and Cisco PEAP.

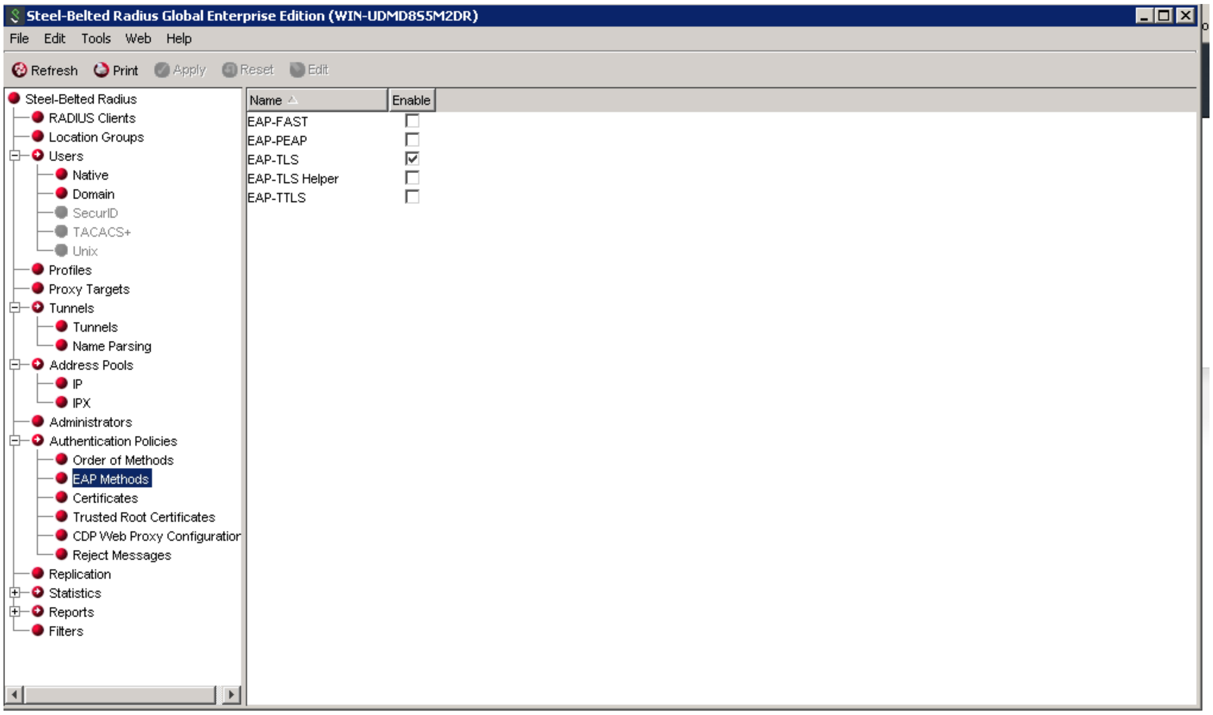
Configuring EAP-PEAP

Note: You must configure the server certificate for the Steel-Belted Radius server before you use the EAP-PEAP authentication method. For information on configuring your server certificate, see "Configuring Server Certificates".

To configure EAP-PEAP on a Steel-Belted Radius server:

1. Select Authentication Policies > EAP Methods to open the EAP Methods panel.

Figure 136: EAP Methods Panel



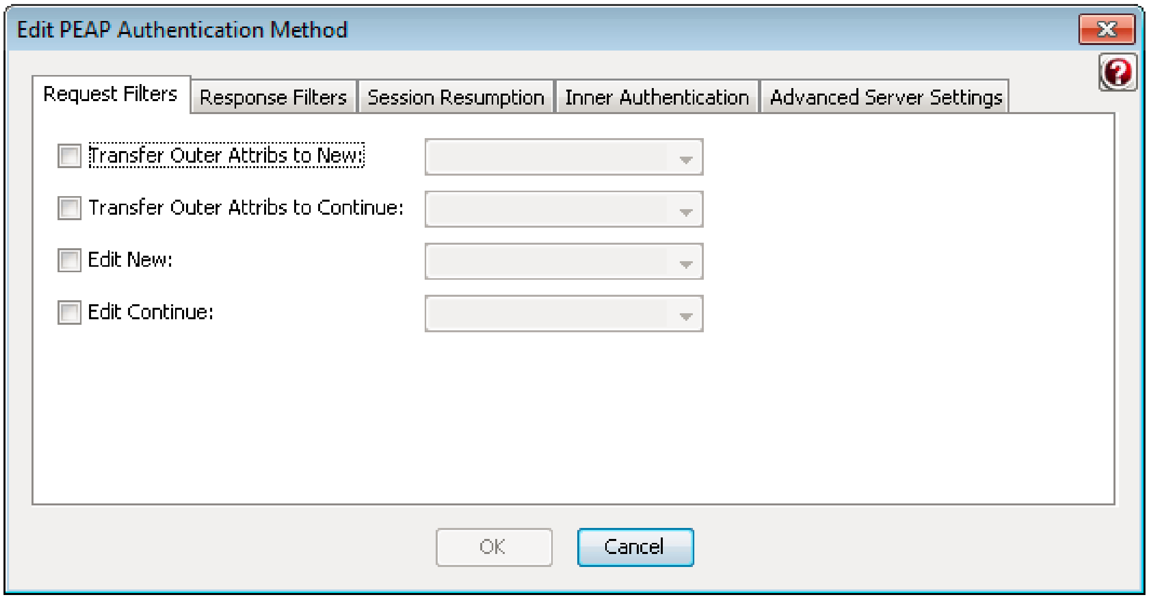
Name	Enable
EAP-FAST	□
EAP-PEAP	□
EAP-TLS	☑
EAP-TLS Helper	□
EAP-TTLS	□

2. Click the Enable check box for the EAP-PEAP authentication method.

3. Select the EAP-PEAP entry and click the Edit button on the toolbar (or double-click the EAP-PEAP entry).

   The Edit PEAP Authentication Method dialog opens.

Figure 137: Edit PEAP Authentication Method dialog



□ Transfer Outer Attribs to New:

□ Transfer Outer Attribs to Continue:

□ Edit New:

□ Edit Continue:

4. Use the tabs in the Edit PEAP Authentication Method dialog to configure the following settings:

- Request filters
- Response filters
- Session resumption
- Inner authentication
- Advanced server settings

Each configuration task is described separately below.

Configuring Request Filters

Request filters affect the attributes of inner authentication requests. By default, Steel-Belted Radius does not use request filters.

ⓘ Note: You must configure filters using the Filters panel before you can associate them with the EAP-PEAP authentication method. For information on configuring filters, refer to “Setting Up Filters via Legacy SBR Administrator”.

To configure request filtering for the EAP-PEAP protocol:

1. Click the **Request Filters** tab in the Edit PEAP Authentication method dialog.
2. Optionally, click the **Transfer Outer Attribs to New check** box and select the filter you want to use from the dropdown list.

   This filter affects only a new inner authentication request (rather than continuations of previous requests).

   - If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.
   - If this filter is not specified, no attributes from the outer request are transferred to the inner request.
3. Optionally, click the Transfer Outer Attribs to Continue check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than the first inner authentication request). If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.

   If this filter is not specified, no attributes from the outer request are transferred to the inner request.
4. Optionally, click the **Edit New** check box and select the filter you want to use from the dropdown list.

   This filter affects only a new inner authentication request (rather than continuations of previous requests). If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (by the filter specified in Step 2) and attributes included in the inner authentication request sent through the tunnel by the client.

If this filter is not specified, the request remains unaltered.

5. Optionally, click the **Edit Continue** check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than a new inner authentication request). If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (by the filter specified in Step 3) and attributes included in the inner authentication request sent through the tunnel by the client.

   If this filter is not specified, the request remains unaltered.

Configuring Response Filters

Response filters affect the attributes in the final response (Access-Accept or Access-Reject) returned to the originating NAD. By default, Steel-Belted Radius does not use response filters.

To configure response filtering for the EAP-PEAP protocol:

1. Click the **Response Filters** tab in the Edit PEAP Authentication method dialog.
2. Optionally, click the **Transfer Inner Attribs to Accept** check box and select the filter you want to use from the dropdown list.

   This filter affects only an outer Access-Accept response that is sent back to a network access device.

   - If this filter is specified, the filter is applied to the inner authentication response and all resulting attributes are transferred to the outer authentication response.
   - If this filter is not specified, no inner authentication response attributes are transferred to the outer authentication response.
3. Optionally, click the Transfer Inner Attribs to Reject check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than the first inner authentication request). If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.

   If this filter is not specified, no attributes from the outer request are transferred to the inner request.

Configuring Session Resumption

Session resumption settings let you specify whether session resumption is permitted and under what circumstances session resumption is performed.

Note: For session resumption to work, the network access device must be configured to handle the Session-Timeout return list attribute, because the network access device must be able to tell the client to reauthenticate after the session timer has expired.

To configure session resumption for the EAP-PEAP protocol:

1. Click the **Session Resumption** tab in the Edit PEAP Authentication method dialog.
2. Enter the maximum number of seconds you want the client to remain connected to the network

access device before having to re-authenticate in the **Session TImeout** field.

If you enter a number greater than 0, the lesser of this value and the remaining resumption limit is sent in a Session-Limit attribute to the RADIUS client on the RADIUS Access-Accept response.

If you enter 0, no Session-Limit attribute is generated. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

Entering a value such as 600 (10 minutes) does not necessarily cause a full re-authentication to occur every 10 minutes. You can configure the resumption limit to make most re-authentications fast and computationallycheap.

3. Enter the value that you want returned in a Termination-Action attribute in the **Termination Action** field.

   The Termination-Action attribute is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached. Valid values are:

   - -1: Do not send the attribute.
   - 0: Send the Termination-Action attribute with a value of 0.
   - 1: Send the Termination-Action attribute with a value of 1.

   Default value is -1. Note that this does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

4. Enter the maximum number of seconds you want the client to be able to re-authenticate using the TLS session resumption feature in the **Resumption Limit** field.

   This type of re-authentication is fast and computationally cheap. It does, however, depend on previous authentications and may not be considered as secure as a complete (computationally expensive) authentication. Specifying a value of 0 disables the session resumption feature.

Configuring Inner Authentication Settings

Inner authentication settings let you specify the manner in wich the inner authentication step operates. To configure inner authentication settings for the EAP-PEAP protocol:

1. Click the **Inner Authentication** tab in the Edit PEAP Authentication method dialog.

2. Optionally, enter the name of a directed realm in the **Directed Realm** field.

   Specifying the name of a directed realm causes the request to be routed based on the methods listed in the directed realm. Omitting this setting causes the inner authentication request to be handled like any other request received from a network access device.

3. Optionally, enter the name of a realm selection script in the **Realm Selection Script** field.

   You must license the Steel-Belted Radius scripting module to use realm selection scripts. For information on the Steel-Belted Radius scripting module, refer to the Steel-Belted Radius Scripting Guide.

Configuring Advanced Server Settings

Advanced server settings specify the manner in which the inner authentication step operates. To configure advanced server settings for the EAP-PEAP protocol:

1. Click the Advanced Server Settings tab in the Edit PEAP Authentication method dialog.
2. Enter the maximum length of the TLS message that may be generated during each iteration of the TLS exchange. in the **TLS Message Fragment Length** field.

   Enter a number in the range 500–4096. This value affects the number of RADIUS challenge/response round-trips required to conclude the TLS exchange. A value of 1400 may result in 6 round-trips, while a value of 500 may result in 15 round-trips.

   Some Access Points may have problems with RADIUS responses or EAP messages that exceed the size of one Ethernet frame (1500 bytes including IP/UDP headers).

   The default length for TLS messages is 1020 bytes, which prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame.
3. Enter xxx in the **Max Transaction** Time field.
4. Enable the **Return MPPE Keys** check box to specify whether the EAP-PEAP module includes RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Access-Accept response sent to the Access Point.

   You should enable this option if the Access Point needs to key the WEP encryption. If the Access Point is authenticating only end-users and WEP is not being used, you can clear this check box.
5. Use the DH Prime Bits list to specify the number of bits in the prime number that the module uses for Diffie-Hellman exponentiation.

   Selecting a longer prime number makes the system less susceptible to certain types of attacks but requires more CPU processing to compute the Diffie-Hellman key agreement operation.

   Valid values are 512, 1024, 1536, 2048, 3072, and 4096 bits.
6. Enter the TLS cipher suites (in order of preference) that the server is to use in the **Cipher Suites** field.

   These cipher suites are documented in RFC 2246, "The TLS Protocol Version1.2," and other TLS-related RFCs and draft RFCs.

   Default value is 0x3C, 0x3D, 0x67, 0x6B, 0x40, 0x6A, 0x9C, 0x9D, 0x9E, 0x9F, 0xA2, 0xA3.
7. Enter the minimum version of the PEAP protocol that the server should negotiate in the **PEAP Minimum Version** field.

   If you enter 0, the server negotiates version 0, which is compatible with Microsoft's initial PEAP implementation (shipped in Microsoft XP Service Pack 1).

   If you enter 1, the server negotiates version 1, which is compatible with Cisco's initial PEAP implementation (shipped in Cisco ACU).

   ⓘ Note: The value entered in this setting must be less than or equal to the value entered for the PEAP Maximum Version field.
8. Enter the maximum version of the PEAP protocol that the server should negotiate in the PEAP

Maximum Version field.

If you enter 0, the server negotiates version 0, which is compatible with Microsoft's initial PEAP implementation (shipped in Microsoft XP Service Pack 1).

If you enter 1, the server negotiates version 1, which is compatible with Cisco's initial PEAP implementation (shipped in Cisco ACU).

Note: The value entered in this setting must be equal to or greater than the value entered for the **PEAP Minimum Version** field.

Configuring Server Certificates

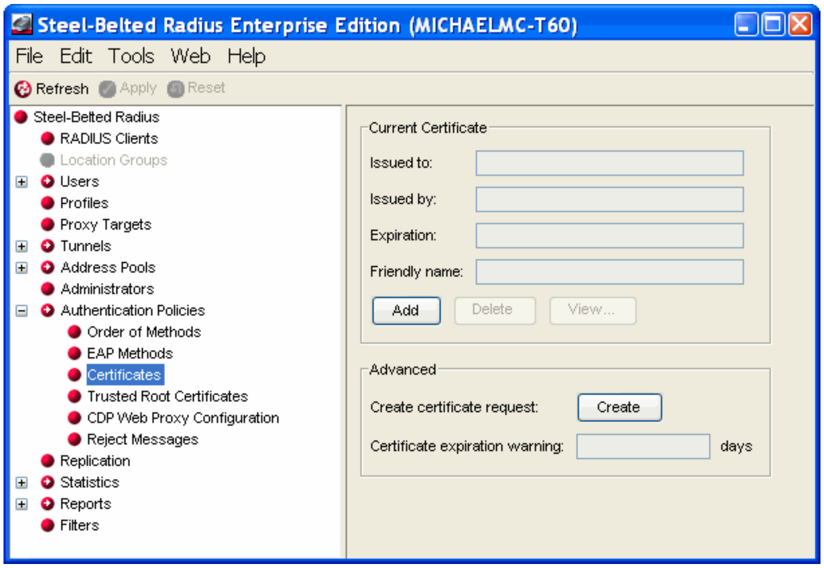
You must install cert if

Adding a Server Certificate

To add a certificate to the Steel-Belted Radius server:

1. Select Authentication Policies > Certificates to open the Certificates panel (Figure 138: Certificates Panel).

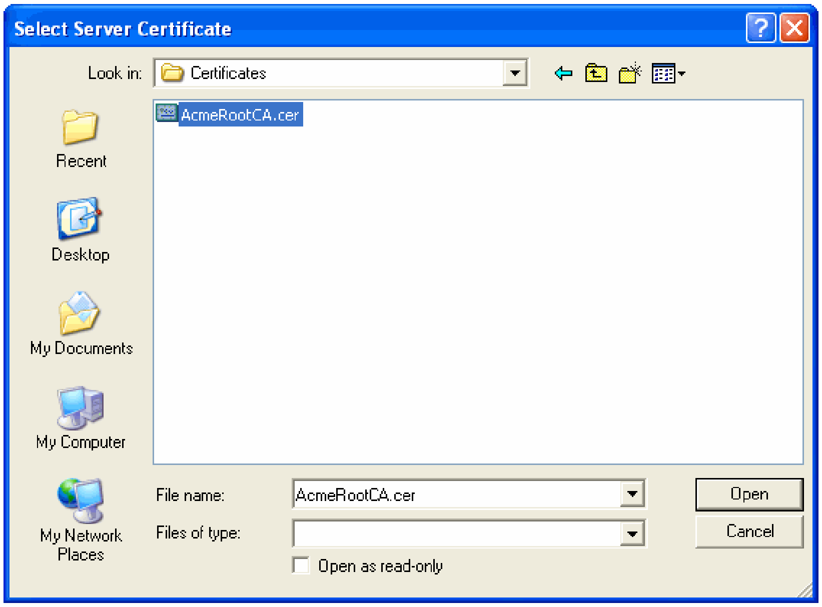
Figure 138: Certificates Panel



2. Click the **Add** button.

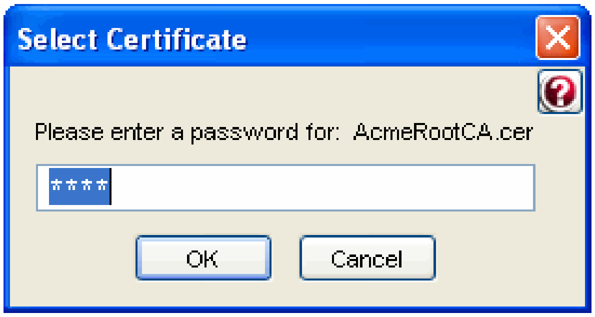
3. When the Select Server Certificate dialog (Figure 139: Select Server Certificate Dialog) opens, navigate to the location of your server certificate, select the certificate you want to use, and click **Open.**

Figure 139: Select Server Certificate Dialog



4. When the Select Certificate dialog (Figure 140: Select Certificate Dialog) opens, enter the password for the certificate you selected and click OK.

Figure 140: Select Certificate Dialog

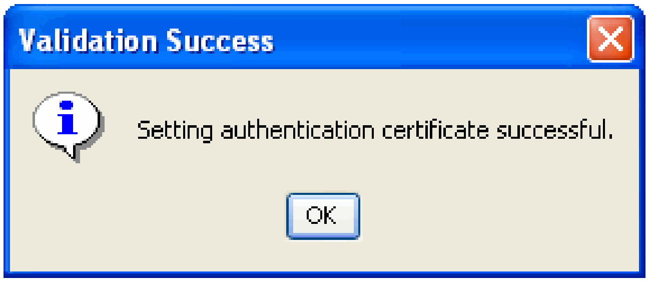

Select Certificate

Please enter a password for: AcmeRootCA.cer

****

5. When the Validation Success dialog appears, click OK.

Figure 141: Validation Success Dialog


Validation Success

Setting authentication certificate successful.

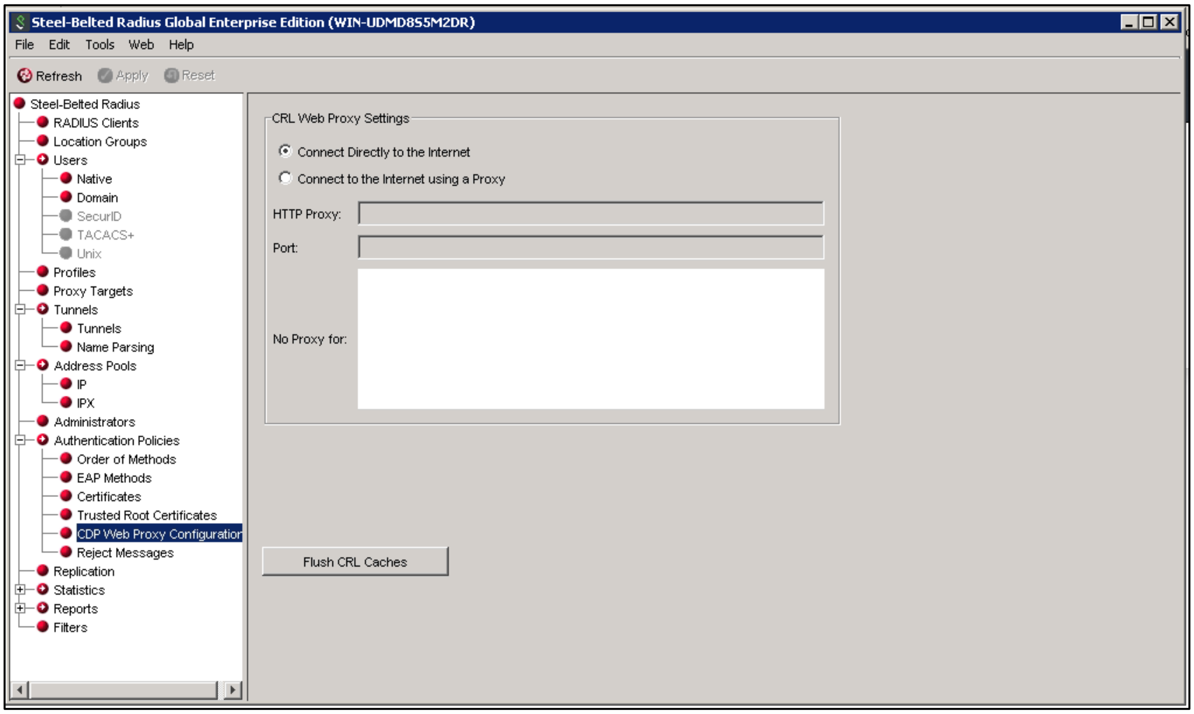
Configuring a CDP Web Proxy

Your network security policies may prohibit Steel-Belted Radius from making a direct HTTP connection to a CRL distribution points (CDP). You can configure an HTTP proxy server to relay requests from Steel-Belted Radius for updated certificate revocation lists to an external CDP.

To configure a CDP web proxy server:

1. Select Authentication Policies > CDP Web Proxy Configuration to open the CDP Web Proxy Configuration panel.

Figure 142: CDP Web Proxy Configuration Panel



CRL Web Proxy Settings

● Connect Directly to the Internet

○ Connect to the Internet using a Proxy

HTTP Proxy:

Port:

No Proxy for:

2. Specify whether you want Steel-Belted Radius to use a proxy to connect to an external CDP.

   - If you click **Connect Directly to the Internet**, Steel-Belted Radius can connect to an external CDP without going through an HTTP proxy. If you select this option, you can skip the rest of this procedure.

   - If you click **Connect to the Internet Using a Proxy**, Steel-Belted Radius must go through an HTTP proxy to connect to a CDP.

3. Enter the name or IP address of the HTTP proxy in the **HTTP Proxy** field.

4. Enter the port number to use for the HTTP proxy in the **Port** field.

5. Optionally, identify the hosts for which no HTTP proxy is required in the **No Proxy For** field. If a CDP host matches an entry in this field, Steel-Belted Radius bypasses the HTTP proxy and attempts to open a connect to the host directly.

   You can enter the names or IP addresses of hosts or the names of domains, separating each entry with a comma or semi-colon. Steel-Belted Radius compares IP addresses and host names using an exact string match. For example, if you enter cdp.pulsesecure.net in the exclusion list, that will match the CDP hostname cdp.pulsesecure.net but not host.cdp.pulsesecure.net or host-cdp.pulsesecure.net.

   To exclude all hosts within a domain (but not the host name that matches the domain name), start the domain name with a period (.pulsesecure.net). To exclude both the host and the domain pulsesecure.net, create two entries in the exclusion list (.pulsesecure.net, pulsesecure.net).

Wildcard matching for host or domain names is not supported.

The values localhost and 127.0.0.1 are included in the No Proxy For list by default.

6. Optionally, click the **Flush CRL Caches** button to purge all information in the TLS and TTLS CRL caches immediately. When the Flush CRL Caches button is clicked, all CRL entries for registered clients are purged from the in-memory cache and deletes all files from the CRL cache directories.

Note: If you click the **Flush CRL Caches** button, the caches are purged immediately. You are not asked to confirm your action.

Configuring the Server

Depending on your authentication requirements, you may need to configure Steel-Belted Radius to work with an external SQL or LDAP database, RSA SecurID service, or TACACS+.

Configuring External Databases (Linux)

If you run Linux and want to use external databases for authentication or accounting purposes (and you did not configure this feature when prompted by the Steel-Belted Radius installation script), you can set up external database configuration settings.

To configure Steel-Belted Radius to work with an external database:

1. Optionally, perform the instructions in "Configuring SQL Authentication" and/or "Configuring SQL Accounting".
2. If you want to use Steel-Belted Radius with an LDAP database, review your LDAP database vendor's documentation.
3. Perform the instructions in "Configuring LDAP Authentication".

Configuring SecurID Authentication

If you want to use SecurID authentication, you must configure Steel-Belted Radius to communicate with the RSA SecurID server.

Perform the following steps to configure a Steel-Belted Radius server to work with an RSA SecurID server. If you are not familiar with the RSA SecurID server, contact your RSA SecurID server administrator for assistance.

1. Verify that the Steel-Belted Radius server has an entry on the RSA SecurID server.

   Start the RSA SecurID server administration program and display the list of clients. If the list of clients does not include the Steel-Belted Radius server, select **Client > Add Client** and complete the Client window, giving the Steel-Belted Radius server a Client type of **Net OS Client**.

2. Copy the sdconf.rec file from the \ACE\data directory on the RSA SecurID server to the appropriate directory on the Steel-Belted Radius server:

   - **Windows:** C:\winnt\system32

- **Linux:** the directory that contains the radius daemon on the Steel-Belted Radius server.

3. Edit the [SecurID] section of radius.ini. The radius.ini file is found in the same directory as the Steel-Belted Radius service or daemon.

   Verify that the **CachePasscodes** field is set to **yes** and the **SecondsToCachePasscodes** field is set to an appropriate number of seconds. These settings ensure that authenticated SecurID users can open a second B-channel during an ISDN connection.

4. Edit the [SecurID] section of the eap.ini file, which is found in the same directory as the Steel-Belted Radius service or daemon.

   Verify that the EAP settings in this section are enabled (remove the semi-colon from the start of each line) if you plan to use RSA SecurID authentication with EAP Generic-Token protocol support. The client system must support this protocol as well for this combination to work.

5. If you copy the sdconf.rec file after the Steel-Belted Radius service (daemon) has been started, or if you edit the radius.ini or eap.ini files after Steel-Belted Radius has been started, stop and restart Steel-Belted Radius.

6. Verify connectivity between the Steel-Belted Radius server and the RSA SecurID server.

   The RSA SecurID server offers a monitoring window on which it logs every authentication transaction, complete with the reason for the accept or reject decision. You can verify that pass-through to RSA SecurID is working, by creating a SecurID User called <ANY> and then attempting to access the network. Look for your request on the RSA SecurID monitor screen. If access is denied, you'll know that there's a configuration problem. Try these steps again, or contact your RSA SecurID administrator for assistance.

These steps complete initial setup of the two servers. To fully enable pass-through authentication to the RSA SecurID server, you must also set up the SecurID authentication method.

Configuring the Location of the sdconf.rec File

The VAR_ACE variable in the sbrd script file (Linux) lets you specify the directory holding the sdconf.rec file. The VAR_ACE variable must be exported so that Steel-Belted Radius can use it.

For example:

```
VAR_ACE=”radiusdir/ace”

export VAR_ACE
```

This variable is set by default in the file to point to the radiusdir directory. If the variable is not set at all in the file, the server sets the value of this variable to

```
/var/ace.
```

Configuring TACACS+ Authentication

If you want to use TACACS+ authentication, you must configure Steel-Belted Radius to communicate with the TACACS+ server.

Perform the following steps to configure a Steel-Belted Radius server to work with a TACACS+ server.

1. Verify the tacplus.ini file is present in the Steel-Belted Radius directory.

The tacplus.ini file must be present in the same directory as the Steel-Belted Radius service (in the case of Windows, usually C:\Program Files (x86)\Pulse Secure\Steel-Belted Radius\Service), or daemon (in the case of Linux). This happens automatically following installation.

2. Edit the tacplus.ini file to identify the shared secret and host machine that you use for TACACS+. For more information on the tacplus.ini file, refer to the Steel-Belted Radius Reference Guide.

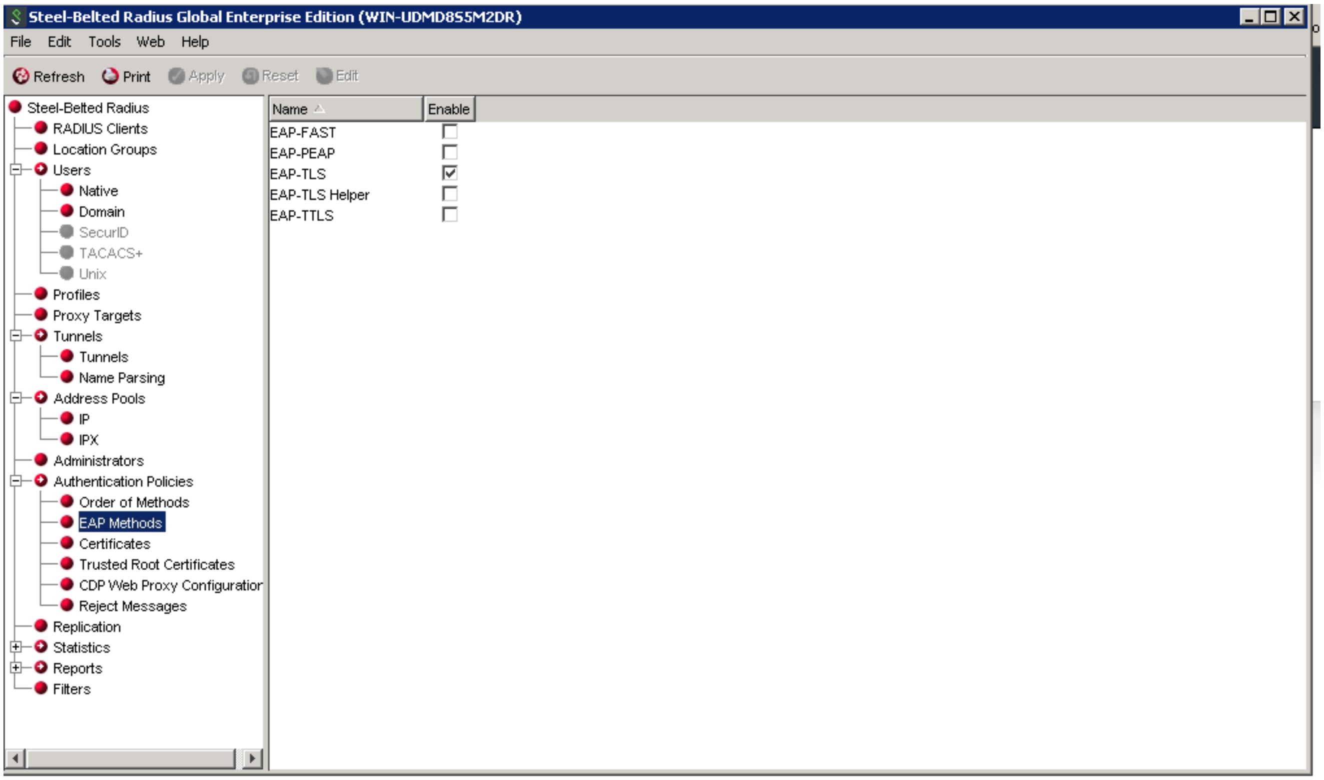
3. If you edit tacplus.ini after Steel-Belted Radius has been started, then you must stop and restart it before your changes take effect.

To enable pass-through authentication to the TACACS+ server, you must also set up the TACACS+ authentication method. For more information, see "Configuring TACACS+ Authentication".

Activating EAP Methods

The EAP Methods panel permits you to activate authentication methods and define the order in which different authentication methods are attempted.

Figure 143: EAP Methods Panel



Name	Enable
EAP-FAST	☐
EAP-PEAP	☐
EAP-TLS	☑
EAP-TLS Helper	☐
EAP-TTLS	☐

To use the EAP Methods panel:

1. Select Authentication Policies >EAP Methods in the Sidebar.

2. Click the Enable check box to enable the EAP authentication methods you want Steel-Belted Radius to use.

To revert to the previous settings, click Reset.

Configuring EAP Settings

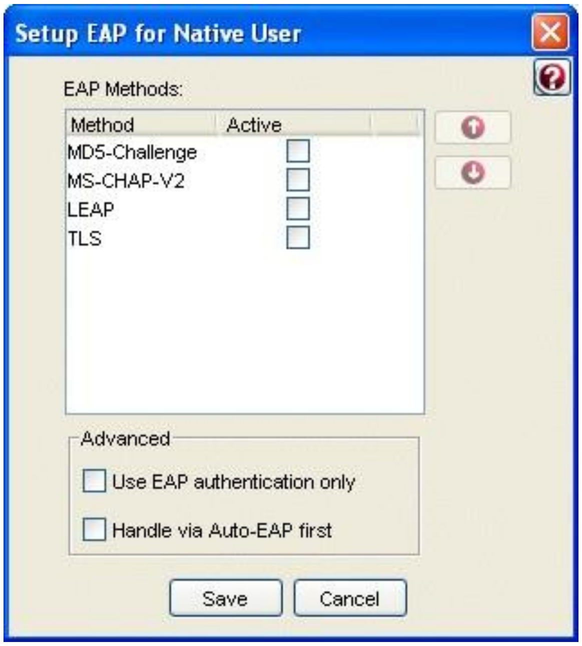
When Steel-Belted Radius receives a username, it does not know in advance to which authentication category this user belongs. It must try each method that it currently has configured and enabled. The authentication methods list allows you to fine-tune the sequence of authentication attempts.

Note: The EAP Setup dialog displays the authentication methods that have been enabled (by editing the Enabled setting in the appropriate *.aut file).

To set up EAP settings for an authentication method:

1. Select the authentication method you want to set up in the Authentication Methods tab.
2. Click the **EAP Setup** button.
   The Setup EAP dialog opens.

Figure 144: Setup EAP Dialog



Setup EAP for Native User

EAP Methods:

Method	Active
MD5-Challenge	□
MS-CHAP-V2	□
LEAP	□
TLS	□

Advanced

□ Use EAP authentication only

□ Handle via Auto-EAP first

3. Optionally, change the order in which the methods are tried by highlighting a method and clicking the **Up** or **Down** buttons.
4. To activate a method, (so that it can be used for authentication), click the **Active** check box.

   If you want to deactivate a method (so that it is not used for authentication), unclick the applicable **Active** check box.
5. If you want to restrict use of this authentication method to requests that contain EAP credentials, click the **Use EAP authentication only** check box.

   When this option is enabled, Steel-Belted Radius prevents the authentication method from being called for any request that does not contain EAP credentials, and bypasses the authentication method if an authentication request specifically requests an EAP protocol that is not listed in the authentication method’s **EAP-Type** list in the **eap.ini** file.
6. If you want Steel-Belted Radius to use an automatic EAP helper to generate credentials for a user, click the **Handle via Auto-EAP first** check box.

You should unclick the check box if an authentication method is capable of handling EAP credentials on its own (without an EAP helper).

Refer to "First-Handle-Via-Auto-EAP Setting" for more information.

7. Click **Save** to return to the Authentication Methods tab.

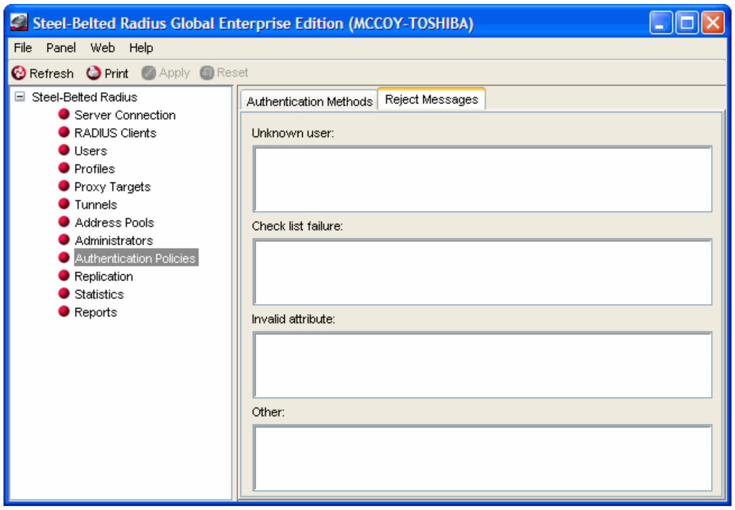
Configuring Authentication Rejection Messages

When Steel-Belted Radius issues an Access-Reject message in response to a failed authentication request, it can identify the reason why the request was rejected. You can configure the message text returned to the RADIUS client (and possibly to the user, if the RADIUS client forwards the message) when a particular type of error occurs. This text is inserted into the standard RADIUS attribute Reply-Message within the Access-Reject response.

To configure the text for authentication rejection messages:

1. Open the Authentication Policies panel.
2. Click the **Reject Messages** tab.

Figure 145: Authentication Policies Panel: Reject Messages Tab



3. Use the **Unknown User** field to specify the message Steel-Belted Radius returns when the username and password authentication failed.
4. Use the **Checklist failure** field to specify the message Steel-Belted Radius returns when the user was authenticated but is being rejected because the RADIUS request did not fulfill the requirements of the checklist.
5. Use the I**nvalid attribute** field to specify the message Steel-Belted Radius returns when the request contained an attribute in violation of the RADIUS specification.
6. Use the **Other** field to specify the message Steel-Belted Radius returns when some other error, such as a resource failure, occurred.
7. When you are asked to confirm that you want to save your changes, click **Yes.**

Chapter 25
Setting Up EAP Authentication Policies via WebGUI

This chapter presents an overview of concepts relating to the Extensible Authentication Protocol (EAP) and describes how to configure Steel-Belted Radius to use EAP authentication methods and plug-ins via WebGUI.

About the Extensible Authentication Protocol

Steel-Belted Radius supports the Extensible Authentication Protocol (EAP), a standard for communication between network access devices and servers that provides for the future extensibility of authentication protocols.

EAP allows specialized knowledge about authentication protocols to be taken out of a NAD so that it acts solely as a conduit between authentication server and client. This means that new types of authentication can be supported by adding the appropriate functionality to server and client, without any changes to PPP or network access devices. When the authentication process is complete, the RADIUS server simply informs the NAD of the result.

Steel-Belted Radius supports several EAP authentication mechanisms, such as TTLS, TLS, PEAP, LEAP, MD5-Challenge, and Generic Token. Support for EAP has been designed to anticipate other authentication types as they become available.

TLS v1.2 (latest version) protocol provides improved flexibility and enhanced security. It supports modern encryption algorithms like SHA-256, AES cipher suites to communicate with RADIUS clients and avoid any weak cipher suite negotiations. OpenSSL is used to support TLS v1.2 and address various security vulnerabilities.

The [EapSettings] section of radius.ini initialization file contains two parameters: AllowTLSFallback and MinimumProtocolVersion.

- Parameter AllowTLSFallback enables fallback to support SSL/TLS protocol versions
- Parameter MinimumProtocolVersion specifies the protocol version (TLSv10/TLSv11/TLSv12) to be used for EAP

Note: AllowTLSFallback option in radius.ini is applicable only for certificate based EAP plugins. The TLS fallback is a unique feature, aimed to provide backward compatibility to deprecated EAP client.

When AllowTLSFallback option is enabled in radius.ini, SBR EAP component will be initialized with protocol independent SSL structure. Depending on EAP client's hello message, SBR will either negotiate in TLS v1.2 or it can degrade itself till SSL v3 compliant server. This option is very specific to EAP plugins only.

For technical details about EAP, see RFC 2284, "PPP Extensible Authentication Protocol (EAP)," and RFC 2869, "RADIUS Extensions."

Handling EAP Requests

The flow of RADIUS packets in an EAP scenario is quite different from the transactions using standard user credentials (for example, PAP or CHAP). Standard user credentials involve the transmission of a RADIUS request from the NAD to Steel-Belted Radius and a response (either an Accept or Reject) from the server back to the NAD.

With EAP, the first packet sent from the NAD to Steel-Belted Radius contains an EAP-Message attribute containing an EAP Identity Response. This is a signal sent by the system being authenticated that it wants to be authenticated by means of EAP. It is now up to Steel-Belted Radius to select the EAP protocol with which it is to authenticate the end-user.

The contents of the User-Name attribute is the only guideline available to Steel-Belted Radius in selecting the appropriate EAP protocol. Should Steel-Belted Radius select an EAP protocol that is not supported by the client, the client has the opportunity to send an EAP-NAK and to request a specific alternate protocol.

Note: Given this general flow, a RADIUS request with EAP credentials must incur a minimum of two network round-trips between the RAS (or Access Point) and the Steel-Belted Radius before reaching a successful conclusion.

Automatic EAP Helpers

Automatic EAP helpers serve as intermediaries between EAP and traditional authentication methods. These helper modules can be configured (using an associated .eap file) to work with existing authentication methods to shield the authentication methods from the particulars of the selected EAP protocol.

Table 30 indicates whether each EAP type is implemented as an EAP helper or stand-alone module in Steel-Belted Radius.

Table 30: EAP Implementations

EAP-Type	Implemented As
EAP-TTLS	Standalone Authentication Method Module
EAP-TLS	Standalone Authentication Method Module
EAP-TLS	Automatic EAP helper
LEAP	Automatic EAP helper for MS-CHAP-v1
EAP Generic-Token	Standalone Authentication Method Module (SecurID)
EAP MD5-Challenge	Automatic EAP helper for CHAP
EAP MS-CHAP-v2	Automatic EAP helper for MS-CHAP-v2 (needed for PEAP)

Whether an automatic EAP helper can be used in conjunction with a specific authentication method depends on what types of credentials the authentication method supports.

The automatic EAP helper that implements EAP MD5-Challenge generates CHAP credentials, while the helper that implements LEAP generates MS-CHAP-v1 credentials. As such, EAP MD5-Challenge can be used only with authentication methods that support CHAP, and LEAP can be used only with authentication methods that support MS-CHAP-v1.

Table 31 summarizes the support for MS-CHAP-v1 and CHAP in the Steel-Belted Radius authentication methods.

Table 31: MS-CHAP-v1 and CHAP Support

Authentication Method	MS-CHAP-V1	CHAP
LDAP	Yes for BindName (password must be stored in the clear or encrypted using enc-md5 in LDAP server), No for Bind	Yes for BindName (password must be stored in the clear or encrypted using enc-md5 in LDAP server), No for Bind
Local	Yes	Yes
Proxy RADIUS	Yes	Yes
SecurID	No	No
SQL	Yes if password is in clear or encrypted using enc-md5 in SQL database	Yes if password is in clear or encrypted using enc-md5 in SQL database
TACACS+	No	Yes
UNIX User	No	No
UNIX Group	No	No
Windows Domain User	Yes (server must be running under SYSTEM account)	No
Windows Domain Group	Yes (server must be running under SYSTEM account)	No

Authentication Request Routing

The order in which authentication methods and automatic EAP helpers are called to handle an authentication request depends on two factors:

- The ordered list of enabled authentication methods (viewable in the Authentication Policies panel in SBR Administrator). Refer to "Activating EAP Methods" for information on using the Authentication Policies panel.
- The EAP-related configuration for each of the enabled authentication methods in the eap.ini file, which you configure from the Authentication Policies panel.

When Steel-Belted Radius receives an authentication request that does not contain EAP credentials, it passes the request to each enabled authentication method until one of the methods claims the request. The EAP settings in the eap.ini file come into play only when a request with EAP credentials is received. An authentication request contains EAP credentials if it includes one or more EAP-Message attributes and contains no other form of user credentials (for example, User-Password).

EAP-Only Setting

When an authentication method's EAP-Only setting is 1, Steel-Belted Radius prevents the authentication method from being called for any request that does not contain EAP credentials. Under this setting, the authentication method is also bypassed if an authentication request specifically requests an EAP protocol that is not listed in the authentication method's EAP-Type list in the eap.ini file.

Note: The PEAP authentication method plug-in converts the inner EAP/Generic Token credentials to PAP for security reasons. If you are using SecurID with PEAP, you should set the EAP-Only setting to 0.

First-Handle-Via-Auto-EAP Setting

If your configuration involves clients using more than one EAP protocol, Steel-Belted Radius must select an initial EAP protocol with which to proceed when receiving an authentication request with EAP credentials.

Selecting the incorrect EAP protocol is not fatal; the client simply sends an EAP NAK in response to the server's selected protocol and suggests an alternate one. After one additional network round-trip, the correct EAP protocol becomes active.

Depending on the capabilities of the authentication methods being used, you might be able to cut out this additional network round-trip that affects a portion of your EAP-based authentication requests.

If an authentication method can check for the existence of a user and can retrieve the user's password information with only the information available in the authentication request (for example, the username), it is said to be prefetch-capable. A prefetch-capable authentication method could be consulted first to see if a user exists in its database before committing to a specific EAP protocol.

If your authentication method is prefetch-capable, you would set First-Handle-Via-Auto-EAP to 0, indicating that the authentication method should have the first chance to handle the request. You would also set First-Handle-Via-Auto-EAP to 0 if the authentication method is capable of handling EAP credentials all on its own (clearly, it would not expect an automatic helper EAP method to do work on its behalf in this case).

By configuring the authentication method to be called first, Steel-Belted Radius can delay selection of an EAP protocol until it has ascertained whether the user exists in a particular authentication method's database. This is a useful technique when you plan to use more than one EAP protocol, but you do not know which one the client will want. Even in this scenario, automatic EAP helpers can still end up performing the EAP protocol processing; they will take over after the authentication method has retrieved a user's password information, rather than before.

The goal of an automatic EAP helper is to generate credentials against which traditional authentication methods (ones that do not understand EAP) can operate. Once an automatic EAP helper has generated these credentials, the authentication method that triggered the use of the helper is checked first for a password/credential match. Should this match not be present, the same traditional credentials are passed to all remaining enabled authentication methods in the master list (in the order in which they appear in the list).

Table 32: Authentication Method Prefetch Capability

Authentication Method	Prefetch Capable?
LDAP	Yes, if using BindName (rather than the Bind option)
Native User	Yes
SQL	Yes, if password does not need to be used as an input parameter in the SQL statement
UNIX User	No
Windows Domain	No

Note: If you enable the lockout facility in Steel-Belted Radius and you use a tunneled authentication method

(TTLS or PEAP) with a prefetch-capable method (native user, SQL, or LDAP) and an enabled EAP protocol (MS-CHAPv2, MD5-Challenge, LEAP, TLS), then you must enable First Handle via Auto- EAP in that prefetch-capable method to prevent the outer username (anonymous) from being added to the lockout list.

Otherwise, when Steel-Belted Radius receives an authentication request that uses an unconfigured EAP method, Steel-Belted Radius will reject the user (because the EAP method is not configured) and add the outer username (anonymous) to its lockout list. This will result in all users with an outer authentication name of anonymous being rejected until the lockout period expires.

EAP-NAK Notifications

If you are supporting only one type of client or only one EAP protocol, Steel-Belted Radius selects that EAP protocol for all EAP-based authentication requests it receives. If you are planning to support multiple EAP protocols and do not intend to maintain databases that track the appropriate EAP protocol on a user-by-user basis, Steel-Belted Radius automatically selects the appropriate EAP protocol for you.

When multiple EAP protocols are in play, you should configure each authentication method you plan to use with all the EAP protocols that can be used with it. In this configuration, when Steel-Belted Radius receives an authentication request containing EAP information, it chooses the first EAP protocol listed for the first authentication method that claims the request. Should the client require a different EAP protocol, it sends back an EAP-NAK that specifies the EAP protocol it would prefer to use.

After receiving an EAP-NAK, Steel-Belted Radius performs a scan of the authentication methods to find the first authentication method that has the requested EAP protocol listed (the authentication method might support this EAP protocol directly or with the help of an automatic EAP helper).

If the requested EAP protocol does not appear in any of the authentication methods’ lists of supported EAP protocols, Steel-Belted Radius rejects the authentication request.

Reauthenticating Connections

Most Access Points understand only a limited number of attributes that can be included in a RADIUS response to signal that the user has been accepted. The Session-Timeout attribute is of particular significance in a WLAN realm as it instructs the Access Point how long to allow the user to remain connected to a WLAN before having to re-authenticate to Steel-Belted Radius.

You can configure your choice of Session-Timeout settings using standard Steel-Belted Radius reply-list items on a user-by-user basis. If you are using EAP-TLS or EAP-TTLS to authenticate users, you can also have these modules automatically generate Session-Timeout attributes based on policies set in their configuration files. This level of control is necessary for EAP-TLS and EAP-TTLS as these modules also support session resumption, a quicker method of re-authenticating users. The value in the Session-Timeout attribute might need to be dynamically calculated in these cases.

Note: Not all Access Points support the Session-Timeout attribute. You should check your Access Points’ specifications to determine whether this configuration must be performed in a fixed manner on the Access Point or if the Access Point should defer to the server.

Certificates

A certificate is an electronic data structure used to identify an individual, a server, a company, or some other entity, and to associate that identity with a public key and an associated private key. Like a passport, a certificate provides generally recognized proof of an entity’s identity. Certificates bind public key values to entities, so that remote users of an entity's public key can be certain the associated private key is owned by the correct person or system. Certificates help prevent the use of fake public keys for impersonation. Only the public key certified by the certificate will work with the corresponding private key possessed by the entity identified by the certificate. The most widely accepted format for certificates is defined by the ITU-T X.509 international standard, which is described in RFC 3280, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile."

Certificate authorities (CAs) are entities that validate identities and issue certificates. An organization that wants to serve as its own CA can issue its own certificates, or an organization can purchase certificates from a trusted third-party CA. The methods used to validate an identity vary depending on the policies of a given CA. In general, before issuing a certificate, a CA must verify the identity of the entity and must digitally sign the certificate to ensure it cannot be modified. This ensures that a certificate issued by a CA binds a particular public key to the name of the entity the certificate identifies (such as the name of an employee).

In addition to a public key, a certificate includes the name of the entity it identifies, an expiration date, the name and URI of the CA that issued the certificate, a serial number, and the digital signature of the issuing CA, which creates a mathematical relationship between the signing CA certificate's public key and the public key of the certificate it signs. The CA's digital signature allows the certificate to function as a "letter of introduction" for users who know and trust the CA but don't know the entity identified by the certificate.

Because a certificate's expiration date is part of its signed contents, remote entities can verify that a certificate is valid and current.

Common types of certificates include the following:

- Certificate Authority certificates can sign other certificates.
- Server certificates are used on a server to enable a software client to verify the validity of the connection to a machine ("Am I really connecting to www.pulsesecure.net?") and to create an encrypted channel between a client and a server.
- Client certificates are used to allow a server to verify a client's identity (certificate based authentication) and to allow a user to digitally sign or encrypt data. Client certificates, which is digitally signed by a trusted certificate authority, is stronger proof of a client's identity than username/ password credentials alone.

Certificate Chains

A certificate chain is a sequence of certificates, where each certificate in the chain is signed by the certificate above it in the chain. At the top of the chain is a self-signed certificate. Each CA in the chain vouches for the identity in the entity to which it issues a signed digital certificate. Certificate chains establish a chain of trust; if you trust the CA at the top of the chain, this implies you can trust the signed certificates below it in the chain.

Certificate Revocation Lists

Under normal circumstances, a certificate remains valid until it reaches its expiration date. However, a certificate might become invalid before it expires. For example, if an employee whose identity is bound to a certificate terminates employment or if an enterprise suspects the confidentiality of the private key associated with a certificate's public key has been compromised, the certificate might be declared invalid and revoked before its expiration date.

When a CA revokes a certificate, it must let other entities know the certificate is no longer valid and should not be accepted. A certificate revocation list (CRL) is a signed data structure that identifies the serial numbers of certificates that have been issued and subsequently revoked by the CA. When a remote entity is asked to use a certificate to verify a remote user's identity, it can download a current copy of the applicable CRL from a CRL distribution point (CDP) and confirm that the certificate's serial number is not present. If a CRL has expired, the entity must connect to the CDP to download a new revocation list.

CRLs can be issued by a CA periodically (hourly, daily, or weekly) or as needed. When a certificate is revoked, its serial number is listed in the CRL, and that serial number remains in the CRL at least one period after the certificate's expiration date. CRLs, like certificates, can be distributed by untrusted servers and untrusted communications.

Under some circumstances, latency (the time between when a certificate is revoked and when the certificate's serial number appears on the CRL of the issuing CA) might be a concern. For example, if a revocation is reported today, that revocation will not be reliably notified to certificate-using systems until all currently issued CRLs are updated, which might take hours, days, or even weeks. Online revocation checking can reduce the latency between a revocation report and the distribution of the information to relying parties.

If CRL checking is enabled, Steel-Belted Radius uses the URI information contained in a client certificate to connect to the certificate's CDP. Steel-Belted Radius then uses HTTP, LDAP, or a network file system to retrieve the appropriate CRLs. Steel-Belted Radius stores these retrieved CRLs in the CRLCache directory under the radiusdir server directory.

When a client certificate is presented during EAP-TLS or EAP-TTLS authentication, Steel-Belted Radius can evaluate the client's certificate chain against its set of stored CRLs to verify none of the certificates in the chain have been revoked.

You can configure the following settings for CRL checking:

- Static CDPs—A static CDP is a CDP whose address (URI) is specified in the [Static_CDPs] section of a TLS or TTLS initialization file.
- CRL expiration—The CRL checking feature can be configured to operate in strict or lax mode.
    - In strict mode, a cached CRL that has expired will be immediately discarded; if Steel-Belted Radius cannot acquire a new CRL in the allotted time during a CRL check on a chain, the user is rejected.
    - In lax mode, you can configure Steel-Belted Radius to accept an expired CRL for a a period past its expiration. Note that Steel-Belted Radius attempts to obtain a current CRL whether it is running in strict or lax mode.
- Missing CDP attribute—When a CRL check is performed on a certificate chain, Steel-Belted Radius reads the contents of the CDP attribute for each certificate past the root certificate and uses the CDP information to retrieve the appropriate CRL. If a non-root certificate in the chain does not contain a CDP attribute, no CRL checking will be performed for that certificate. You can configure EAP-TLS to reject the user if it encounters a non-root certificate that is missing a CDP attribute.
- Incomplete LDAP CDP—Some CAs can create certificates that contain an LDAP-style CDP (//ldap:\\...) that does not specify the identity of the LDAP server to be queried. You can designate a default LDAP server that will be used when such CDPs are encountered. If you do not designate a default LDAP server and an LDAP-style CDP is encountered, the CRL retrieval will fail.
- HTTP proxies for CRL checking—Network security policy may prevent Steel-Belted Radius from making a direct HTTP connection to a CDP. In such cases, you can configure Steel-Belted Radius to

download CRLs through an HTTP proxy server on its local network. Optionally, you can specify the hosts or domains that do not require an HTTP proxy.

- CRL cache flushing—You can flush the CRL caches used for EAP-TLS and EAP-TTLS authentication at any time.

EAP-TLS

The EAP-TLS (Transport Layer Security) protocol requires that both user and authentication server have certificates for mutual authentication. While the mechanism is very strong, it requires that the corporation that deploys it maintain a certificate infrastructure for all of its users.

EAP-TLS can be deployed as an authentication method or as an automatic EAP helper.

- When EAP-TLS is deployed as an authentication method, EAP-TLS appears in the Authentication Policies panel in SBR Administrator after it is deployed as an authentication method. You can use the Authentication Policies panel to enable the EAP-TLS method and specify its sequence relative to other authentication methods Steel-Belted Radius uses.

  When EAP-TLS is deployed as an authentication method, you can configure it to perform certificate revocation list (CRL) checking. When CRL checking is enabled, EAP-TLS confirms that the client's certificate chain traces back to one of the trusted root certificates installed at initialization and checks the serial number of each certificate in the chain against the contents of CRLs to verify that none of the certificates in the chain have been revoked.

  You can configure the tlsauth.aut file to call a fixed profile when TLS-EAP is used. This profile specifies the attributes that are sent back in response to a successful authentication.

  You cannot use secondary authorization when EAP-TLS is deployed as an authentication method.

- When EAP-TLS is deployed as an automatic EAP helper, you must list TLS in the EAP-Type list of an authentication method. When EAP-TLS is triggered, the tlsauth authentication goes through the TLS handshake required by the EAP-TLS specification. Assuming the user provides a certificate that the server can verify against a list of trusted root certificates, the EAP-TLS part of the exchange concludes successfully.

  You might not want to grant access to your network to every user with a trusted certificate. By enabling the optional secondary authorization feature of the tlsauth plugin, you can have Steel-Belted Radius authorize users with valid certificates on a case-by-case basis. Secondary authorization also allows you to include user-specific attributes in an Access-Accept response; these attributes can be used to communicate options that are to be active for a user's connection to the NAD. Without secondary authorization, the only attributes returned on an Access-Accept are those generated by the tlsauth plug-in itself (termination-action and session-limit).

  If you enable the TLS authentication method, secondary authorizations must be performed by local authentication methods (they cannot be proxied). The authentication method you select for secondary authorizations must be able to authenticate users in a single pass; it cannot challenge the authorization request and request additional information. The username employed during secondary authorization is derived from a field in the user's certificate. Since a user's certificate does not include a password, you must configure tlsauth to make the secondary authorization request with no password or with a fixed password.

  If you configure secondary authorization with no password, your selected authentication method must be capable of handling requests that do not include passwords; the only authentication methods that support this style of authentication and ship with Steel-Belted Radius are Native User, LDAP and SQL. If you configure secondary authorization with a fixed password, you can use any authentication method that supports PAP authentication. In this configuration all user records must have the same fixed password.

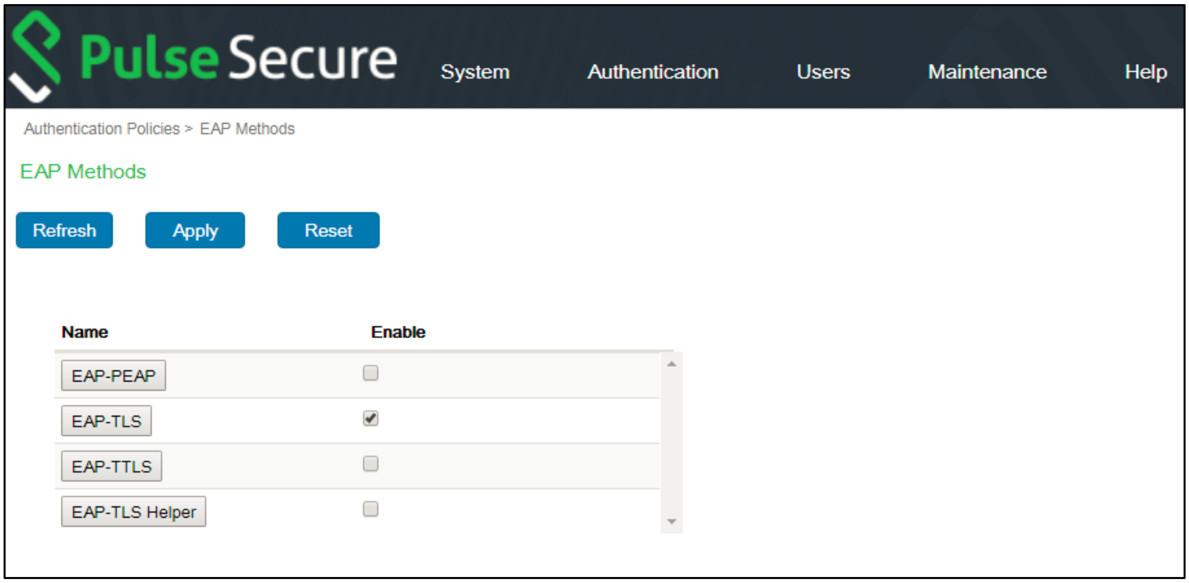
Configuring EAP-TLS as an Authentication Method

Note: You must configure the server certificate for the Steel-Belted Radius server before you use the EAP-TLS authentication method. For information on configuring your server certificate, see "Configuring Server Certificates".

To configure EAP-TLS as an authentication method:

1. Select Authentication > Authentication Policies > EAP Methods to open the EAP Methods page.

Figure 146: EAP Methods Page

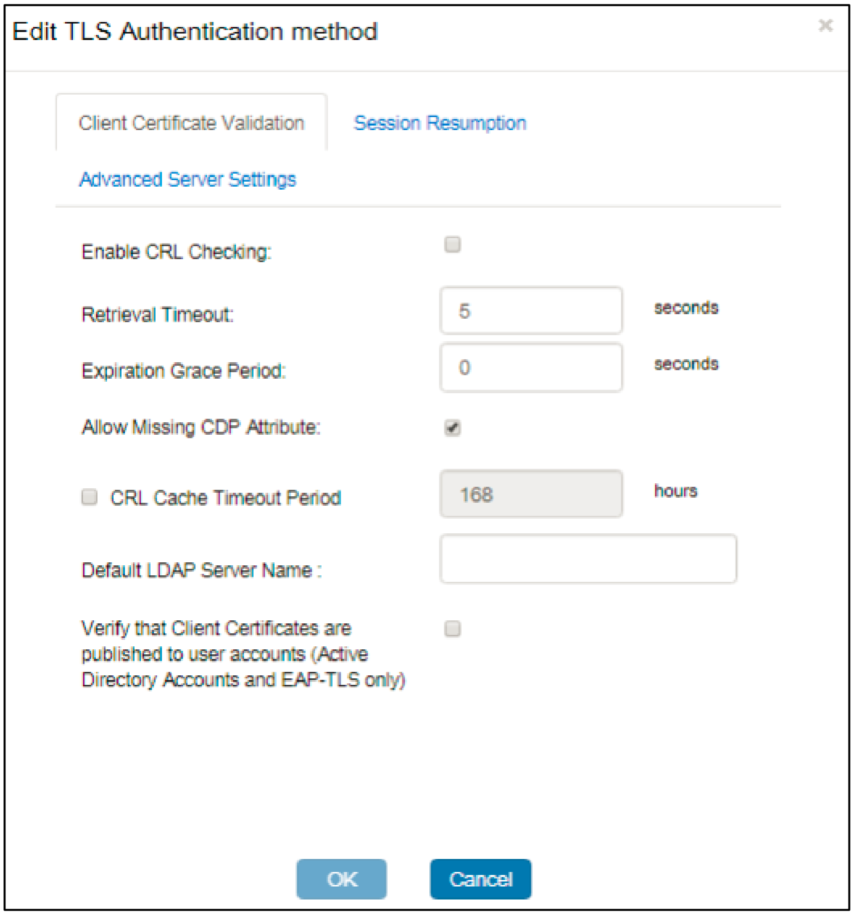


Name	Enable
EAP-PEAP	☐
EAP-TLS	☑
EAP-TTLS	☐
EAP-TLS Helper	☐

2. Click the **Enable** check box for the EAP-TLS authentication method.
3. Click the **EAP-TLS** button to edit the **EAP-TLS.**

   The Edit TLS Authentication Method page opens.

Figure 147: Edit TLS Authentication Method Page



Edit TLS Authentication method

Enable CRL Checking: □

Retrieval Timeout: 5 seconds

Expiration Grace Period: 0 seconds

Allow Missing CDP Attribute: ☑

□ CRL Cache Timeout Period 168 hours

Default LDAP Server Name :

Verify that Client Certificates are published to user accounts (Active Directory Accounts and EAP-TLS only) □

4. Use the tabs in the Edit TLS Authentication Method page to configure the following settings:

   - Client certificate validation
   - Session resumption
   - Advanced server settings

   Each configuration task is described separately below.

Configuring Client Certificate Validation

Client certificate validation settings let you specify how Steel-Belted Radius performs certificate revocation list (CRL) checking.

To configure session resumption for the EAP-TLS protocol:

1. Click the **Client Certificate Validation** tab in the Edit TLS Authentication Method page.

2. Click the **Enable CRL Checking** check box to enable CRL checking.

3. Enter the number of seconds that EAP-TLS will wait for a CRL checking transaction to complete when the CRL check involves a CRL retrieval in the **Retrieval Timeout** field.

   When CRL retrieval takes longer than the specified time, the user’s authentication request results in a reject.

4. Enter the number of seconds during which a CRL is still considered acceptable after it has expired in the **Expiration Grace Period** field.

   EAP-TLS always attempts to retrieve a new CRL when it is presented with a certificate chain and it finds an expired CRL in its cache.

   - If you enter 0 (strict expiration mode), EAP-TLS does not accept a CRL that has expired.
   - If you enter a value greater than 0 (lax expiration mode), EAP-TLS considers the expired CRL as an acceptable stand-in from the time the CRL expires to the time the grace period ends.

5. Click the Allow Missing CDP Attribute check box if you want Steel-Belted Radius to accept a non-root certificate that does not have a CDP attribute.

   Without a CDP attribute, EAP-TLS will not know how to retrieve a CRL and will not be able to perform a revocation check on the certificate.

   If you click the Allow Missing CDP Attribute check box, EAP-TLS allows such certificates and skips CRL checking for them.

   If you clear the Allow Missing CDP Attribute check box, EAP-TLS does not accept a CRL with a missing CDP attribute.

6. If you want to specify a CRL cache timeout period, click the CRL Cache Timeout Period check box and enter the number of hours in the timeout period in the hours field.

   - If you do not enable this setting, the CRL will be refreshed whenever it expires.
   - If you enable this setting and enter 0, Steel-Belted Radius always regards the CRL in the cache as expired and downloads a new CRL every time it receives a client certificate request.
   - If you enable this setting and enter a number greater than 0, the CRL begins to expire when the age of the CRL in the cache exceeds the number of hours specified in this field or when the scheduled CRL expiration time occurs, whichever comes first.

   After a CRL has expired (because its scheduled expiration time has passed or because the CRL cache has timed out), Steel-Belted Radius uses the expiration grace period to determine whether it should use the current CRL.

7. Enter the name of the LDAP server to use if the CDP contains a value that begins with the string //ldap:\\\ in the Default LDAP Server Name field.

   CDPs generated by some CAs do not include the identity of the LDAP server. If you expect to encounter certificates with this style CDP, specify the name of the LDAP server that contains the CRLs.

   If you don’t specify a server name and such certificates are encountered, the CRL retrieval fails.

8. If your enterprise uses Active Directory and you want client certificates published to user accounts, click the Verify that Client Certificates are published to user accounts check box.

Configuring Session Resumption

Session resumption settings let you specify whether session resumption is permitted and under what circumstances session resumption is performed.

Note: For session resumption to work, the network access device must be configured to handle the Session-Timeout return list attribute, because the network access device must be able to tell the client to reauthenticate after the session timer has expired.

To configure session resumption for the EAP-TLS protocol:

1. Click the **Session Resumption** tab in the Edit TLS Authentication Method page.
2. Enter the maximum number of seconds you want the client to remain connected to the network access device before having to re-authenticate in the **Session TImeout** field.

   If you enter a number greater than 0, the lesser of this value and the remaining resumption limit is sent in a Session-Limit attribute to the RADIUS client on the RADIUS Access-Accept response.

   If you enter 0, no Session-Limit attribute is generated. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

   Entering a value such as 600 (10 minutes) does not necessarily cause a full re-authentication to occur every 10 minutes. You can configure the resumption limit to make most re-authentications fast and computationallycheap.

3. Enter the value that you want returned in a Termination-Action attribute in the Termination Action field.

   The Termination-Action attribute is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached. Valid values are:

   - -1: Do not send the attribute.
   - 0: Send the Termination-Action attribute with a value of 0.
   - 1: Send the Termination-Action attribute with a value of 1.

   Default value is -1. Note that this does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

4. Enter the maximum number of seconds you want the client to be able to re-authenticate using the TLS session resumption feature in the Resumption Limit field.

   This type of re-authentication is fast and computationally cheap. It does, however, depend on previous authentications and might not be considered as secure as a complete (computationally expensive) authentication. Specifying a value of 0 disables the session resumption feature.

ConfiguringAdvancedServerSettings

Advanced server settings specify the manner in which the inner authentication step operates. To configure advanced server settings for the EAP-TLS protocol:

1. Click the Advanced Server Settings tab in the Edit TLS Authentication Method page.
2. Enter the maximum length of the TLS message that might be generated during each iteration of the TLS exchange. in the **TLS Message Fragment Length** field.

Enter a number in the range 500–4096. This value affects the number of RADIUS challenge/response round trips required to conclude the TLS exchange. A value of 1400 might result in 6 round-trips, while a value of 500 might result in 15 round trips.

Some Access Points might have problems with RADIUS responses or EAP messages that exceed the size of one Ethernet frame (1500 bytes including IP/UDP headers).

The default length for TLS messages is 1020 bytes, which prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame.

3. Enable the **Return MPPE Keys** check box to specify whether the TLS authentication method includes RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Access-Accept response sent to the Access Point.

   You should enable this option if the Access Point needs to key the WEP encryption. If the Access Point is authenticating only end-users and WEP is not being used, you can clear this check box.

4. Use the **DH Prime Bits** list to specify the number of bits in the prime number that the module uses for Diffie-Hellman exponentiation.
   Selecting a longer prime number makes the system less susceptible to certain types of attacks but requires more CPU processing to compute the Diffie-Hellman key agreement operation.

   Valid values are 512, 1024, 1536, 2048, 3072, and 4096 bits.

5. Enter the TLS cipher suites (in order of preference) that the server is to use in the **Cipher Suites** field.

   These cipher suites are documented in RFC 2246, "The TLS Protocol Version 1.2," and other TLS-related RFCs and draft RFCs.

   Default value is 0x3C, 0x3D, 0x67, 0x6B, 0x40, 0x6A, 0x9C, 0x9D, 0x9E, 0x9F, 0xA2, 0xA3.

6. Optionally, enable the **Verify User Name is Principal** check box if you want Steel-Belted Radius to verify that the contents of the RADIUS User-Name attribute match the Principal Name of the certificate used to authenticate the user.

   Certificates issued by Microsoft's Windows 2000 Certificate Server typically include a Subject Alternative Name/Other Name attribute, where Principal Name set to something like **user@certtest.acme.com.**

   The Windows XP client that supports EAP-TLS in conjunction with 802.1X extracts this attribute value from the client's certificate and uses it to respond to the Access Point's EAP Identity Request. The Access Point, in turn, packages up this value as the RADIUS User-Name attribute in requests it sends to a RADIUS server.

   You should clear (disable) the check box if the certificates used do not include a Principal Name or if the client being used does not report the contents of Principal Name as the user's identity in response to an EAP Identity Request.

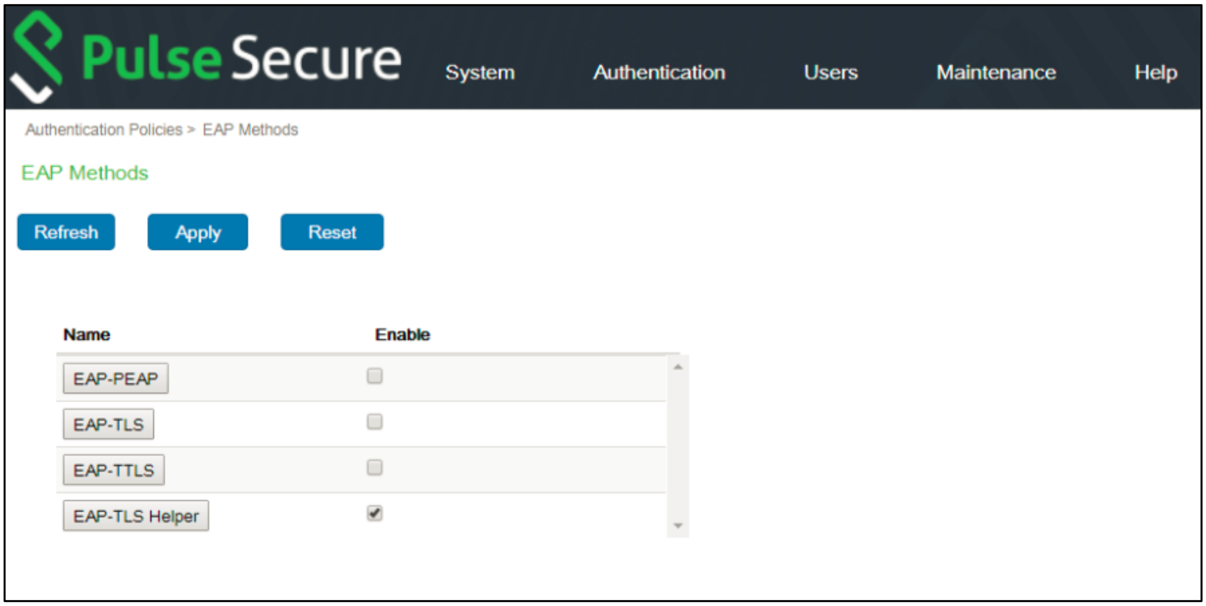
Configuring EAP-TLS as an Automatic EAP Helper

ℹ Note: You must configure the server certificate for the Steel-Belted Radius server before you use the TLS EAP helper. For information on configuring your server certificate, see "Configuring Server Certificates".

To configure EAP-TLS as an EAP helper:

1. Choose Authentication > Authentication Policies > EAP Methods from the menu bar.

Figure 148: EAP Methods Page



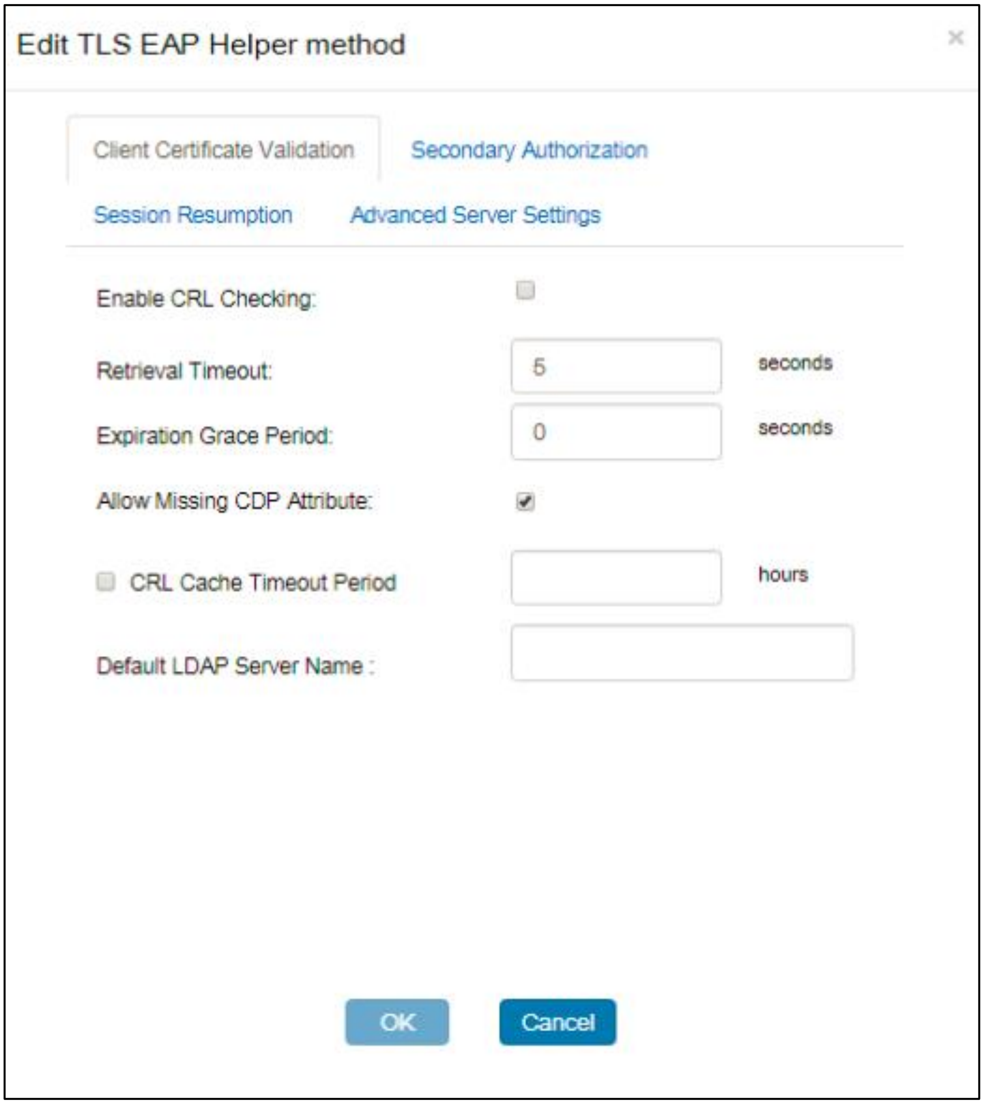
Name	Enable
EAP-PEAP	□
EAP-TLS	□
EAP-TTLS	□
EAP-TLS Helper	☑

2. Click the Enable check box for the EAP-TLS Helper method.

3. Click the EAP-TLS Helper button to edit the EAP-TLS Helper.

   The Edit TLS EAP Helper Method page opens.

Figure 149: Edit TLS EAP Helper Method Page



Enable CRL Checking: □

Retrieval Timeout: 5 seconds

Expiration Grace Period: 0 seconds

Allow Missing CDP Attribute: ☑

□ CRL Cache Timeout Period hours

Default LDAP Server Name :

4. Use the tabs in the Edit TLS EAP Helper Method page to configure the following settings:

   - Client certificate validation
   - Secondary authorization
   - Session resumption
   - Advanced server settings

Each configuration task is described separately below.

Configuring Client Certificate Validation

Client certificate validation settings let you specify how Steel-Belted Radius performs certificate revocation list (CRL) checking.

To configure session resumption for the the TLS EAP helper protocol:

1. Click the **Client Certificate Validation** tab in the Edit TLS EAP Helper Method page.
2. Click the **Enable CRL Checking** check box to enable CRL checking.
3. Enter the number of seconds that the TLS EAP helper will wait for a CRL checking transaction to complete when the CRL check involves a CRL retrieval in the **Retrieval Timeout** field.

   When CRL retrieval takes longer than the specified time, the user's authentication request results in a reject.
4. Enter the number of seconds during which a CRL is still considered acceptable after it has expired in the **Expiration Grace Period** field.

   The TLS EAP helper always attempts to retrieve a new CRL when it is presented with a certificate chain and it finds an expired CRL in its cache.

   - If you enter 0 (strict expiration mode), the TLS EAP helper does not accept a CRL that has expired.
   - If you enter a value greater than 0 (lax expiration mode), the TLS EAP helper considers the expired CRL as an acceptable stand-in from the time the CRL expires to the time the grace period ends.
5. Click the **Allow Missing CDP Attribute** check box if you want Steel-Belted Radius to accept a non-root certificate that does not have a CDP attribute.

   Without a CDP attribute, the TLS EAP helper will not know how to retrieve a CRL and will not be able to perform a revocation check on the certificate.

   If you click the **Allow Missing CDP Attribute** check box, the TLS EAP helper allows such certificates and skips CRL checking for them.

   If you clear the **Allow Missing CDP Attribute** check box, the TLS EAP helper does not accept a CRL with a missing CDP attribute.
6. If you want to specify a CRL cache timeout period, click the **CRL Cache Timeout Period** check box and enter the number of hours in the timeout period in the **hours** field.

   - If you do not enable this setting, the CRL will be refreshed whenever it expires.
   - If you enable this setting and enter 0, Steel-Belted Radius always regards the CRL in the cache as expired and downloads a new CRL every time it receives a client certificate request.
   - If you enable this setting and enter a number greater than 0, the CRL begins to expire when the age of the CRL in the cache exceeds the number of hours specified in this field or when the scheduled CRL expiration time occurs, whichever comes first.

   After a CRL has expired (because its scheduled expiration time has passed or because the CRL cache has timed out), Steel-Belted Radius uses the expiration grace period to determine whether it should use the current CRL.
7. Enter the name of the LDAP server to use if the CDP contains a value that begins with the string //ldap:\\\ in the **Default LDAP Server Name** field.

CDPs generated by some CAs do not include the identity of the LDAP server. If you expect to encounter certificates with this style CDP, specify the name of the LDAP server that contains the CRLs.

If you don't specify a server name and such certificates are encountered, the CRL retrieval fails.

Configuring Secondary Authentication

Secondary authorization settings let you specify whether secondary authorization is performed and, if it is, what information is used in the secondary authorization request.

To configure session resumption for the TLS EAP helper protocol:

1. Click the **Secondary Authorization** tab in the Edit TLS EAP Helper Method page.
2. Click the **Enable Secondary Authorization** check box to enable secondary authorization checking.

   If secondary authorization is disabled, the EAP-TLS plug-in accepts the user upon proof of ownership of a private key that matches a valid certificate.

   If secondary authorization is enabled, a secondary authorization check against a traditional authentication method such as an SQL plug-in is performed.
3. Specify whether you want user names to be converted to Subject CN names or principal names.

   After the EAP-TLS module has concluded its processing, it might still defer to a traditional authentication method (core or plug-in) for final authorization. To do so, it must provide a user name and password to the traditional authentication method.

   - If you click the **Subject CN** option button, the EAP-TLS module parses the Subject attribute of the client's certificate for the least significant 'CN=' and takes the value of this attribute (for example, 'George Washington') as the user name being passed to the traditional authentication method.
   - If you click the **Principal Name** option button, the EAP-TLS module uses the principal name (Subject Alternate Name or Other Name) from the client certificate (for example, joe@acme.com) as the user name being passed to the traditional authentication method.
4. If you plan to use secondary authorization against an authentication method (for example, LDAP) that cannot be configured to ignore the lack of user credentials, specify a fixed password that the plug-in uses on all secondary authorization checks in the **Fixed Password** field.

   By default, the secondary authorization check includes a user name but no other user credentials, because no password or similar credential for the client is available at the conclusion of the TLS handshake. Some authentication methods (Native User, LDAP, and SQL) can be configured to not require user credentials.
5. If you want the EAP-TLS plug-in to add four attributes to the request before the secondary authorization check is performed, click the Include Certificate Info check box.

   When the Include Certificate Info check box is clicked, Steel-Belted Radius adds the following attributes to the request:

   - The Funk-Peer-Cert-Subject attribute contains the value of the Subject attribute in the client certificate.

- The Funk-Peer-Cert-Principal attribute contains the value of the principal name (Subject Alternate Name or Other Name) attribute of the client certificate.
- The Funk-Peer-Cert-Issuer attribute contains the value of the Issuer attribute in the client certificate.
- The Funk-Peer-Cert-Hash attribute contains a hexadecimal ASCII representation of the SHA1 hash of the client certificate.

These attributes are ignored if the authentication method that performs the authentication check does not use them.

Configuring Session Resumption

Session resumption settings let you specify whether session resumption is permitted and under what circumstances session resumption is performed.

ⓘ Note: For session resumption to work, the network access device must be configured to handle the Session-Timeout return list attribute, because the network access device must be able to tell the client to reauthenticate after the session timer has expired.

To configure session resumption for the TLS EAP helper protocol:

1. Click the **Session Resumption** tab in the Edit TLS EAP Helper Method page.
2. Enter the maximum number of seconds you want the client to remain connected to the network access device before having to re-authenticate in the **Session TImeout** field.

   If you enter a number greater than 0, the lesser of this value and the remaining resumption limit is sent in a Session-Limit attribute to the RADIUS client on the RADIUS Access-Accept response.

   If you enter 0, no Session-Limit attribute is generated. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

   Entering a value such as 600 (10 minutes) does not necessarily cause a full re-authentication to occur every 10 minutes. You can configure the resumption limit to make most re-authentications fast and computationallycheap.

3. Enter the value that you want returned in a Termination-Action attribute in the Termination Action field.

   The Termination-Action attribute is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached. Valid values are:

   - -1: Do not send the attribute.
   - 0: Send the Termination-Action attribute with a value of 0.
   - 1: Send the Termination-Action attribute with a value of 1.

   Default value is -1. Note that this does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

4. Enter the maximum number of seconds you want the client to be able to re-authenticate using the

TLS session resumption feature in the **Resumption Limit** field.

This type of re-authentication is fast and computationally cheap. It does, however, depend on previous authentications and may not be considered as secure as a complete (computationally expensive) authentication. Specifying a value of 0 disables the session resumption feature.

Configuring Advanced Server Settings

Advanced server settings specify the manner in which the inner authentication step operates. To configure advanced server settings for the TLS EAP helper protocol:

1. Click the **Advanced Server Settings** tab in the Edit TLS EAP Helper Method page.
2. Enter the maximum length of the TLS message that may be generated during each iteration of the TLS exchange in the **TLS Message Fragment Length** field.

   Enter a number in the range 500–4096. This value affects the number of RADIUS challenge/response round-trips required to conclude the TLS exchange. A value of 1400 may result in 6 round-trips, while a value of 500 may result in 15 round-trips.

   Some Access Points may have problems with RADIUS responses or EAP messages that exceed the size of one Ethernet frame (1500 bytes including IP/UDP headers).

   The default length for TLS messages is 1020 bytes, which prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame.
3. Enter the maximum number of seconds you want to allow for the EAP authentication sequence in the **Max Transaction Time** field.

   If the EAP authentication sequence takes longer than the number of seconds specified in this field, Steel-Belted Radius terminates the user authentication.
4. Enable the **Return MPPE Keys** check box to specify whether the TLS EAP helper includes RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Access-Accept response sent to the Access Point.

   You should enable this option if the Access Point needs to key the WEP encryption. If the Access Point is authenticating only end-users and WEP is not being used, you can clear this check box.
5. Use the **DH Prime Bits** list to specify the number of bits in the prime number that the module uses for Diffie-Hellman exponentiation.

   Selecting a longer prime number makes the system less susceptible to certain types of attacks but requires more CPU processing to compute the Diffie-Hellman key agreement operation.

   Valid values are 512, 1024, 1536, 2048, 3072, and 4096 bits.
6. Enter the TLS cipher suites (in order of preference) that the server is to use in the **Cipher Suites** field.

   These cipher suites are documented in RFC 2246, "The TLS Protocol Version 1.2," and other TLS-related RFCs and draft RFCs.

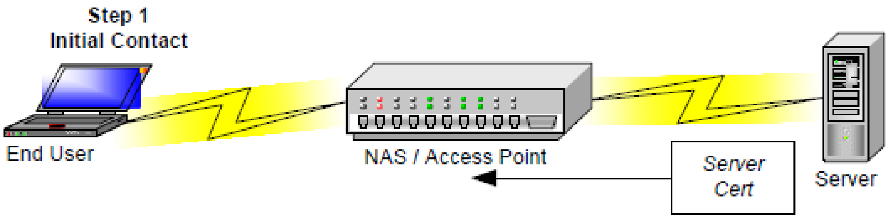
   Default value is 0x3C, 0x3D, 0x67, 0x6B, 0x40, 0x6A, 0x9C, 0x9D, 0x9E, 0x9F, 0xA2, 0xA3.

EAP-TTLS

EAP-TTLS (Tunneled Transport Layer Security) is designed to provide authentication that is as strong as EAP-TLS, but it does not require that each user be issued a certificate. Instead, only the authentication servers are issued certificates. User authentication is performed by password; but the password credentials are transported in a securely encrypted "tunnel" established based upon the server certificates.

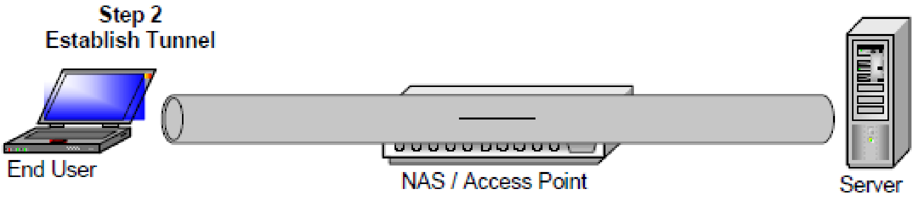
1. After the authentication server determines that the user has made an authentication request, it sends its certificate to the user's system.

Figure 150: Server Certificate Sent to RAS



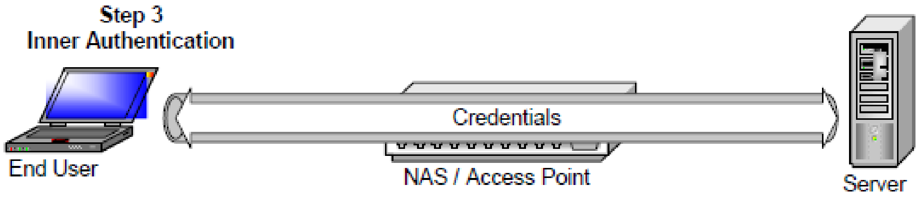
2. The authentication server's certificate is used to establish a tunnel between the user and the server.

Figure 151: Tunnel Established



3. Once the tunnel is established, credentials can be exchanged safely between the server and the user since tunnels encrypt all data in a secure fashion. This stage is called ***inner authentication.***

Figure 152: Inner Authentication



With EAP-TTLS, it is not necessary to create a new infrastructure of user certificates. User authentication is performed against the same security database that is already in use on the corporate LAN; for example, Windows Domain Controllers, SQL or LDAP databases, or token systems.

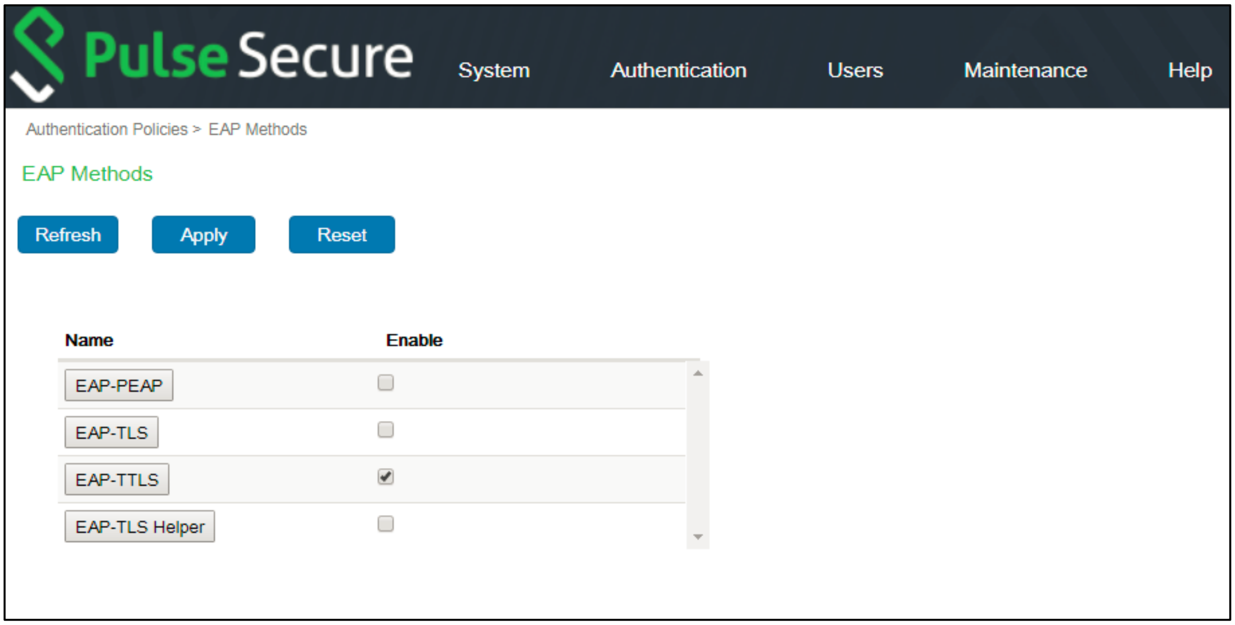
The routing of the inner authentication request can be handled either by means of standard Steel-Belted Radius authentication request routing or by means of a directed realm. If your EAP-TTLS tunnel ends at a dedicated server and all the inner authentication requests are to be performed by other servers, you should use standard request routing so the proxy realm target can be determined in a standard fashion (that is, the decoration of the username revealed by inner authentication). If your EAP-TTLS tunnel and inner authentication are handled by the same server, you can use a directed realm to specify which authentication method(s) handle the inner authentication.

Configuring EAP-TTLS

To configure EAP-TTLS as an authentication method:

1. Chosse Authentication > Authentication Policies > EAP Methods from the menu bar.

Figure 153: EAP Methods Page



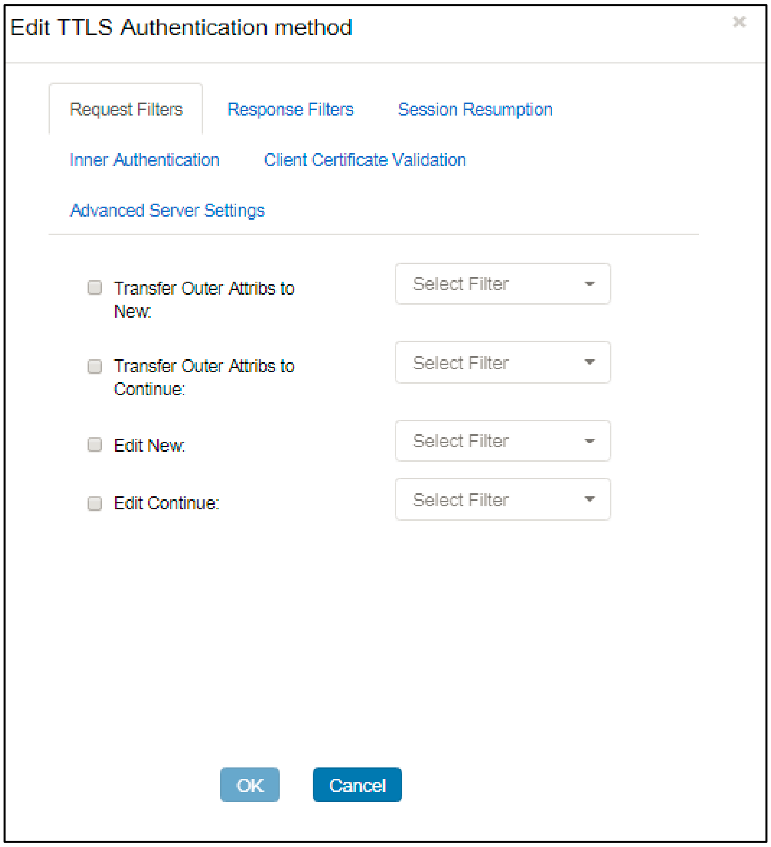
Name	Enable
EAP-PEAP	☐
EAP-TLS	☐
EAP-TTLS	☑
EAP-TLS Helper	☐

2. Click the Enable check box for the EAP-TTLS method.

3. Click the EAP-TTLS button to edit the EAP-TTLS.

   The Edit TTLS Authentication Method page opens.

Figure 154: Edit TTLS Authentication Method Page



□ Transfer Outer Attribs to New: Select Filter

□ Transfer Outer Attribs to Continue: Select Filter

□ Edit New: Select Filter

□ Edit Continue: Select Filter

4. Use the tabs in the Edit TTLS Authentication Method page to configure the following settings:

   - Request filters
   - Response filters
   - Client certificate validation
   - Session resumption
   - Inner authentication
   - Advanced server settings

   Each configuration task is described separately below.

Configuring Request Filters

Request filters affect the attributes of inner authentication requests. By default, Steel-Belted Radius does not use request filters.
To configure request filtering for the EAP-TTLS protocol:

1. Click the **Request Filters** tab in the Edit TTLS Authentication method page.

2. Optionally, click the **Transfer Outer Attribs to New** check box and select the filter you want to use from the dropdown list.

   This filter affects only a new inner authentication request (rather than continuations of previous requests).

- If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.
- If this filter is not specified, no attributes from the outer request are transferred to the inner request.

3. Optionally, click the **Transfer Outer Attribs to Continue** check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than the first inner authentication request). If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.

   If this filter is not specified, no attributes from the outer request are transferred to the inner request.

4. Optionally, click the **Edit New** check box and select the filter you want to use from the dropdown list.

   This filter affects only a new inner authentication request (rather than continuations of previous requests). If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (by the filter specified in Step 2) and attributes included in the inner authentication request sent through the tunnel by the client.

   If this filter is not specified, the request remains unaltered.

5. Optionally, click the **Edit Continue** check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than a new inner authentication request). If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (by the filter specified in Step 3) and attributes included in the inner authentication request sent through the tunnel by the client.

   If this filter is not specified, the request remains unaltered.

Configuring Response Filters

Response filters affect the attributes in the final response (Access-Accept or Access-Reject) returned to the originating NAD. By default, Steel-Belted Radius does not use response filters.

To configure response filtering for the EAP-TTLS protocol:

1. Click the **Response Filters** tab in the Edit TTLS Authentication method page.
2. Optionally, click the **Transfer Inner Attribs to Accept** check box and select the filter you want to use from the dropdown list.

   This filter affects only an outer Access-Accept response that is sent back to a network access device.

   - If this filter is specified, the filter is applied to the inner authentication response and all resulting attributes are transferred to the outer authentication response.

- If this filter is not specified, no inner authentication response attributes are transferred to the outer authentication response.

3. Optionally, click the **Transfer Inner Attribs to Reject** check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than the first inner authentication request). If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.

   If this filter is not specified, no attributes from the outer request are transferred to the inner request.

Configuring Client Certificate Validation

Client certificate validation settings let you specify how Steel-Belted Radius performs certificate revocation list (CRL) checking.

To configure session resumption for the EAP-TTLS protocol:

1. Click the **Client Certificate Validation** tab in the Edit TTLS Authentication Method page.

2. Click the **Enable CRL Checking** check box to enable CRL checking.

3. If you want to require that the client must provide a certificate as part of the TTLS exchange, click the **Require Client Certificate** check box.

4. Enter the number of seconds that EAP-TTLS will wait for a CRL checking transaction to complete when the CRL check involves a CRL retrieval in the **Retrieval Timeout** field.

   When CRL retrieval takes longer than the specified time, the user's authentication request results in a reject.

5. Enter the number of seconds during which a CRL is still considered acceptable after it has expired in the **Expiration Grace Period** field.

   EAP-TTLS always attempts to retrieve a new CRL when it is presented with a certificate chain and it finds an expired CRL in its cache.

   - If you enter 0 (strict expiration mode), EAP-TTLS does not accept a CRL that has expired.

   - If you enter a value greater than 0 (lax expiration mode), EAP-TTLS considers the expired CRL as an acceptable stand-in from the time the CRL expires to the time the grace period ends.

6. Click the **Allow Missing CDP Attribute** check box if you want Steel-Belted Radius to accept a non- root certificate that does not have a CDP attribute.

   Without a CDP attribute, EAP-TTLS will not know how to retrieve a CRL and will not be able to perform a revocation check on the certificate.

   - If you click the **Allow Missing CDP Attribute** check box, EAP-TTLS allows such certificates and skips CRL checking for them.

   - If you clear the **Allow Missing CDP Attribute** check box, EAP-TTLS does not accept a CRL with a missing CDP attribute.

7. If you want to specify a CRL cache timeout period, click the **CRL Cache Timeout** Period check box and enter the number of hours in the timeout period in the **hours** field.

    - If you do not enable this setting, the CRL will be refreshed whenever it expires.
    - If you enable this setting and enter 0, Steel-Belted Radius always regards the CRL in the cache as expired and downloads a new CRL every time it receives a client certificate request.
    - If you enable this setting and enter a number greater than 0, the CRL begins to expire when the age of the CRL in the cache exceeds the number of hours specified in this field or when the scheduled CRL expiration time occurs, whichever comes first.

    After a CRL has expired (because its scheduled expiration time has passed or because the CRL cache has timed out), Steel-Belted Radius uses the expiration grace period to determine whether it should use the current CRL.

8. Enter the name of the LDAP server to use if the CDP contains a value that begins with the string //ldap:\\\ in the Default LDAP Server Name field.

    CDPs generated by some CAs do not include the identity of the LDAP server. If you expect to encounter certificates with this style CDP, specify the name of the LDAP server that contains the CRLs.

    If you don't specify a server name and such certificates are encountered, the CRL retrieval fails.

Configuring Session Resumption

Session resumption settings let you specify whether session resumption is permitted and under what circumstances session resumption is performed.

ℹ Note: For session resumption to work, the network access device must be configured to handle the Session-Timeout return list attribute, because the network access device must be able to tell the sclient to reauthenticate after the session timer has expired.

To configure session resumption for the EAP-TTLS protocol:

1. Click the Session Resumption tab in the Edit TTLS Authentication Method page.

2. Enter the maximum number of seconds you want the client to remain connected to the network access device before having to re-authenticate in the **Session TImeout** field.

    - If you enter 0, no Session-Limit attribute is generated. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.
    - If you enter a number greater than 0, the lesser of this value and the remaining resumption limit is sent in a Session-Limit attribute to the RADIUS client on the RADIUS Access-Accept response.

    Entering a value such as 600 (10 minutes) does not necessarily cause a full re-authentication to occur every 10 minutes. You can configure the resumption limit to make most re-authentications fast and computationallycheap.

3. Enter the value that you want returned in a Termination-Action attribute in the **Termination Action** field.

The Termination-Action attribute is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached. Valid values are:

- -1: Do not send the attribute.
- 0: Send the Termination-Action attribute with a value of 0.
- 1: Send the Termination-Action attribute with a value of 1.

Default value is -1. Note that this does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

4. Enter the maximum number of seconds you want the client to be able to re-authenticate using the TLS session resumption feature in the **Resumption Limit** field.

   This type of re-authentication is fast and computationally cheap. It does, however, depend on previous authentications and may not be considered as secure as a complete (computationally expensive) authentication. Specifying a value of 0 disables the session resumption feature.

Configuring Inner Authentication Settings

The inner authentication settings let you specify the way in which the inner authentication step is to operate.

To configure inner authentication settings for the EAP-TTLS protocol:

1. Click the **Inner Authentication** tab in the Edit TTLS Authentication Method page.
2. If you want requests to be routed based on the methods listed in the directed realm, enter the name of a directed realm in the **Directed Realm** field.

   Omitting this setting causes the inner authentication request to be handled like any other request received from a network access device.

3. If you want requests to be processed by means of a realm selection script, enter the name of a script in the **Realm Selection Script** field.

Configuring Advanced Server Settings

Advanced server settings specify the manner in which the inner authentication step operates. To configure advanced server settings for the EAP-TTLS protocol:

1. Click the Advanced Server Settings tab in the Edit TTLS Authentication Method page.
2. Enter the maximum length of the TLS message that may be generated during each iteration of the TLS exchange. in the **TLS Message Fragment Length** field.

   Enter a number in the range 500–4096. This value affects the number of RADIUS challenge/response round-trips required to conclude the TLS exchange. A value of 1400 may result in 6 round-trips, while a value of 500 may result in 15 round-trips.

   Some Access Points may have problems with RADIUS responses or EAP messages that exceed the size of one Ethernet frame (1500 bytes including IP/UDP headers).

   The default length for TLS messages is 1020 bytes, which prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame.

3. Enter the maximum number of seconds you want to allow for the EAP authentication sequence in the **Max Transaction Time** field.

   If the EAP authentication sequence takes longer than the number of seconds specified in this field, Steel-Belted Radius terminates the user authentication.

4. Enable the **Return MPPE Keys** check box to specify whether the TTLS authentication method includes RADIUS MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Access-Accept response sent to the Access Point.

   You should enable this option if the Access Point needs to key the WEP encryption. If the Access Point is authenticating only end-users and WEP is not being used, you can clear this check box.

5. Use the DH Prime Bits list to specify the number of bits in the prime number that the module uses for Diffie-Hellman exponentiation.

   Selecting a longer prime number makes the system less susceptible to certain types of attacks but requires more CPU processing to compute the Diffie-Hellman key agreement operation.

   Valid values are 512, 1024, 1536, 2048, 3072, and 4096 bits.

6. Enter the TLS cipher suites (in order of preference) that the server is to use in the Cipher Suites field.

   These cipher suites are documented in RFC 2246, "The TLS Protocol Version 1.2," and other TLS-related RFCs and draft RFCs.

   Default value is 0x3C, 0x3D, 0x67, 0x6B, 0x40, 0x6A, 0x9C, 0x9D, 0x9E, 0x9F, 0xA2, 0xA3.

EAP-PEAP

The EAP-PEAP (Protected EAP) protocol is similar to EAP-TTLS. Unlike EAP-TTLS, which can tunnel any kind of authentication request (such as PAP or CHAP) and extended attributes, PEAP can tunnel only other EAP protocols inside its connection.

EAP-PEAP works in two phases:

- In Phase 1, the client authenticates the server and uses a TLS handshake to create an encrypted tunnel.
- In Phase 2, the server authenticates the user or machine credentials using an EAP authentication protocol. The EAP authentication is protected by the encrypted tunnel created in Phase 1. The authentication type negotiated during Phase 2 can be any valid EAP type, such as GTC (Generic Token Card) or MS-CHAPv2.

Microsoft's implementation of PEAP and Cisco's implementation of PEAP supports different methods of client authentication through the TLS tunnel.

- The Microsoft PEAP implementation requires MS-CHAP-V2 for client authentication.
- The Cisco PEAP implementation supports client authentication by EAP-Generic Token, which Cisco uses both for authenticating token cards and for authenticating users against Windows domain/ Active Directory accounts.

The Cisco PEAP implementation supports the ability to hide username identities until the TLS encrypted tunnel is established and authentication phase is complete. The Microsoft PEAP implementation sends the username in clear text in Phase 1 of PEAP authentication.

Steel-Belted Radius supports both Microsoft PEAP and Cisco PEAP.

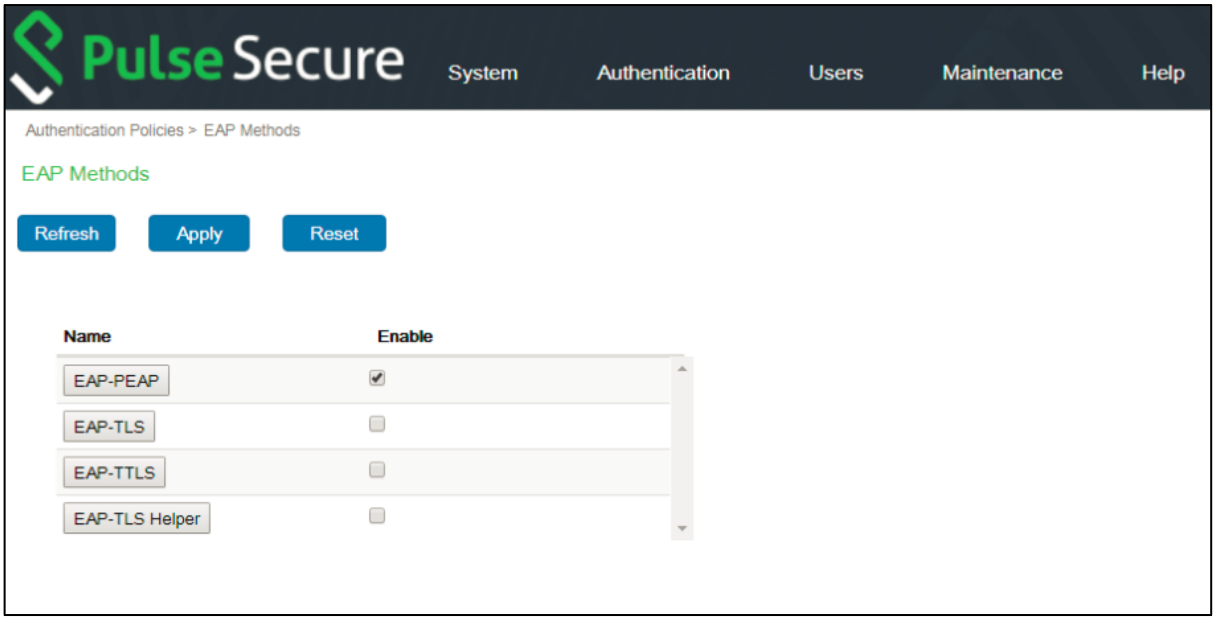
Configuring EAP-PEAP

Note: You must configure the server certificate for the Steel-Belted Radius server before you use the EAP-PEAP authentication method. For information on configuring your server certificate, see “Configuring Server Certificates”.

To configure EAP-PEAP on a Steel-Belted Radius server:

1. Choose Authentication > Authentication Policies > EAP Methods from the menu bar.

Figure 155: EAP Methods Page



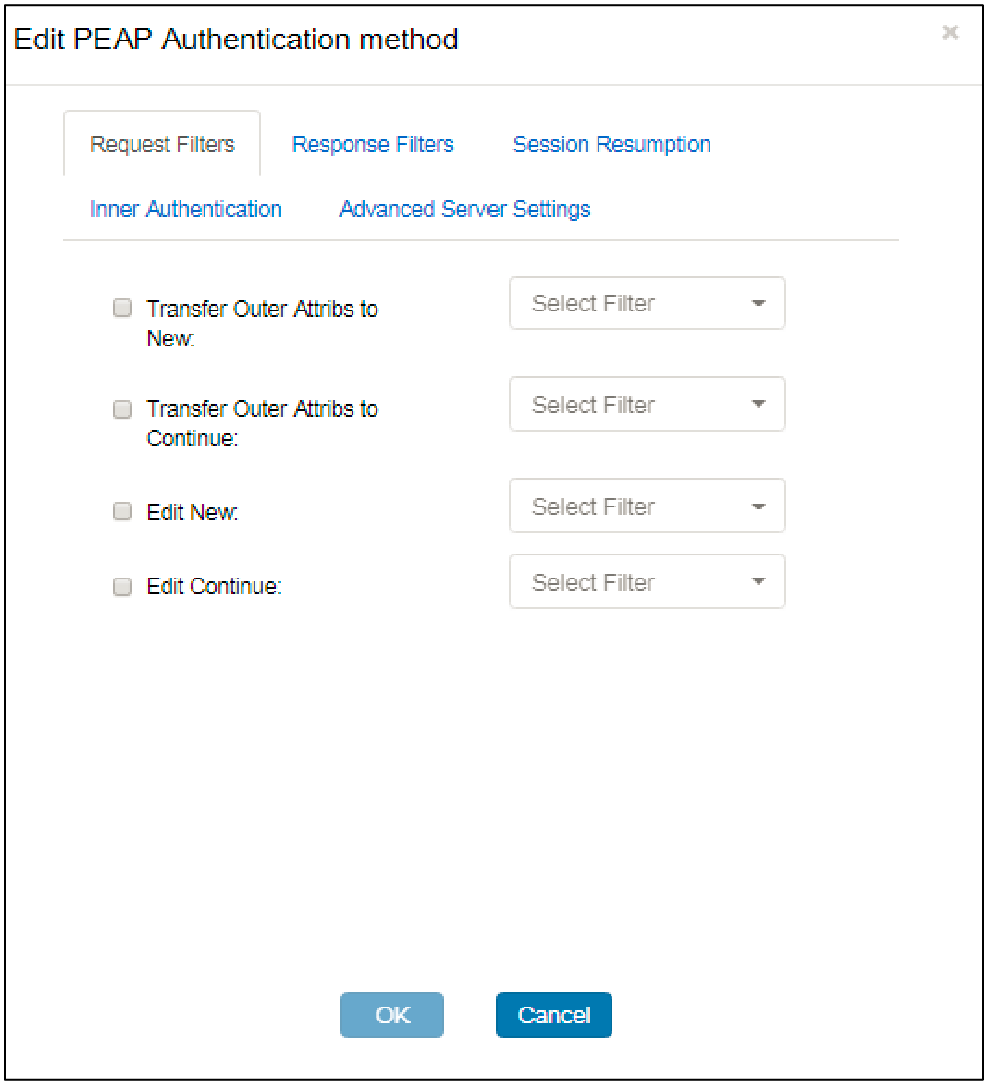
Name	Enable
EAP-PEAP	☑
EAP-TLS	□
EAP-TTLS	□
EAP-TLS Helper	□

2. Click the Enable check box for the EAP-PEAP authentication method.

3. Click the EAP-PEAP button to edit the EAP-PEAP.

   The Edit PEAP Authentication Method page opens.

Figure 156: Edit PEAP Authentication Method Page



Edit PEAP Authentication method

□ Transfer Outer Attribs to New: Select Filter

□ Transfer Outer Attribs to Continue: Select Filter

□ Edit New: Select Filter

□ Edit Continue: Select Filter

4. Use the tabs in the Edit PEAP Authentication Method dialog to configure the following settings:

    - Request filters
    - Response filters
    - Session resumption
    - Inner authentication
    - Advanced server settings

    Each configuration task is described separately below.

Configuring Request Filters

Request filters affect the attributes of inner authentication requests. By default, Steel-Belted Radius does not use request filters.

Note: You must configure filters using the Filters page before you can associate them with the EAP-PEAP authentication method. For information on configuring filters, refer to “Setting Up Filters via WebGUI”.

To configure request filtering for the EAP-PEAP protocol:

1. Click the **Request Filters** tab in the Edit PEAP Authentication method page.
2. Optionally, click the **Transfer Outer Attribs to New check** box and select the filter you want to use from the dropdown list.

   This filter affects only a new inner authentication request (rather than continuations of previous requests).

   - If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.
   - If this filter is not specified, no attributes from the outer request are transferred to the inner request.
3. Optionally, click the Transfer Outer Attribs to Continue check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than the first inner authentication request). If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.

   If this filter is not specified, no attributes from the outer request are transferred to the inner request.
4. Optionally, click the **Edit New** check box and select the filter you want to use from the dropdown list.

   This filter affects only a new inner authentication request (rather than continuations of previous requests). If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (by the filter specified in Step 2) and attributes included in the inner authentication request sent through the tunnel by the client.

   If this filter is not specified, the request remains unaltered.
5. Optionally, click the **Edit Continue** check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than a new inner authentication request). If this filter is specified, it is applied to the inner request that is the cumulative result of attributes transferred from the outer request (by the filter specified in Step 3) and attributes included in the inner authentication request sent through the tunnel by the client.

   If this filter is not specified, the request remains unaltered.

Configuring Response Filters

Response filters affect the attributes in the final response (Access-Accept or Access-Reject) returned to the originating NAD. By default, Steel-Belted Radius does not use response filters.

To configure response filtering for the EAP-PEAP protocol:

1. Click the **Response Filters** tab in the Edit PEAP Authentication method page.
2. Optionally, click the **Transfer Inner Attribs to Accept** check box and select the filter you want to use from the dropdown list.

   This filter affects only an outer Access-Accept response that is sent back to a network access device.

   - If this filter is specified, the filter is applied to the inner authentication response and all resulting attributes are transferred to the outer authentication response.
   - If this filter is not specified, no inner authentication response attributes are transferred to the outer authentication response.
3. Optionally, click the Transfer Inner Attribs to Reject check box and select the filter you want to use from the dropdown list.

   This filter affects only a continued inner authentication request (rather than the first inner authentication request). If this filter is specified, all attributes from the outer request are transferred to the inner request and this filter is applied. The transfer occurs and the filter is applied before any attributes specified in the inner authentication are added to the request.

   If this filter is not specified, no attributes from the outer request are transferred to the inner request.

Configuring Session Resumption

Session resumption settings let you specify whether session resumption is permitted and under what circumstances session resumption is performed.

Note: For session resumption to work, the network access device must be configured to handle the Session-Timeout return list attribute, because the network access device must be able to tell the client to reauthenticate after the session timer has expired.

To configure session resumption for the EAP-PEAP protocol:

1. Click the **Session Resumption** tab in the Edit PEAP Authentication method page.
2. Enter the maximum number of seconds you want the client to remain connected to the network access device before having to re-authenticate in the **Session TImeout** field.

   If you enter a number greater than 0, the lesser of this value and the remaining resumption limit is sent in a Session-Limit attribute to the RADIUS client on the RADIUS Access-Accept response.

   If you enter 0, no Session-Limit attribute is generated. This does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

   Entering a value such as 600 (10 minutes) does not necessarily cause a full re-authentication to occur every 10 minutes. You can configure the resumption limit to make most re-authentications fast and computationally cheap.
3. Enter the value that you want returned in a Termination-Action attribute in the **Termination Action** field.

   The Termination-Action attribute is a standard attribute supported by most Access Points and determines what happens when the session timeout is reached. Valid values are:

   - -1: Do not send the attribute.

- 0: Send the Termination-Action attribute with a value of 0.
- 1: Send the Termination-Action attribute with a value of 1.

Default value is -1. Note that this does not prevent the authentication methods performing secondary authorization from providing a value for this attribute.

4. Enter the maximum number of seconds you want the client to be able to re-authenticate using the TLS session resumption feature in the **Resumption Limit** field.

   This type of re-authentication is fast and computationally cheap. It does, however, depend on previous authentications and may not be considered as secure as a complete (computationally expensive) authentication. Specifying a value of 0 disables the session resumption feature.

Configuring Inner Authentication Settings

Inner authentication settings let you specify the manner in wich the inner authentication step operates. To configure inner authentication settings for the EAP-PEAP protocol:

1. Click the **Inner Authentication** tab in the Edit PEAP Authentication method page.
2. Optionally, enter the name of a directed realm in the **Directed Realm** field.

   Specifying the name of a directed realm causes the request to be routed based on the methods listed in the directed realm. Omitting this setting causes the inner authentication request to be handled like any other request received from a network access device.

3. Optionally, enter the name of a realm selection script in the **Realm Selection Script** field.

   You must license the Steel-Belted Radius scripting module to use realm selection scripts. For information on the Steel-Belted Radius scripting module, refer to the Steel-Belted Radius Scripting Guide.

Configuring Advanced Server Settings

Advanced server settings specify the manner in which the inner authentication step operates. To configure advanced server settings for the EAP-PEAP protocol:

1. Click the Advanced Server Settings tab in the Edit PEAP Authentication method page.
2. Enter the maximum length of the TLS message that may be generated during each iteration of the TLS exchange. in the **TLS Message Fragment Length** field.

   Enter a number in the range 500–4096. This value affects the number of RADIUS challenge/response round-trips required to conclude the TLS exchange. A value of 1400 may result in 6 round-trips, while a value of 500 may result in 15 round-trips.

   Some Access Points may have problems with RADIUS responses or EAP messages that exceed the size of one Ethernet frame (1500 bytes including IP/UDP headers).

   The default length for TLS messages is 1020 bytes, which prevents the RADIUS challenge response (carried in a UDP packet) from exceeding one Ethernet frame.

3. Enter xxx in the **Max Transaction** Time field.
4. Enable the **Return MPPE Keys** check box to specify whether the EAP-PEAP module includes RADIUS

MS-MPPE-Send-Key and MS-MPPE-Recv-Key attributes in the final RADIUS Access-Accept response sent to the Access Point.

You should enable this option if the Access Point needs to key the WEP encryption. If the Access Point is authenticating only end-users and WEP is not being used, you can clear this check box.

5. Use the DH Prime Bits list to specify the number of bits in the prime number that the module uses for Diffie-Hellman exponentiation.

   Selecting a longer prime number makes the system less susceptible to certain types of attacks but requires more CPU processing to compute the Diffie-Hellman key agreement operation.

   Valid values are 512, 1024, 1536, 2048, 3072, and 4096 bits.

6. Enter the TLS cipher suites (in order of preference) that the server is to use in the **Cipher Suites** field.

   These cipher suites are documented in RFC 2246, "The TLS Protocol Version1.2," and other TLS-related RFCs and draft RFCs.

   Default value is 0x3C, 0x3D, 0x67, 0x6B, 0x40, 0x6A, 0x9C, 0x9D, 0x9E, 0x9F, 0xA2, 0xA3.

7. Enter the minimum version of the PEAP protocol that the server should negotiate in the **PEAP Minimum Version** field.

   If you enter 0, the server negotiates version 0, which is compatible with Microsoft's initial PEAP implementation (shipped in Microsoft XP Service Pack 1).

   If you enter 1, the server negotiates version 1, which is compatible with Cisco's initial PEAP implementation (shipped in Cisco ACU).

   ℹ Note: The value entered in this setting must be less than or equal to the value entered for the PEAP Maximum Version field.

8. Enter the maximum version of the PEAP protocol that the server should negotiate in the PEAP Maximum Version field.

   If you enter 0, the server negotiates version 0, which is compatible with Microsoft's initial PEAP implementation (shipped in Microsoft XP Service Pack 1).

   If you enter 1, the server negotiates version 1, which is compatible with Cisco's initial PEAP implementation (shipped in Cisco ACU).

   ℹ Note: The value entered in this setting must be equal to or greater than the value entered for the **PEAP Minimum Version** field.

Configuring Server Certificates

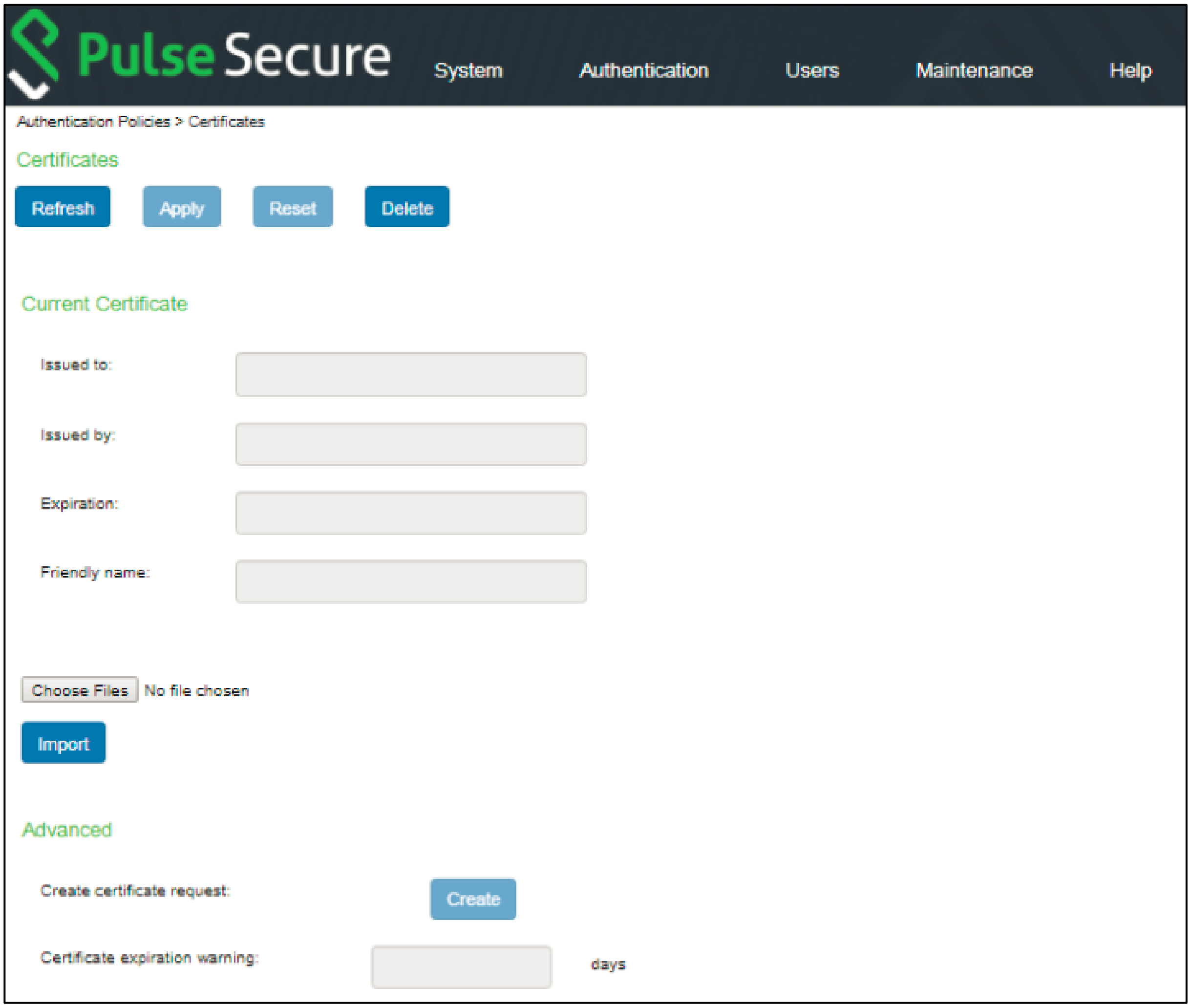
You must install cert if

Adding a Server Certificate

To add a certificate to the Steel-Belted Radius server:

1. Select Authentication > Authentication Policies > Certificates from the menu bar to open the Certificate page.

Figure 157: Certificates Page

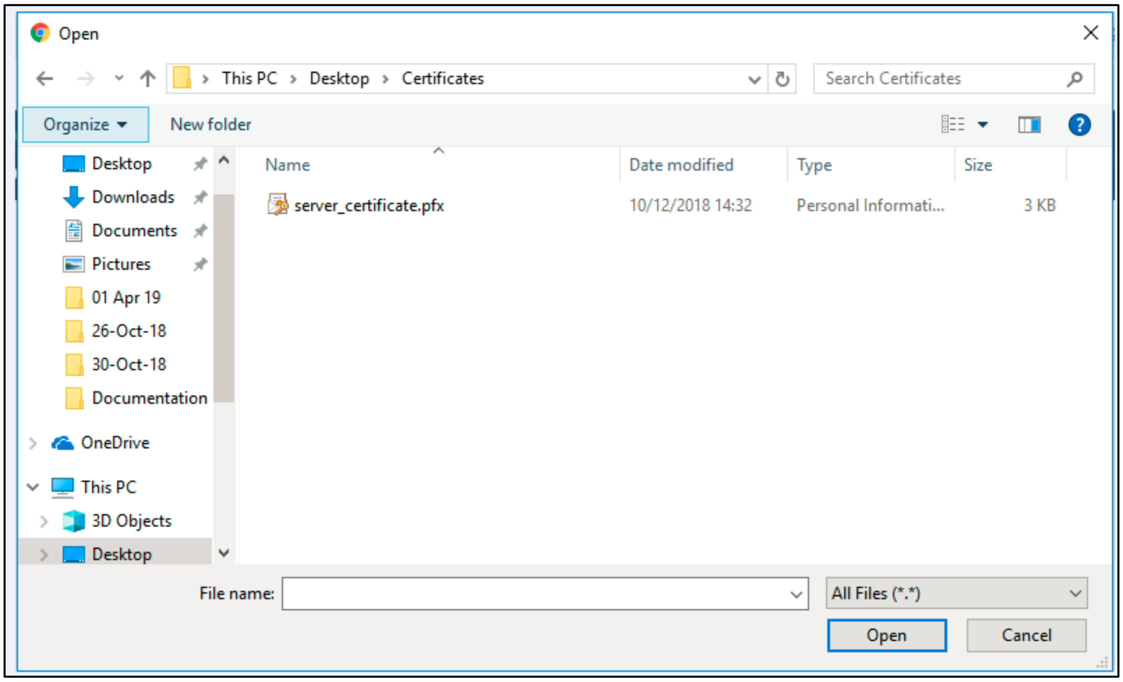


2. Click the **Choose File/ Browse** button.

3. When the Select Server Certificate dialog (Figure 158: Select Server Certificate Page) opens, navigate to the location of your server certificate, select the .pfx extension certificate you want to use, and click **Open.**

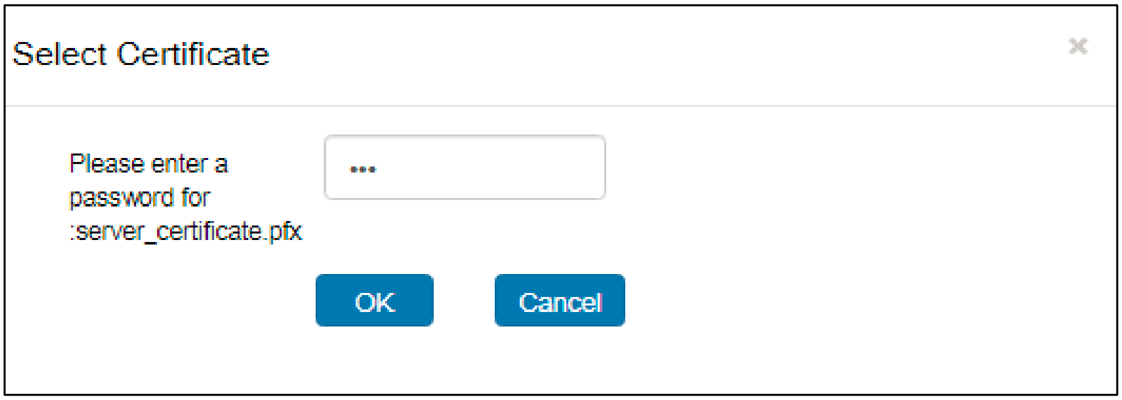
Note: The new WebGUI Administrator accepts certificates with only .pfx extension.

Figure 158: Select Server Certificate Page



4. When the Select Certificate page (Figure 159: Select Certificate Page) opens, enter the password for the certificate you selected and click OK.

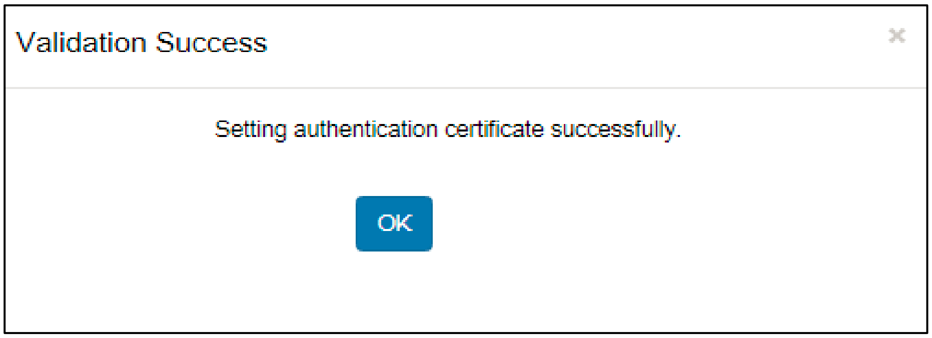
Figure 159: Select Certificate Page


Select Certificate

Please enter a password for :server_certificate.pfx

5. When the Validation Success page appears, click OK.

Figure 160: Validation Success Page


Validation Success

Setting authentication certificate successfully.

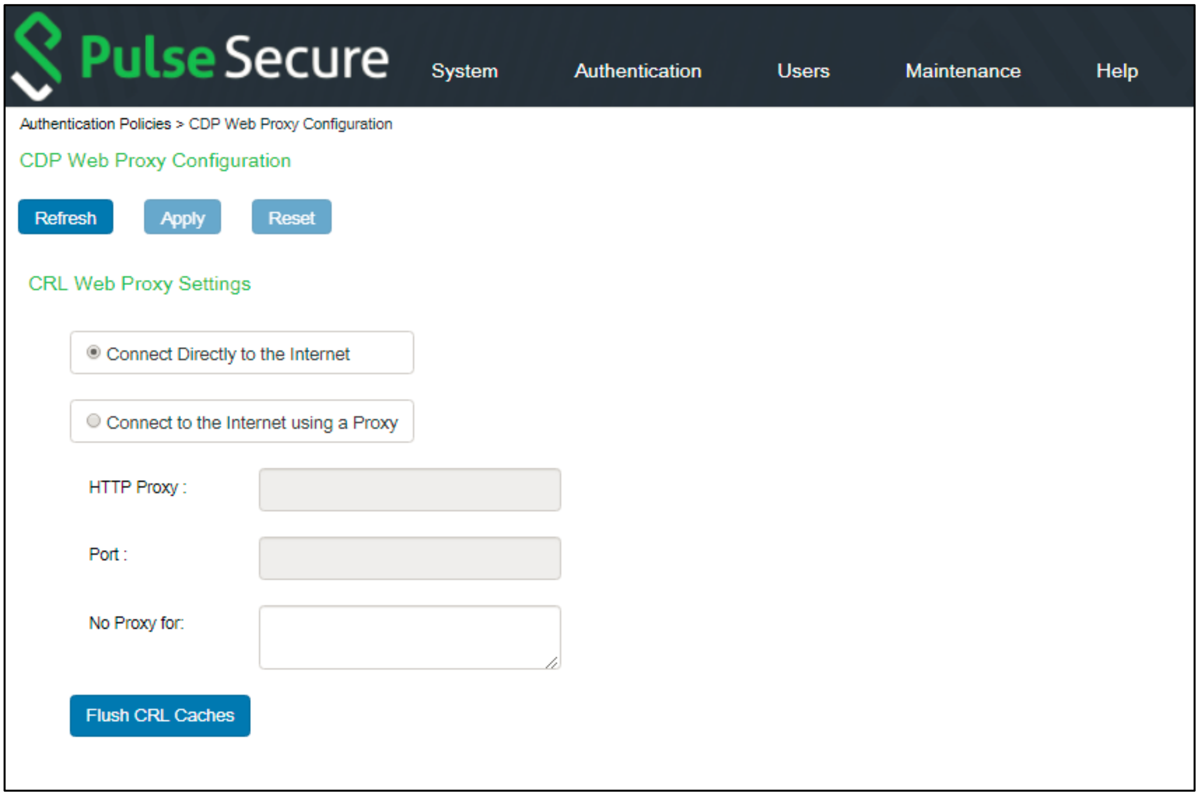
Configuring a CDP Web Proxy

Your network security policies may prohibit Steel-Belted Radius from making a direct HTTP connection to a CRL distribution points (CDP). You can configure an HTTP proxy server to relay requests from Steel-Belted Radius for updated certificate revocation lists to an external CDP.

To configure a CDP web proxy server:

1. Select Authentication > Authentication Policies > CDP Web Proxy Configuration from the menu bar to open CDP Web Proxy Configuration page.

Figure 161: CDP Web Proxy Configuration Page



Authentication Policies > CDP Web Proxy Configuration

CDP Web Proxy Configuration

CRL Web Proxy Settings

● Connect Directly to the Internet

○ Connect to the Internet using a Proxy

HTTP Proxy :

Port :

No Proxy for:

2. Specify whether you want Steel-Belted Radius to use a proxy to connect to an external CDP.
   - If you click **Connect Directly to the Internet**, Steel-Belted Radius can connect to an external CDP without going through an HTTP proxy. If you select this option, you can skip the rest of this procedure.
   - If you click **Connect to the Internet Using a Proxy**, Steel-Belted Radius must go through an HTTP proxy to connect to a CDP.
3. Enter the name or IP address of the HTTP proxy in the **HTTP Proxy** field.
4. Enter the port number to use for the HTTP proxy in the **Port** field.
5. Optionally, identify the hosts for which no HTTP proxy is required in the **No Proxy For** field. If a CDP host matches an entry in this field, Steel-Belted Radius bypasses the HTTP proxy and attempts to

open a connect to the host directly.

You can enter the names or IP addresses of hosts or the names of domains, separating each entry with a comma or semi-colon. Steel-Belted Radius compares IP addresses and host names using an exact string match. For example, if you enter cdp.pulsesecure.net in the exclusion list, that will match the CDP hostname cdp.pulsesecure.net but not host.cdp.pulsesecure.net or host-cdp.pulsesecure.net.

To exclude all hosts within a domain (but not the host name that matches the domain name), start the domain name with a period (.pulsesecure.net). To exclude both the host and the domain pulsesecure.net, create two entries in the exclusion list (.pulsesecure.net, pulsesecure.net).

Wildcard matching for host or domain names is not supported.

The values localhost and 127.0.0.1 are included in the No Proxy For list by default.

6. Optionally, click the **Flush CRL Caches** button to purge all information in the TLS and TTLS CRL caches immediately. When the Flush CRL Caches button is clicked, all CRL entries for registered clients are purged from the in-memory cache and deletes all files from the CRL cache directories.

Note: If you click the **Flush CRL Caches** button, the caches are purged immediately. You are not asked to confirm your action.

Configuring the Server

Depending on your authentication requirements, you may need to configure Steel-Belted Radius to work with an external SQL or LDAP database, RSA SecurID service, or TACACS+.

Configuring External Databases (Linux)

If you run Linux and want to use external databases for authentication or accounting purposes (and you did not configure this feature when prompted by the Steel-Belted Radius installation script), you can set up external database configuration settings.

To configure Steel-Belted Radius to work with an external database:

1. Optionally, perform the instructions in "Configuring SQL Authentication" and/or "Configuring SQL Accounting".
2. If you want to use Steel-Belted Radius with an LDAP database, review your LDAP database vendor's documentation.
3. Perform the instructions in "Configuring LDAP Authentication".

Configuring SecurID Authentication

If you want to use SecurID authentication, you must configure Steel-Belted Radius to communicate with the RSA SecurID server.

Perform the following steps to configure a Steel-Belted Radius server to work with an RSA SecurID server. If you are not familiar with the RSA SecurID server, contact your RSA SecurID server administrator for assistance.

1. Verify that the Steel-Belted Radius server has an entry on the RSA SecurID server.

Start the RSA SecurID server administration program and display the list of clients. If the list of clients does not include the Steel-Belted Radius server, select **Client > Add Client** and complete the Client window, giving the Steel-Belted Radius server a Client type of **Net OS Client**.

2. Copy the sdconf.rec file from the \ACE\data directory on the RSA SecurID server to the appropriate directory on the Steel-Belted Radius server:

   - **Windows:** C:\winnt\system32
   - **Linux:** the directory that contains the radius daemon on the Steel-Belted Radius server.

3. Edit the [SecurID] section of radius.ini. The radius.ini file is found in the same directory as the Steel-Belted Radius service or daemon.

   Verify that the **CachePasscodes** field is set to **yes** and the **SecondsToCachePasscodes** field is set to an appropriate number of seconds. These settings ensure that authenticated SecurID users can open a second B-channel during an ISDN connection.

4. Edit the [SecurID] section of the eap.ini file, which is found in the same directory as the Steel-Belted Radius service or daemon.

   Verify that the EAP settings in this section are enabled (remove the semi-colon from the start of each line) if you plan to use RSA SecurID authentication with EAP Generic-Token protocol support. The client system must support this protocol as well for this combination to work.

5. If you copy the sdconf.rec file after the Steel-Belted Radius service (daemon) has been started, or if you edit the radius.ini or eap.ini files after Steel-Belted Radius has been started, stop and restart Steel-Belted Radius.

6. Verify connectivity between the Steel-Belted Radius server and the RSA SecurID server.

   The RSA SecurID server offers a monitoring window on which it logs every authentication transaction, complete with the reason for the accept or reject decision. You can verify that pass-through to RSA SecurID is working, by creating a SecurID User called <ANY> and then attempting to access the network. Look for your request on the RSA SecurID monitor screen. If access is denied, you'll know that there's a configuration problem. Try these steps again, or contact your RSA SecurID administrator for assistance.

These steps complete initial setup of the two servers. To fully enable pass-through authentication to the RSA SecurID server, you must also set up the SecurID authentication method.

Configuring the Location of the sdconf.rec File

The VAR_ACE variable in the sbrd script file (Linux) lets you specify the directory holding the sdconf.rec file. The VAR_ACE variable must be exported so that Steel-Belted Radius can use it.

For example:

```
VAR_ACE=”radiusdir/ace”

export VAR_ACE
```

This variable is set by default in the file to point to the radiusdir directory. If the variable is not set at all in the file, the server sets the value of this variable to

/var/ace.

Configuring TACACS+ Authentication

If you want to use TACACS+ authentication, you must configure Steel-Belted Radius to communicate with the TACACS+ server.

Perform the following steps to configure a Steel-Belted Radius server to work with a TACACS+ server.

1. Verify the tacplus.ini file is present in the Steel-Belted Radius directory.

   The tacplus.ini file must be present in the same directory as the Steel-Belted Radius service (in the case of Windows, usually C:\Program Files (x86)\Pulse Secure\Steel-Belted Radius\Service), or daemon (in the case of Linux). This happens automatically following installation.

2. Edit the tacplus.ini file to identify the shared secret and host machine that you use for TACACS+. For more information on the tacplus.ini file, refer to the Steel-Belted Radius Reference Guide.

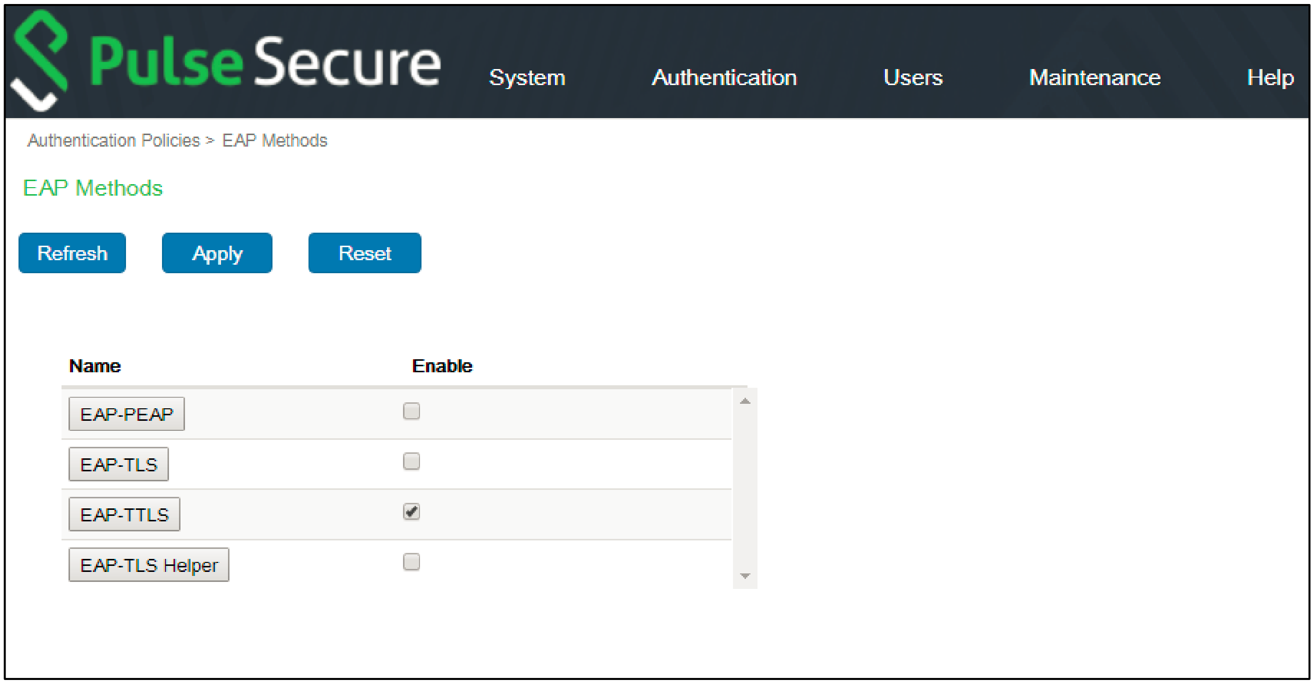
3. If you edit tacplus.ini after Steel-Belted Radius has been started, then you must stop and restart it before your changes take effect.

To enable pass-through authentication to the TACACS+ server, you must also set up the TACACS+ authentication method. For more information, see "Configuring TACACS+ Authentication".

Activating EAP Methods

The EAP Methods page permits you to activate authentication methods and define the order in which different authentication methods are attempted.

Figure 162: EAP Methods Page



Name	Enable
EAP-PEAP	☐
EAP-TLS	☐
EAP-TTLS	☑
EAP-TLS Helper	☐

To use the EAP Methods page:

1. Choose Authentication > Authentication Policies > EAP Methods from the menubar.

2. Click the Enable check box to enable the EAP authentication methods you want Steel-Belted Radius to use.

To revert to the previous settings, click Reset.

Configuring EAP Settings

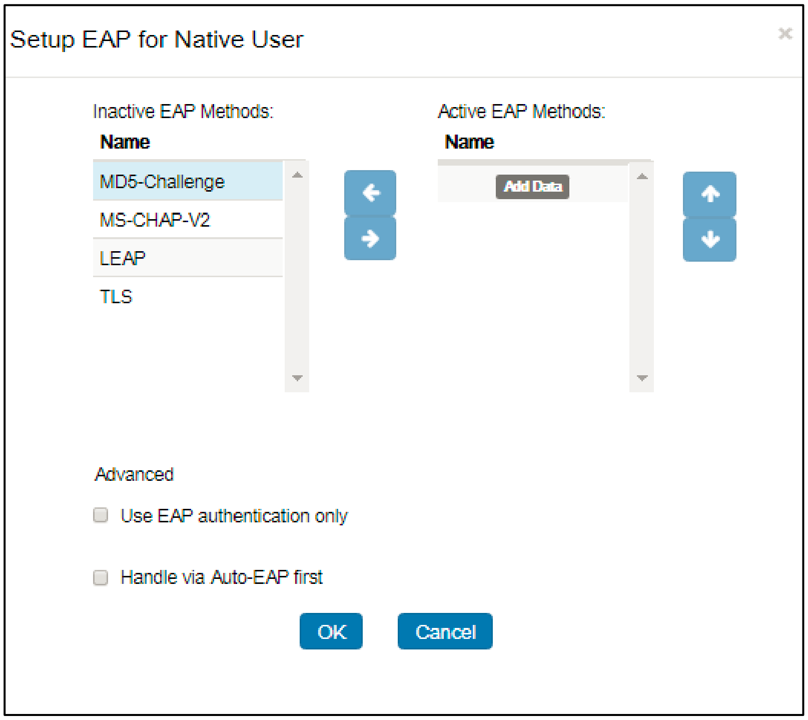
When Steel-Belted Radius receives a username, it does not know in advance to which authentication category this user belongs. It must try each method that it currently has configured and enabled. The authentication methods list allows you to fine-tune the sequence of authentication attempts.

Note: The EAP Setup page displays the authentication methods that have been enabled (by editing the Enabled setting in the appropriate *.aut file).

To set up EAP settings for an authentication method:

1. Select the authentication method you want to set up in the Authentication Methods tab.
2. Click the **EAP Setup** button.
   The Setup EAP page opens.

Figure 163: Setup EAP Page



Setup EAP for Native User

Inactive EAP Methods:

Name
MD5-Challenge
MS-CHAP-V2
LEAP
TLS

Active EAP Methods:

Name
Add Data

Advanced

☐ Use EAP authentication only

☐ Handle via Auto-EAP first

3. Optionally, change the order in which the methods are tried by highlighting a method and clicking the **Up** or **Down** buttons.
4. To activate a method, (so that it can be used for authentication), move methods from Inactive EAP Methods to Active EAP Methods.

   If you want to deactivate a method (so that it is not used for authentication), move the methods from Active EAP Methods to Inactive EAP Methods.
5. If you want to restrict use of this authentication method to requests that contain EAP credentials,

click the **Use EAP authentication only** check box.

When this option is enabled, Steel-Belted Radius prevents the authentication method from being called for any request that does not contain EAP credentials, and bypasses the authentication method if an authentication request specifically requests an EAP protocol that is not listed in the authentication method's **EAP-Type** list in the **eap.ini** file.

6. If you want Steel-Belted Radius to use an automatic EAP helper to generate credentials for a user, click the **Handle via Auto-EAP first** check box.

   You should unclick the check box if an authentication method is capable of handling EAP credentials on its own (without an EAP helper).

   Refer to "First-Handle-Via-Auto-EAP Setting" for more information.

7. Click **OK** to return to the Authentication Methods tab.

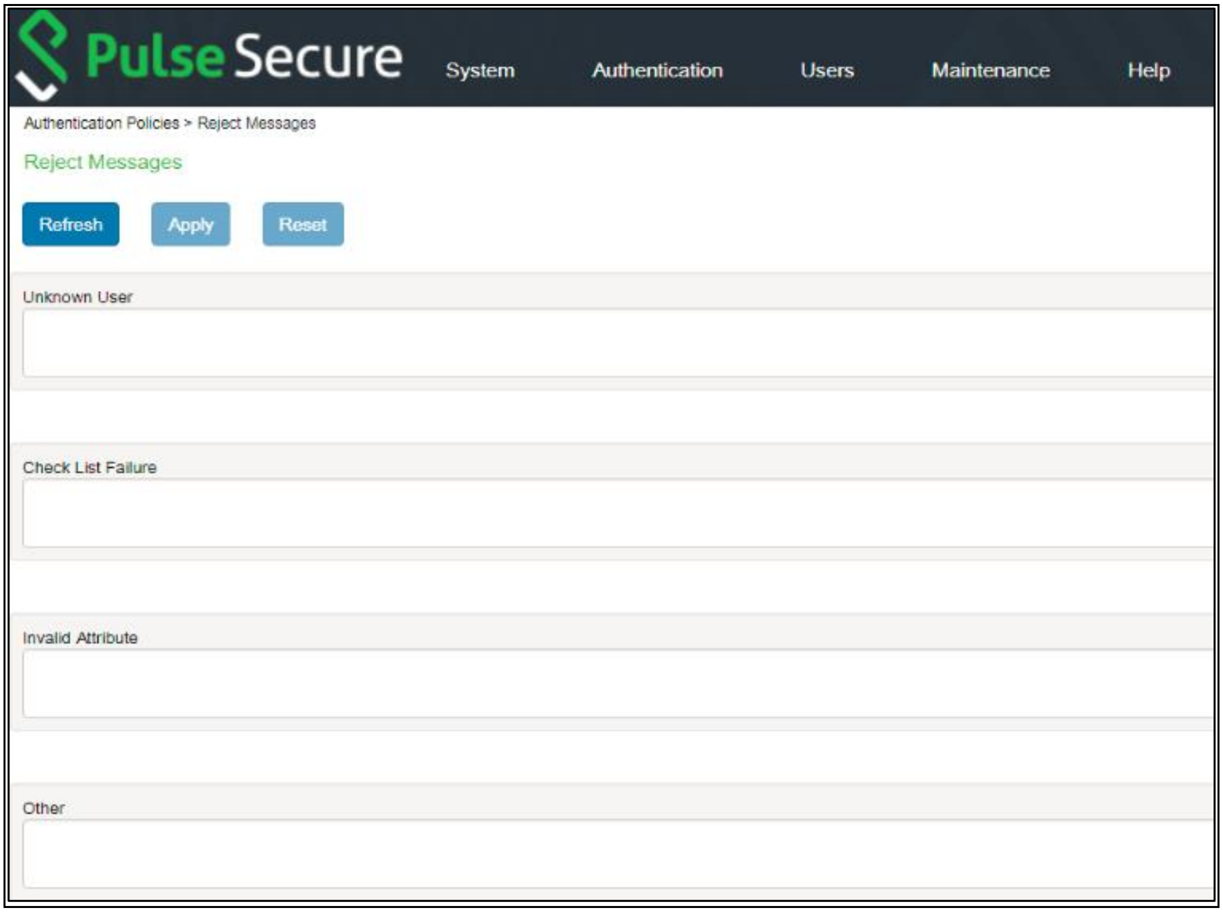
Configuring Authentication Rejection Messages

When Steel-Belted Radius issues an Access-Reject message in response to a failed authentication request, it can identify the reason why the request was rejected. You can configure the message text returned to the RADIUS client (and possibly to the user, if the RADIUS client forwards the message) when a particular type of error occurs. This text is inserted into the standard RADIUS attribute Reply-Message within the Access-Reject response.

To configure the text for authentication rejection messages:

1. Choose Authentication > Authentication Policies > Reject Messages from the menu bar.

Figure 164: Authentication Policies page: Reject Messages page



2. Use the **Unknown User** field to specify the message Steel-Belted Radius returns when the username and password authentication failed.

3. Use the **Checklist failure** field to specify the message Steel-Belted Radius returns when the user was authenticated but is being rejected because the RADIUS request did not fulfill the requirements of the checklist.

4. Use the I**nvalid attribute** field to specify the message Steel-Belted Radius returns when the request contained an attribute in violation of the RADIUS specification.

5. Use the **Other** field to specify the message Steel-Belted Radius returns when some other error, such as a resource failure, occurred.

6. When you are asked to confirm that you want to save your changes, click **Yes.**

Chapter 26

Configuring TACACS+ Server

This chapter describes how to configure TACACS+ Server in Steel-Belted Radius.

TACACS+ Basics

TACACS+ provides access control for routers, network access servers and other networked computing devices via one or more centralized servers. TACACS+ provides separate authentication, authorization and accounting services. The TACACS+ protocol is the latest generation of TACACS.

TACACS+ improves on TACACS and XTACACS by separating the functions of Authentication, Authorization and Accounting and by encrypting all traffic between the NAS and the daemon. It allows for arbitrary length and content authentication exchanges which will allow any authentication mechanism to be utilized with TACACS+ clients. It is extensible to provide for site customization and future development features, and it uses TCP to ensure reliable delivery. The protocol allows the TACACS+ client to request very fine-grained access control and allows the daemon to respond to each component of that request.

A very important benefit to separating authentication from authorization is that authorization (and per-user profiles) can be a dynamic process. Instead of a one shot user profile, TACACS+ can be integrated with other negotiations, such as a PPP negotiation, for far greater flexibility. The accounting portion can serve to provide security auditing or accounting/billing services.

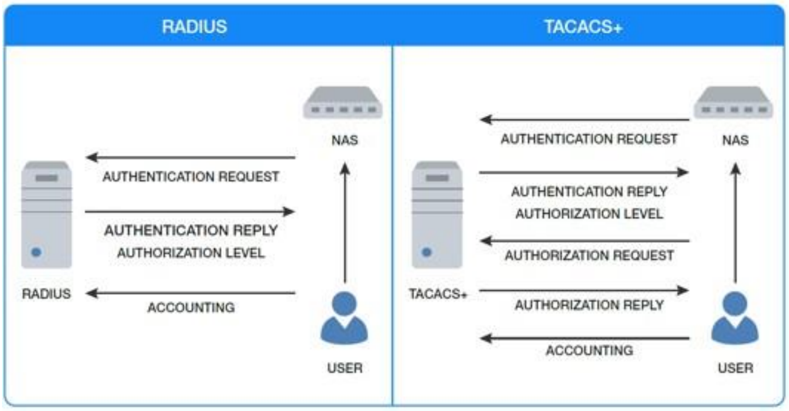
TACACS+ uses TCP for its transport. The daemon should listen at port 49 which is the "LOGIN" port assigned for the TACACS protocol. This port is reserved in the assigned numbers RFC for both UDP and TCP. Current TACACS and extended TACACS implementations use port 49.

Limitations of Radius protocol and how TACACS+ addresses them:

- Seperates the functions of Authentication, Authorization and Accounting. A very important benefit to separate authentication from authorization
- Encrypting all traffic between the NAS and the daemon.
- Allows for arbitrary length and content authentication exchanges which will allow any authentication mechanism to be utilized with TACACS+ clients.
- Extensible to provide for suite customization and future development features and its uses TCP to ensure reliable delivery. TACACS+ daemon should listen at port 49 which is the “Login”port assigned foe the TACACS protocol.
- RADIUS is mainly designed for subscriber AAA and TACAS+ is mainly designed for administrator AAA.

Figure 165: Radius and TACACS+ Message flow comparison illustrates the Radius and TACACS+ Message flow comparison.

Figure 165: Radius and TACACS+ Message flow comparison



TACACS+ Server Overview in Steel Belted Radius

"tac_plus"is a TACACS+ daemon. It provides Steel-Belted Radius server with TACACS+ authentication, authorization and accounting services. Key features include:

- NAS specific host keys, prompt and enable passwords
- NAS and ACL dependent group memberships
- Flexible external backends for user profiles (e.g via PERL scipts or C LDAP (Active Directory)
- Connection multiplexing (multiple concurrent NAS clients per process)
- Session multiplexing (multiple concurrent sessions per connection, single-connection)
- Scalable, no limit on users, clients or servers
- Compliant to latest TACACS+ protocol specification

By default TACACS+ Server functionality is disabled in Steel-Belted Radius. In order to use TACACS+ Server functionality in Steel-Belted Radius Server, enable parameter "EnableTACACSPlusServer"by setting to 1 in the "radius.ini"file and restarting 'radius"process.

tac_plus" daemon makes used of "tac_plusd.cfg" file for all its configurations. A single configuration file is sufficient for configuring all the use cases that are supported by TACACS+ Server.

Any changes to the 'tac_plusd.cfg" configuration file requires saving the file and restarting Steel-Belted Radius process.

Configuring TACACS+ port and number of TACACS+ instances

In general, for any daemon to run continuously it has to listen on a specific port for the incoming requests, TACACS+ is a multi-process application, where the master service that listens for TACACS+ requests runs as a thread in RADIUS process scope. Master service is only to receive a request and pass on to child entity to further processing of the packet. These child entries are multiple processes that serves a specific TACACS+ request assigned by its master. Following are the necessary and basic attributes for the process to run.

- Listen – mentions the port number
- Spawn – mentions the minimum and maximum instances/processes to be created. Minimum number gets created during start up itself. If the incoming requests are too high that the existing number of processes are not able to handle, then the number specified in maximum instances gets created during runtime to serve the high incoming requests.
- Background – whether to run in background or foreground

Sample configuration as follows,

```
id = spawnd {
    listen = { port = 49 }
    spawn = {
            instances min = 2
            instances max = 10
    }
    background = yes
}
```

The following use cases are supported in TACACS+ Server configuration,

Configuring TACACS+ Client in “tac_plusd” configuration file

Configuring TACACS+ Client is identified with the keyword “host” followed by actual hostname/TACACS+ Client Name. Then it contains the following necessary host attributes as follows,

- Address – IP Address of the TACACS+ Client host
- Prompt – <To add>
- Enable - <To add>
- Key – Shared Secret between TACACS+ Client and TACACS+ Server running in SBR

Sample configuration as follows,

```
host = world {
            address = ::/0
            prompt = "Welcome\n"
            enable 15 = clear secret
            key = sharedsecret
    }
```

Configuring Users in “tac_plusd” configuration file

Configuring User is identified with the keyword “user” followed by actual user name. Then it contains the following necessary user attributes as follows,

- Password – Password of the user
- Member – Group name that the user belongs to
- Service – Authorization Profile that gets assigned to the user and privilege level

Sample configuration as follows,

```
user = test {
            password = clear testpasswd
            member = admin
            service = shell {
                    default command = permit
                    default attribute = permit
                    set priv-lvl = 15
            }
    }
```

Password of the Tacacs+ User

Tacacs+ allows you to store user details in the tac_plusd.cfg file. The Password of the user can be stored as Clear-text or hash.

Clear-text Password:

The Clear-text passwords will be configured in tac_plusd.cfg using the keyword “Clear”.

```
user = test {
                   password = clear clear_text_password
                   member = admin
                   service = shell {
                       default command = permit
                       default attribute = permit
                       set priv-lvl = 15
                      }
                 }
```

Hashed password:

Md5 and crypt hashed password will be configured using the keyword “Crypt”.

```
user = test {
                  password = crypt $1$e4mIojbD$pV7HH6CQVLium6muNePDd/
                  member = admin
                  service = shell {
                       default command = permit
                       default attribute = permit
                       set priv-lvl = 15
                      }
                 }
```

Generate Password Hashes:

You can use the openssl passwd utility to compute password hashes.

For example:

openssl passwd -1 clear_text_password

returns a MD5 hash, while

openssl passwd -crypt clear_text_password

returns a DES hash.

DES passwords are by design truncated to eight characters.

The above hashed passwords provide security for the user- password.

A user may be member of a group (also known as role or profile). Actual group membership may depend on various factors, e.g. the NAS the user is on, the NAC, or time ranges. Each group may in turn be member of another group and so on, ad infinitum.

Configuring Groups in “tac_plusd” configuration file

Configuring Group is identified with the keyword “group” followed by actual group name. Then it contains the following necessary group attributes as follows,

- Default service – Whether to permit or deny the service
- Service - Authorization Profile that gets assigned to the group and privilege level

Sample configuration as follows,

```
group = admin {
            default service = permit
            service = shell {
                    default command = permit
                    default attribute = permit
                    set priv-lvl = 15
            }
           }
```

Order of Authentication methods

Currently TACACS+ Server in Steel-Belted Radius application supports only the following authentication methods

1. Flat File Users Authentication – Users/Groups/Profiles configured in the configuration file "tac_plusd.cfg"
2. LDAP Backend Authentication – Users stored in LDAP Backend (eg, Microsoft Active Directory LDAP backend)
3. Shadow Backend Authentication – Users stored in SHADOW files

By default, users configured in Flat File will be given the first preference. Then for other 2 backends (LDAP &SHADOW), the order of authentication depends on the order in which the authentication methods are placed in the configuration file "tac_plusd.cfg" file. If "external" module SHADOW was placed before "external" module LDAP, then the authentication processing will be in the following order,

- Flat File Users Authentication
- Shadow Backend Authentication
- LDAP Backend Authentication

Configuring LDAP Backend Authentication

TACACS+ Server can authenticate against records stored in an external LDAP database. Any attribute(s) such as username and password, can be used to query the database.

External database authentication is typically used when an organization has a large amount of user information stored in an LDAP database and wants to authenticate these users using TACACS+. Authentication against an existing LDAP database extends authentication services to user accounts without requiring an administrator to enter user information into TACACS+ database.

TACACS+ offers LDAP authentication as a plug-in software module. "mavis_tacplus_ldap.pl" is an authentication/authorization backend for the external module. It interfaces to various kinds of LDAP servers, e.g. OpenLDAP, Fedora DS and Active Directory. SBR-E 6.24 release supports Microsoft Active Directory. Its behaviour is controlled by a list of environmental variables.

Configuring LDAP Authentication is identified with the keyword "external" followed by specifying the executable file "mavis_tacplus_ldap.pl" with the keyword "exec". Then it contains the following necessary LDAP attributes as follows,

Variable	Description
LDAP_SERVER_TYPE	One of: generic,

Variable	Description
	tacacs_schema, microsoft. Default: tacacs_schema Value set to “Microsoft”
LDAP_HOSTS	Space-separated list of LDAP URLs or IP addresses or hostnames Examples: "ldap01 ldap02", "ldaps://ads01:636 ldaps://ads02:636" "1.1.1.1:389"
LDAP_SCOPE	LDAP search scope (base, one, sub) Default: sub
LDAP_BASE	Base DN of your LDAP server Example: dc=example,dc=com Example: "dc=64windows2008,dc=pulse,dc=com"
LDAP_FILTER	LDAP search filter. Defaults: • for LDAP_SERVER_TYPE=generic: "(uid=%s)" • for LDAP_SERVER_TYPE=tacacs_schema: "(&(uid=%s)(objectClass=tacacsAccount))" • for LDAP_SERVER_TYPE=microsoft: "(&(objectclass=user)(sAMAccountName=%s))"
LDAP_FILTER_CHPW	LDAP search filter for password changes. Defaults: • for LDAP_SERVER_TYPE=generic: "(uid=%s)" • for LDAP_SERVER_TYPE=tacacs_schema: "(&(uid=%s)(objectClass=tacacsAccount)(!(tacacsFlag=staticpasswd)))" • for LDAP_SERVER_TYPE=microsoft: "(&(objectclass=user)(sAMAccountName=%s))"
LDAP_USER	User to use for LDAP bind if server doesn't permit anonymous searches. Default: unset
LDAP_PASSWD	Password for LDAP_USER Default: unset

Variable	Description
	Note: 1. "LDAP_PASSWD" (password for LDAP_USER) entered will be encrypted and overwritten after restart. 2. Encrypted password starts with "ENC" to denote that it is encrypted. 3. After encryption, if the password content needs to be modified then replace the complete encrypted string with a new password text and restart SBR for the new encrypted string to be written.
AD_GROUP_PREFIX	An AD group starting with this prefix will be used as the user's TACACS+ group membership. The value of AD_GROUP_PREFIX will be stripped from the group name. Example: With AD_GROUP_PREFIX set to tacacs (which is actually the default), an AD group membership of TacacsNOC will assign the user to the NOC TACACS+ group. Note that TACACS+ group names are case-sensitive.
REQUIRE_AD_GROUP_PRE FIX	If set, user needs to be in one of the AD_GROUP_PREFIX groups. Default: unset
USE_TLS	If set, the server is required to support start_tls. Default: unset, Value set to 0
FLAG_CHPW	Permit password changes via this backend. Default: unset
FLAG_PWPOLICY	Try to enforce a simplicistic password policy. Default: unset
FLAG_CACHE_CONNECTI ON	Keep connection to LDAP server open. Default: unset
FLAG_FALLTHROUGH	If searching for the user in LDAP fails, try the next MAVIS module (if any). Default: unset
FLAG_USE_MEMBEROF	Use the memberOf attribute for determining group membership. Setting LDAP_SERVER_TYPE to microsoft implies this. May be used if you're running OpenLDAP with memberof overlay enabled. Default: unset, value set to 1

Sample configuration as follows,

```
mavis module = external {
        setenv LDAP_SERVER_TYPE = "microsoft"
        setenv LDAP_HOSTS = "1.1.1.1:389"
        setenv LDAP_SCOPE = sub
        setenv LDAP_BASE = "dc=64windows2008,dc=pulse,dc=com"
        setenv LDAP_FILTER = "(&(objectclass=user)(sAMAccountName=%s))"
        setenv LDAP_USER = test@64windows2008.pulse.com
        setenv LDAP_PASSWD = test
        setenv USE_TLS = 0
          setenv FLAG_USE_MEMBEROF = 1
```

```
    setenv AD_GROUP_PREFIX = tes
  exec = /opt/PSsbr/radius/mavis/mavis_tacplus_ldap.pl
  # see the MAVIS configuration manual for more options
}
```

Configuring SHADOW Backend Authentication

TACACS+ Server can authenticate against records stored in an SHADOW backend of that particular local machine. Attribute(s) such as username and password, can be used to query the database.

Authentication against an existing SHADOW database extends authentication services to user accounts without requiring an administrator to enter user information into TACACS+ database.

TACACS+ offers SHADOW authentication as a plug-in software module. “mavis_tacplus_shadow.pl” is an authentication backend for the external module. It interfaces to the SHADOW database configured in the local Linux Operating System. Its behaviour is controlled by a list of environmental variables.

Configuring SHADOW Authentication is identified with the keyword “external” followed by specifying the executable file “mavis_tacplus_shadow.pl” with the keyword “exec”. Then it contains the following necessary SHADOW attributes as follows,

- SHADOWFILE - /etc/shadow is generally the file that stores the local username/password configured in the Linux Operating System of the local machine

Sample configuration as follows,

```
mavis module = external {
        setenv SHADOWFILE = /etc/shadow
        exec = /opt/PSsbr/radius/mavis/mavis_tacplus_shadow.pl
        # see the MAVIS configuration manual for more options
    }
```

TACACS+ Logging

TACACS+ daemon running in Steel Belted Radius application makes use of syslog utility to log its transactions. Ensure that syslog is running fine in the Linux machine before starting with the Steel Belted Radius application.

Tuning “tac_plusd” configuration file for TACACS+ Logging

TACACS+ application log can be broadly classified as follows,

- Default – Only default logs such as success, failure information
- Debug – Detailed application processing information

It is required to tune the Syslog daemon for viewing the default or detailed application flow processing and refer Section “Tuning Syslog Daemon for TACACS+ Logging” for more details.

To rotate TACACS+ Log files, based on the criteria such as Time or Size, use "Logrotate" Linux utility. Refer to the section "Log Rotation of TACACS+ Log Files" for more details.

Trace (or debugging) options may be specified in global, host, user and group context. The current debugging level is a combination of all those. Generic syntax is:

```
Debug = option…
```

For example, getting command authorization to work in a predictable way can be tricky – the exact attributes the NAS sends to the daemon may depend on the IOS version and may in general not match your expectations. If your regular expressions don’t work, add

Debug = REGEX

Where appropriate and the daemon may log some useful information to syslog.

Multiple trace options may be specified. Example:

Debug = REGEX CMD

The debugging options available are summarized in the Debug and Trace Values Table.

Bit	Value	Name	Description
0	1	PARSE	Configuration file parsing
1	2	AUTHOR	Authorization related
2	4	AUTHEN	Authentication related
3	8	ACCT	Accounting related
4	16	CONFIG	Configuration related
5	32	PACKET	Packet dump
6	64	HEX	Packet hex-dump
7	128	LOCK	File locking
8	256	REGEX	Regular expressions
9	512	ACL	Access Control Lists
10	1024	RADIUS	unused
11	2048	CMD	Command lookups
12	4096	BUFFER	Buffer handling
13	8192	PROC	Procedural traces
14	16384	NET	Network related
15	32768	PATH	File system path related
16	65536	CONTROL	Control connection related
17	131072	INDEX	Directory index related
18	262144	AV	Attribute-Value pair handling
19	524288	MAVIS	MAVIS related
20	1048576	LWRES	DNS related
31	2147483648	NONE	Disable debugging

There is also a facility provided for the administrator to provide logs for

Note: All accounting records are written, as text, to the file (or command) specified with the accounting log directive.

Accounting records are text lines containing tab-separated fields. The first 6 fields are always the same. These are:

- timestamp
- NAS address
- username
- port
- NAC address
- record type

Following these, a variable number of fields are written, depending on the accounting record type. All the form attribute=value. There will always be a task_id field.

Current attributes are:

unknown service start_time port elapsed_time status priv_level cmd protocol cmd-arg bytes_in bytes_out paks_in paks_out address task_id callback-dialstring nocallback-verify callback-line callback-rotary

More may be added over time.

Example records (lines wrapped for legibility) are thus:

1995-07-13 13:35:28 -0500 172.16.1.4 chein tty5 198.51.100.141 stop task_id=12028 service=exec port=5 elapsed_time=875

1995-07-13 13:37:04 -0500 172.16.1.4 lol tty18 198.51.100.129 stop task_id=11613 service=exec port=18 elapsed_time=909

1995-07-13 14:09:02 -0500 172.16.1.4 billw tty18 198.51.100.152 start task_id=17150 service=exec port=18

1995-07-13 14:09:02 -0500 172.16.1.4 billw tty18 198.51.100.152 start task_id=17150 service=exec port=18

Elapsed time is in seconds, and is the field most people are usually interested in.

- Accounting (accounting) - accounting log = /opt/PSsbr/radius/tacplus_accounting.log
- Access (authentication & authorization) – access log = /opt/PSsbr/radius/tacplus_access.log

TACACS+ configuration file is set by default with the following parameters

- syslog facility = local6
- syslog level = debug

Tuning Syslog Daemon for TACACS+ Logging

TACACS+ application log can be classified as follows,

1. Default
   TACACS+ Default logs will be displayed in syslog level “info”. By default, all syslog “info” logs can be viewed in “/var/log/messages”. If not displayed in this file, refer syslog configuration file (/etc/syslog.conf until RHEL5 and /etc/rsyslog.conf from RHEL 6 onwards) to know where syslog “info” level logs are logged.
2. Debug
   TACACS+ Debug logs will be displayed in syslog level “debug”. In general, debug configurations will not be explicitly mentioned in syslog configuration file (/etc/syslog.conf until RHEL5 and

/etc/rsyslog.conf from RHEL 6 onwards) by default. Therefore, it has to be explicitly mentioned in syslog configuration file.

a. Before making any changes to syslog configuration file, the syslog service has to be stopped. In the commandline, execute the following command
service syslog stop (until RHEL5)
service rsyslog stop (from RHEL6 onwards)

Below suggested syslog configurations are required to be added manually in the syslog configuration file (/etc/syslog.conf until RHEL5 and /etc/rsyslog.conf from RHEL 6 onwards) for viewing all the TACACS+ debug logs

b. Open the syslog configuration file (/etc/syslog.conf until RHEL5 and /etc/rsyslog.conf from RHEL 6 onwards) using "vim" editor
c. Add the following entries under "#### MODULES ####"section

```
$SystemLogRateLimitInterval 0

$SystemLogRateLimitBurst 0

$IMUXSockRateLimitBurst 0

$IMUXSockRateLimitInterval 0

$IMUXSockRateLimitSeverity 7
```

d. Add the following entries under "#### RULES ####"section
local6.debug   <path to log TACACS+ debug logs (eg, /var/log/Tacplus.log (or) /opt/PSsbr/radius/Tacplus.log)>

Note: When SELINUX is enabled on Linux Server and TACACS+ module in SBR refers to a log file that is not a descendant of /var/log directory, following steps have to be adhered,

On the command prompt of Linux Server,

a. Create the empty log file that is mentioned in syslog configuration file (eg, /opt/PSsbr/radius/Tacplus.log is mentioned in rsyslog.conf file)
   touch /opt/PSsbr/radius/Tacplus.log
b. View the default ownership/permissions of that file (By default, it will not be a part of var_log_t group)
   ls -Z Tacplus.log
   rw-rr-. root root unconfined_u:object_r:usr_t:s0 Tacplus.log
c. Assign the file permissions of /var/log/messages to the newly created log file
   chcon --reference /var/log/messages /opt/PSsbr/radius/Tacplus.log
d. Now verify the file permissions of newly created log file (It will be part of var_log_t group)
   ls -Z Tacplus.log
   rw-rr-. root root system_u:object_r:var_log_t:s0 Tacplus.log.
e. Save the syslog configuration file.
f. All the required changes are completed and the syslog service can be started now
   service syslog start (until RHEL5)
   service rsyslog start (from RHEL6 onwards)

Log Rotation of TACACS+ Log Files

The logrotate program in Linux is a log file manager. It is used to regularly cycle (or rotate) the log files by removing the oldest log files from your system and creating the new log files. It may be used to rotate, based on the age of the file or the file’s size, and usually runs automatically through the cron utility. The logrotate program may also be used to compress log files and to configure e-mail for the users when they are rotated.

The logrotate program is configured by entering the options in the */etc/logrotate.conf* file.

In general, default configurations available in "*/etc/logrotate.conf*" will be applicable for the system log files. The configurations provided here can also be utilized for specific log files. Since TACACS+ Log file does not have log rotation implemented, this utility can be used to implement log rotation.

TACACS+ Log file and all the required configurations can be specified within this context.
In the following example, TACACS+ Log file is available in path /opt/PSsbr/radius/Tacplus.log, daily is specified to mention that Logrotation is done on daily basis and rotate 5 specifies that 5 weeks of Log files can be retained at a time, before it could be deleted.

For Example:

```
/opt/PSsbr/radius/Tacplus.log {
    daily
    rotate 5
}
```

For more configuration options refer to "man logrotate".

Chapter 27
Configuring SNMP

This chapter describes how to configure and use the optional Simple Network Management Protocol (SNMP) package to monitor your Steel-Belted Radius server.

Note: SNMP is supported on the Linux versions of Steel-Belted Radius. SNMP is not supported on the Windows version of Steel-Belted Radius.

About SNMP

SNMP is an IETF standard protocol that lets an administrator set configuration parameters and monitor operating statistics and status for a managed device, such as a server or router, from a remote location.

About the SNMP Package

The SNMP agent that you can install for Steel-Belted Radius is customized software based on version 5.0.9 of the open-source net-snmp toolkit. After it is installed, the SNMP agent exchanges information with Steel-Belted Radius automatically. The daemons and files used to configure and maintain SNMP information are installed in the radiusdir/snmp directory on your Steel-Belted Radius server.

Note: Although the SNMP agent for Steel-Belted Radius is based on the net-snmp toolkit, you cannot replace the SNMP agent with software downloaded from the net-snmp website.

SNMP Network Management Architecture

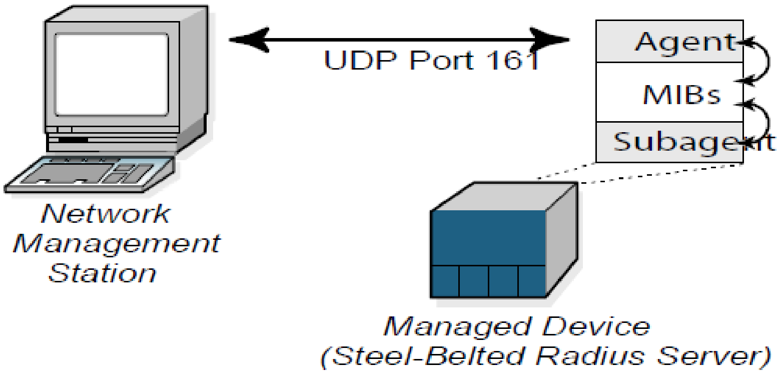
The SNMP network management architecture consists of managed devices, SNMP agents, and network management stations (NMS).

- A managed device is any host or hardware on a network that runs an SNMP agent. The Steel-Belted Radius server is a managed device after you install and configure the optional SNMP agent.
- A network management station (NMS) is an administration workstation that polls management agents for information and provides control information for agents. A network management station can also accept trap messages when an asynchronous event occurs on a managed device.
- An SNMP agent is a software module running on a managed device that is responsible for recording performance statistics and events in a database called a management information base (MIB) and for communicating with the NMS. The SNMP agent for Steel-Belted Radius is called pssnmpd. When an NMS requests information, the SNMP agent processes the request, acquires information from the management database, and forwards the information to the NMS. The SNMP agent can also accept control information from the NMS.

  An SNMP subagent may be responsible for gathering information about network activity relating to a particular service running on the managed device. Steel-Belted Radius runs an SNMP subagent that communicates with the pssnmpd agent transparently; you do not need to register or configure the Steel-Belted Radius subagent to work with the SNMP agent.

Figure 166: SNMP Architectureillustrates the SNMP management architecture.

Figure 166: SNMP Architecture


UDP Port 161
Agent
MIBs
Subagent
Network Management Station
Managed Device (Steel-Belted Radius Server)

SNMP Versions

Steel-Belted Radius supports SNMP version 1 (SNMPv1) and SNMP version 2c (SNMPv2c).

- SNMPv1 is the original implementation of SNMP, as defined in RFC 1157, "Simple Network Management Protocol (SNMP)."
- SNMPv2c is an enhanced version of the SNMP standard that includes improvements to SNMPv1 in the areas of protocol packet types, transport mappings, and MIB structure elements. SNMPv2c uses the SNMPv1 administration structure ("community based" and hence SNMPv2c). It is defined in RFC 1901, RFC 1905, and RFC 1906. SNMPv2c is defined in RFC 1901, "Introduction to Community-based SNMPv2;" RFC 1905, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2);" and RFC 1906, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)."

  Steel-Belted Radius does not support SNMP version 3 (SNMPv3).

A management information base (MIB) is a hierarchical collection of information that resides on a managed device. A MIB defines the types of information (objects) that can be controlled and collected by an NMS and includes thresholds, counters, tables, lists, and values. Managed objects consist of one or more object instances.

MIB objects can be read-only or read-write:

- A read-only object is a variable that can be read but not set from an NMS. For example, an NMS could read (but not increment) the value of a counter showing the number of packets received on the accounting port.
- A read-write object is a variable that can be set from an NMS. For example, an NMS could set the device name or IP address for an SNMP client.

Steel-Belted Radius supports the IETF-standard MIBs for RADIUS server authentication and accounting, as well as proprietary MIBs that record information about Steel-Belted Radius traps and operating statistics. **Table 33** lists the MIBs supported by Steel-Belted Radius.

Table 33: MIBs Supported by Steel-Belted Radius

MIB	MIB Origin	Function
rfc1155-smi.mib	IETF	Defines the structure of management information for SNMP version 1.

MIB	MIB Origin	Function
rfc1212.mib	IETF	Defines MIB syntax.
rfc1213.mib	IETF	Provides network management of TCP/IP-based internets (MIB-II).
rfc1215.mib	IETF	Provides SNMP trap definitions.
rfc2271.mib	IETF	Provides SNMP framework definitions.
rfc2618.mib	IETF	Provides SNMP framework definitions.
rfc2619.mib	IETF	Maintains authentication server statistics.
rfc2620.mib	IETF	Maintains accounting client statistics.
rfc2621.mib	IETF	Maintains accounting server statistics.
fnkrate.mib	Proprietary	Maintains Steel-Belted Radius rate statistics.
fnkradtr.mib	Proprietary	Defines the structure of Steel-Belted Radius traps (SNMP version 1).
fnkradtr-v2.mib	Proprietary	Defines the structure of Steel-Belted Radius traps (SNMP version 2c).
SNMPv2.mib	Proprietary	Defines traps about the state of the SNMP agent.
SNMPv2-CONF.mib	Proprietary	Defines conformance groups for SNMP version 2c.
SNMPv2-SMI.mib	Proprietary	Defines the structure of management information for SNMP version 2c.
SNMPv2-TC.mib	Proprietary	Describes text conventions for SNMP version 2c.

SNMP Messages

SNMP uses different types of messages to send and retrieve information.

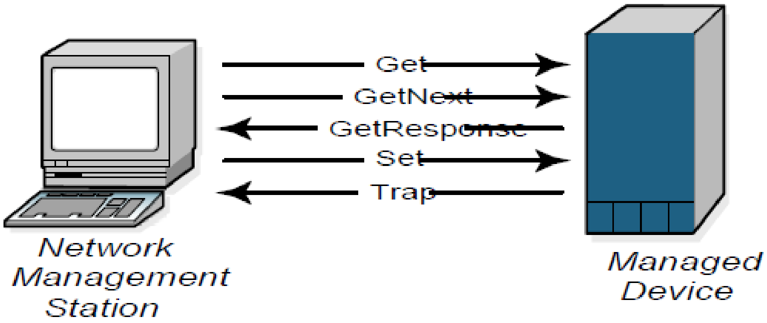
- A Get message requests the value of an object from a table or list maintained by a managed device. For example, a Get message might ask for the number of users since a device was restarted or the number of authentication requests that a server has received.
- A GetNext message requests the value of the next object instance from a table or list maintained by a managed device. GetNext messages let the NMS “walk” a list or table to retrieve MIB object values sequentially.
- A Get-Response message returns the information requested by a Get or GetNext message.

- A Set message sets the value of an object instance within a managed device.
- A Trap message notifies the NMS asynchronously when an important event, such as a change in state or a device or component failure, has occurred. For example, a managed device might send a trap message if the amount of space on the RADIUS server falls below a specified threshold or if the server cannot access its authentication database.

  The SNMP traps supported by Steel-Belted Radius are described in the fnkradtr.mib file. Traps are divided into three types:

  - **Informational** traps are sent to report important RADIUS information that is not an error or a warning, such as when the RADIUS server daemon is loaded or unloaded or when a threshold of some kind has resulted from a previous error or warning condition.
  - **Warnings** traps are sent to report RADIUS behavior that indicates a problem has occurred or may occur, such as when the RADIUS server is unable to connect to an external SQL database or when the file system is almost full. Many of these warning traps can be diluted or have configurable thresholds.
  - **Error** traps are sent to report RADIUS problems that have occurred, such as when the RADIUS server is unable to initialize one or more critical components on startup. Most Error traps indicate that the RADIUS server failed to start properly for some reason, such as the inability to allocate memory from the system. Most of these traps cannot be diluted.

Figure 167: SNMP Messages



Dilution and Threshold

Trap event dilution means you can configure Steel-Belted Radius so that a particular trap is sent to the NMS once for every n occurrences of the condition that generated that trap. This allows for a fine degree of control with respect to trap generation for certain warning and error conditions. Many of the traps defined in fnkradtr. mib can be diluted.

Some traps have configurable thresholds that allow you to set the lower and upper limits of acceptable behavior, and to generate different types of traps depending on the condition. For example, you can configure Steel-Belted Radius to send a warning trap message when if the count of available threads (for authentication and accounting) falls below 10, and to send an informational trap message when the count of available threads rises above 20.

SNMP trap event dilutions and thresholds are configured in the events.ini file, which resides in the RADIUS server directory. If you anticipate using SNMP traps, you should review the fnkradtr.mib file and the events.ini file to understand the options available to you.

SNMP Communities

An SNMP community defines an administrative relationship between a managed device and one or more management stations on your network. Each community has a name called the community string. The community string provides access control for SNMP objects. When an NMS sends a Get or Set message to a managed device that belongs to an SNMP community, it must include the appropriate community string in the request. If the community string in the request is correct, the managed device sends back the requested information. If the community string is incorrect, the managed device discards the request without responding.

Rate Statistics

Rate statistics variables defined in the fnkrate.mib file are derived from existing counter statistics by taking time into consideration. Three types of rate values are calculated for each of these counter statistics:

- Current-rate: the rate measured over the most recent rate interval
- Average-rate: the rate measured since startup, or the most recent statistics reset command
- Peak-rate: the highest rate observed since startup, or the most recent statistics reset command

The funkSbrRatesSecondsPerInterval read-only variable gives the duration in seconds of the interval over which the rate statistics are gathered.

Configuring SNMP

To configure the SNMP agent for Steel-Belted Radius, you must edit the pssnmpd.conf file to reflect your network environment. If you specify that the SNMP agent will use a non-default port number, you must also edit the SNMP port number in the testagent.sh file.

SNMP Files

The following files establish settings for configuring the SNMP support in Steel-Belted Radius. For more information about these files, refer to the Steel-Belted Radius Reference Guide.

Table 34: SNMP Files

File Name	Function
pssnmpd.conf	Stores settings for the Steel-Belted Radius SNMP agent.
testagent.sh	Test script that verifies the Steel-Belted Radius SNMP agent is operating correctly.

Editingpssnmpd.conf

Configuration settings for the SNMP agent are stored in the pssnmpd.conf file. After you install the SNMP agent for Steel-Belted Radius, you must modify the pssnmpd.conf configuration file to reflect your network environment.

Configuring Access Control

pssnmpd supports the View-Based Access Control Model (VACM) described in RFC 2575. To configure access control, you must map community names to security names, map security names to groups, and specify access rights and views for groups.

1. Set up mappings between community names and security names.

   Use the com2sec keyword to map each source/community pair to a security name. The com2sec entry is used to determine a security name from the traditional community string, taking into account where a request has come from.

The syntax for the com2sec keyword is:

```
com2sec security_name source community
```

where:

- security_name identifies the security name you want to create.
- source can be a hostname, an IP address, or the word default.
- community is an SNMP community string, which acts as a password to authenticate SNMP communications.

The first source/community combination that matches an incoming packet is selected. For example, the following configuration creates two security names (local and mynet) and maps them to two different subnet/community name pairs.

```
# sec.name          source              community
com2sec local       localhost           private
com2sec mynet      192.168.21.0/24      public
```

2. Map security names into group names.

   Use the group keyword to map security names into group names. The group keyword gives general control by mapping between a security name (for a particular protocol version), and the internal name used in the access line.

   The syntax for the group keyword is:

   ```
   group name model security
   ```

   where:

   - name is the name of an access group.
   - model identifies the security model you want to use: v1 or v2c.
   - security is a security name.

   For example, the following maps the two security names created in step 1. to four group/model pairs.

   ```
   # sec.                   model sec.          name
   group LocalGroup         v1                  local
   group LocalGroup         v2c                 local
   group LANGroup           v1                  mynet
   group LANGroup           v2c                 mynet
   ```

3. Create a view that gives access rights to the groups you created in step 2. .

   Use the view keyword to specify what portions of the MIB tree a specified group can view or modify. The syntax for the view keyword is:

view name {include | excluded} subtree [mask]

where:

- name is the identifier used for the view.
- included/excluded lets you include or exclude specific portions of the MIB tree from the view.
- subtree identifies the portion of the MIB tree that this name refers to in numeric or named form.
- mask specifies what elements of the MIB subtree should be regarded as relevant. The mask argument allows you to control access to specific rows in a table. When the entire MIB can be viewed, you can omit the mask field or enter ff.

4. Associate a group with the views you created in step 3. to grant the group access rights.

   Use the access keyword to specify who has access to part or all of the MIB tree.

   The syntax for the access keyword is:
   access name context model level prefix read write notify

   where:

   - name is the name of a group.
   - context specifies the context for the view. For SNMPv1 or SNMPv2c, context should be empty.
   - model is the security model: any, v1, or v2c.
   - level can be used to ensure that the request is authenticated or encrypted.
     For SNMPv1 or SNMPv2c, level should be noauth.
   - prefix specifies how the context setting should be matched against the context of the incoming PDU. Enter exact or prefix.
   - read specifies the view to be used for READ access.
   - write specifies the view to be used for WRITE access.
   - notify specifies the view to be used for NOTIFY access.

   For example, the following specifies that the local group uses the all view for READ, WRITE, and NOTIFY access.

```
#               context         sec.model       sec.level       prefix          read    write   notify
access local  “”                any             noauth          exact           mib2    all     all
```

Specifying System Contact Information

You can specify your system contact information in the pssnmpd.conf file or in the MIB. If you configure your system contact information in the pssnmpd.conf file, the objects are locked and cannot be modified via SNMP.

System contact information consists of the following:

- **syslocation—**The physical location of the managed device.
- **syscontact—**The person or department responsible for maintaining the managed device.

- **sysname—**The name of the managed device.

This information is stored in the system group of the MIB-II tree.

The syntax for specifying system contact information is:

```
syslocation string
syscontact  string
sysname string
```

Enabling Traps

Traps can be used by network entities to signal abnormal conditions to management stations. You should identify the NMS that receives trap messages generated by Steel-Belted Radius.

ⓘ Note: You can configure Steel-Belted Radius to use either SNMPv1 or SNMPv2c traps. You cannot configure Steel-Belted Radius to generate both types of traps simultaneously.

1. Use the trapcommunity keyword to specify the default community string to be used when sending traps.

   Syntax for the trapcommunity keyword is:

   ```
   trapcommunitystring
   ```

   The trapcommunity keyword must precede the trap2sink keyword in the pssnmpd.conf file.

2. Specify whether you want Steel-Belted Radius to use either SNMPv1 traps or SNMPv2c traps. Do not enable both types of traps at the same time.

   - If you want to use SNMPv1 traps, uncomment the trapsink keyword and specify the host or hosts to which Steel-Belted Radius should send SNMPv1 trap messages.

     Syntax for the trapsink keyword is:

     ```
     trapsink host [community [port]]
     ```

     where:

     host specifies the host name or IP address of the NMS.

     community specifies the community string the NMS expects.

   port specifies the UDP port on which the NMS is listening for SNMPv1 trap messages. Default is UDP port 162.

   For example:

   ```
   # send v1 traps
   trapsink nms.system.com secret
   ```

   - If you want to use SNMPv2c traps, uncomment the trap2sink keyword to specify the host or hosts to which Steel-Belted Radius should send SNMPv2c trap (notification) messages.

     Syntax for the trap2sink keyword is:

     ```
     trap2sink host [community [port]]
     ```

where:

host specifies the host name or IP address of the NMS.

community specifies the community string the NMS expects.

port specifies the port on which the NMS is listening for SNMPv2c trap messages.

For example:

```
# send v2 traps
trap2sink nms.system.com secret
```

Specifying the Persistent File Location

The SNMP agent uses the pssnmpd.conf file to store static agent configuration information, such as community strings. The SNMP agent uses the persist directory to store information set during the running of the agent, which needs to be persistent from one run to the next.

Use the persistentDir keyword in the [snmp] section of pssnmpd.conf to specify the location of the persist directory. By default, the persist directory is located in the /snmp directory within radiusdir on your server.

The syntax for specifying the location of the persist directory is as follows:

```
[snmp]
persistentDir   radiusdir/snmp/persist
```

Specifying the SNMP Ports

By default, pssnmpd listens for incoming SNMP requests on UDP port 161 on all IP interfaces. You can specify a different UDP port in the pssnmpd.conf file. The syntax for specifying a listening port is as follows:

```
[snmpd]
agentaddress port_number
```

Note: If you change the SNMP port number in pssnmpd.conf, you must also enter the same port number in testagent.sh.

Note: If you run more than one SNMP agent on your server, each agent must use a unique UDP port number.

Specifying Subagent Settings

By default, the SNMP subagent in Steel-Belted Radius communicates with the SNMP agent on TCP port 6669. If you change the port used for subagent-agent communication, you must modify pssnmpd.conf to uncomment the sbr_admin_parameters keyword and specify host, port, and interval values.

Syntax for the sbr_admin_parameters keyword is:

```
sbr_admin_parameters host=localhost port=port tryinterval=interval
```

where:

- **port** identifies the TCP port the Steel-Belted Radius server uses for SNMP communication. The default value is 6669.

- **interval** specifies the number of seconds information can remain in the SNMP subagent cache. If your SNMP management station will issue queries intermittently, you should set the tryinterval value to a small number (1-5) to ensure timely information. If your SNMP management station will poll the server periodically, you should set the tryinterval value to a larger number to avoid flooding the server with queries. The default is 10 seconds.

Specifying the radiusdir Location

Use the sbr_private_directory keyword to specify the location where Steel-Belted Radius is stored on your server.

Syntax for the sbr_private_directory keyword is:

```
sbr_private_directory radiusdir
```

The Steel-Belted Radius installer overwrites radiusdir with the appropriate value for your system. You should not change this value.

Starting the SNMP Agent

You can start the SNMP agent by executing the following at the command line:

```
/etc/init.d/init.pssnmpd start
```

Stopping the SNMP Agent

You can stop the SNMP agent by executing the following at the command line:

```
/etc/init.d/init.pssnmpd stop
```

Re-Reading the pssnmpd.conf File

You can force the pssnmpd agent to re-read its configuration files by sending a kill -HUP signal to the pssnmpd agent process.

Editing testagent.sh

You can run the testagent.sh script to verify that the pssnmpd SNMP agent is functioning. Before you do so, you must configure the testagent.sh file with the community string for your network.

The syntax for the testagent.sh file is as follows:

```
snmpget_path -M mib_location -c community port sysDescr
```

Table 35: testagent.sh Syntax

Keyword	Meaning
snmpget_path	Specifies the path for the snmpget utility. Default value is radiusdir/snmp/bin/snmpget.
-M mib_location	Specifies the path for the MIBs used by the SNMP agent. Default value is radiusdir/snmp/mibs.
-c community	Specifies the community string for your network. Default value is COMMUNITY.
port	Specifies the default port for SNMP traffic.

Keyword	Meaning
	Default value is localhost:161.
sysDescr	Specifies the MIB variable to be retrieved. Default value is system.sysDescr.0.

Using SNMP

Loading the MIBs

You will need to load the MIBs you are using into your SNMP management station. You can find all of the MIB files in radiusdir/snmp/mibs. The MIB files contain further details and may contain additional variables.

Before you can use the RADIUS and Steel-Belted Radius MIBs in your MIB browser, you may need to load certain common MIBs. The RADIUS and Steel-Belted Radius MIBs build upon these common MIBs.

rfc1155-smi.mib	Defines the structure of management information for SNMP version 1.
rfc2271.mib	An Architecture for Describing SNMP Management Frameworks
rfc1212.mib	Concise MIB definitions
rfc1215.mib	Convention for defining traps for use with the SNMP
SNMPv2-CONF.mib	Conformance Statements for SMIv2
SNMPv2-SMI.mib	Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)
SNMPv2-TC.mib	Textual Conventions for Transport Addresses

Using the snmpget Command

To use the snmpget command, you must set the LD_LIBRARY_PATH variable:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:radiusdir/snmp/bin    export    LD_LIBRARY_PATH
```

Use the snmpget command-line tool to obtain a scalar value. The following example combines the OID prefix with the name of the scalar:

```
radiusdir/snmp/bin/snmpget -M radiusdir/snmp/mibs -m all -c community host iso.org.dod.internet.
mgmt.mib-2.radiusMIB.radiusAuthentication.radiusAuthServMI    B.    radiusAuthServMIBObjects.
radiusAuthServ.radiusAuthServIdent
```

To execute a similar command, replace radiusdir, community, and host in thepreceding example with the values appropriate for your network, and put the entire command on a single line.

You can use the numerical equivalent for the OID prefix and a 0 suffix. For example:

```
radiusdir/snmp/bin/snmpget -M radiusdir/snmp/mibs -m all -c community host 1.3.6.1.2.1.67.1.1.1.1.1.0
```

Using the snmpwalk Command

To use the snmpwalk command, you must set the LD_LIBRARY_PATH variable:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:radiusdir/snmp/bin     export     LD_LIBRARY_PATH
```

Use the snmpwalk command-line tool to obtain a table. The following example uses the OID prefix of the table.

```
radiusdir/snmp/bin/snmpwalk -M radiusdir/snmp/mibs -m all -c community host iso.org.dod.internet.
mgmt.mib-2.radiusMIB.radiusAuthentication.radiusAuthServMI     B.     radiusAuthServMIBObjects.
radiusAuthServ.radiusAuthClientTable.        radiusAuthClientEntry
```

To execute a similar command, replace radiusdir, community, and host in the preceding example with the values appropriate for your network, and put the entire command on a single line.

You can use the numerical equivalent for the OID prefix and a 0 suffix. For example:

```
radiusdir/snmp/bin/snmpwalk -M radiusdir/snmp/mibs -m all -c community host
1.3.6.1.2.1.67.1.1.1.1.15.1
```

Resetting Rate Statistics

Although SNMP statistics are automatically reset if you restart Steel-Belted Radius, you can request that all statistics be reset to zero without having to restart the server. To do so, issue the following command:

```
kill -USR2 pIdServer
```

where pIdServer is the process ID of your Steel-Belted Radius server.

Troubleshooting

- The SNMP log file (radiusdir/snmp/pssnmpd.log) records event information relating to the SNMP agent.
- If the SNMP agent fails to run (that is, you do not see it listed if you run the ps -A -f | grep pssnmpd command), review the log file to determine whether another SNMP agent is running.
- Two SNMP agents cannot share the same port. If another SNMP that uses the same port (port 161 by default) is running on your system, the Steel-Belted Radius SNMP agent may fail to start. Review the log file to determine whether a conflict over Port 161 is occurring.
- If the Steel-Belted Radius SNMP agent conflicts with another SNMP agent and you cannot uninstall or disable the other SNMP agent, you should assign it a different port.
- If an SNMP MIB browser or other management station cannot reach the SNMP agent, verify that the agent is running by issuing the ps -A -f | grep pssnmpd command. If the agent is listed, run the testagent.sh script to verify the agent is responding.
    - If the testagent.sh script succeeds, verify that the community strings and access controls configured in the pssnmpd.conf file are correct.
    - If the testagent.sh script fails, verify that the script uses the same port number specified in the pssnmpd.conf file (the default is Port 161) and that the pssnmpd.conf file grants access to localhost, which testagent.sh requires to function.

Chapter 28
Configuring Replication via Legacy SBR Administrator

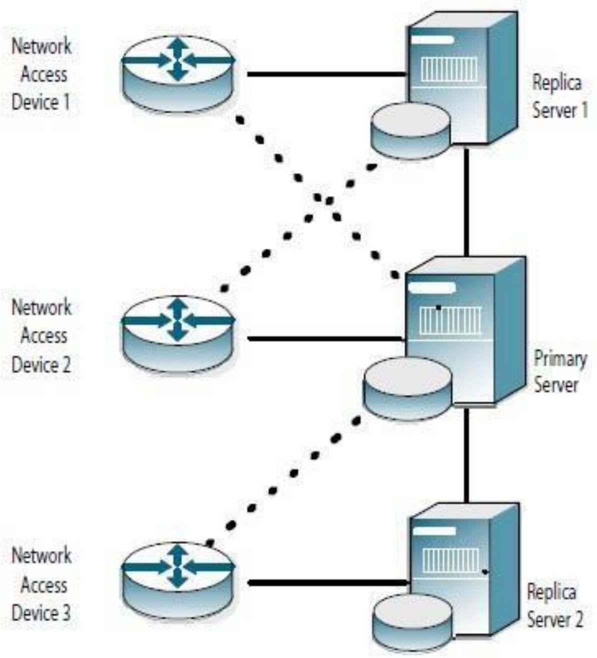
This chapter describes how to configure and use the centralized configuration management (CCM) feature to coordinate Steel-Belted Radius server settings in a replication environment via legacy SBR administrator.

About Replication

Steel-Belted Radius supports the replication of RADIUS configuration data from a primary server to one or more replica servers within a replication realm. Replication provides administrators with an easy way to configure multiple servers that require the same information. Depending on network configuration, you can use replication to increase AAA capacity, balance AAA traffic across RADIUS servers, or ensure that authentication services are not interrupted if access to a primary or replica server is interrupted (redundancy).

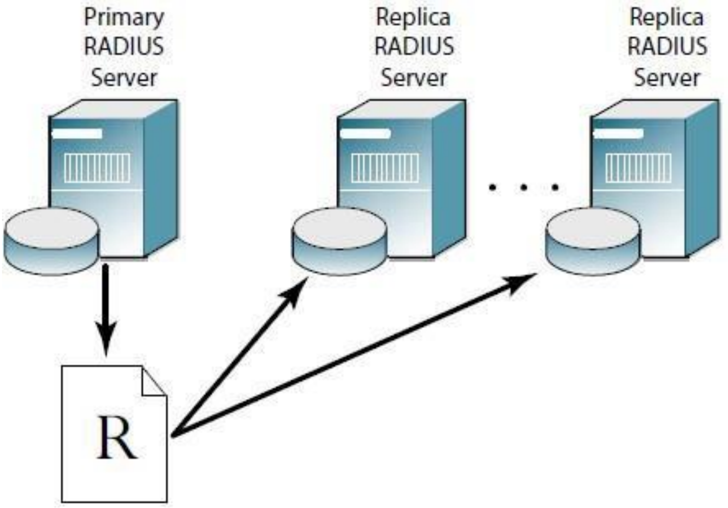
Figure 168: Using Replication for Load Balancing and Redundancy illustrates an environment where RADIUS traffic is load-balanced by configuring each network access device to authenticate users through a different RADIUS server (solid line). If a RADIUS server becomes unavailable, the NAD can fail over to its backup RADIUS server (dotted line).

Figure 168: Using Replication for Load Balancing and Redundancy



All the servers within a realm reflect the current configuration specified by the network administrator: the network administrator modifies the configuration on the primary server, and the primary server propagates the new configuration to its replica servers. For example, after a network administrator configures a new RADIUS client or profile on the primary server, the network administrator tells the primary server to publish a date-stamped configuration package file that contains the updated configuration information. After publication, the primary server notifies each replica server that a new configuration package is ready. Each replica server then downloads and installs the configuration package to update its settings.

Figure 169: Publication and Distribution of Replication Packages



The primary server maintains a list of the replica servers that have registered with it. The primary server use this list to track which servers to notify after it publishes an updated configuration package to resynchronize the configuration of replica servers.

Note: You should limit access to the directory in which you store configuration packages on Windows servers to the CREATOR OWNER, SYSTEM, and Administrators. To set file access permissions for the \ Radius\Service directory, right-click the directory icon, click the Security tab, click the Allow and Deny check boxes to limit access to authorized users.

By default, file permissions for configuration packages on Linux servers are set to rw-rw----, which excludes users other than the file owner and the owner's group from displaying the contents of file packages.

If the primary server needs to be taken out of service for an extended period, the network administrator promotes one of the replica servers to be the new primary server. Thereafter, the other replica servers copy the configuration package from the promoted primary server.

The following types of information are included in a replication package.

Serverinformation

- RADIUS client information
- User information
- Profile information
- Proxy target information
- EAP method configurations
- Filters
- RADIUS tunnel information
- Name parsing information
- Authentication method information
- Authentication realm information
- Rejection messages
- Javascript (.jsi) files

You administer this information by connecting the SBR Administrator to the primary server: the information is propagated to the replica servers in the domain. (If you connect the SBR Administrator to a replica server, you can view this information, but you cannot modify it.)

The following types of information are not included in a replication package:

- Address pool information—You administer address pools for a server by connecting the SBR Administrator to that server. Because an address must not be assigned to two users at the same time, each server in a realm must have its own address pools, and these pools must not overlap.
- Administrator information—Administrator information must be configured for each primary and replica server separately.
- Statistics information—Server statistics are not replicated. You can view statistics for replica servers when you connect SBR Administrator to the primary server.
- Report information—Report information is not replicated. To obtain report information for a primary or replica server, connect SBR Administrator to the applicable server.
- Steel-Belted Radius configuration files— Configuration files (*.ini files (other than filter.ini and eap.ini), *.aut files (other than peapauth.aut, ttlsauth.aut, tlsauth.aut, and talsauth.eap), and *.dir files are not replicated. When you change configuration files on the primary server, you must copy the modified files to the appropriate directory on each replica server.

Note: Configuration packages are retained until they are replaced. An old configuration package is automatically deleted 24 hours after a new configuration package is published.

Replication Requirements

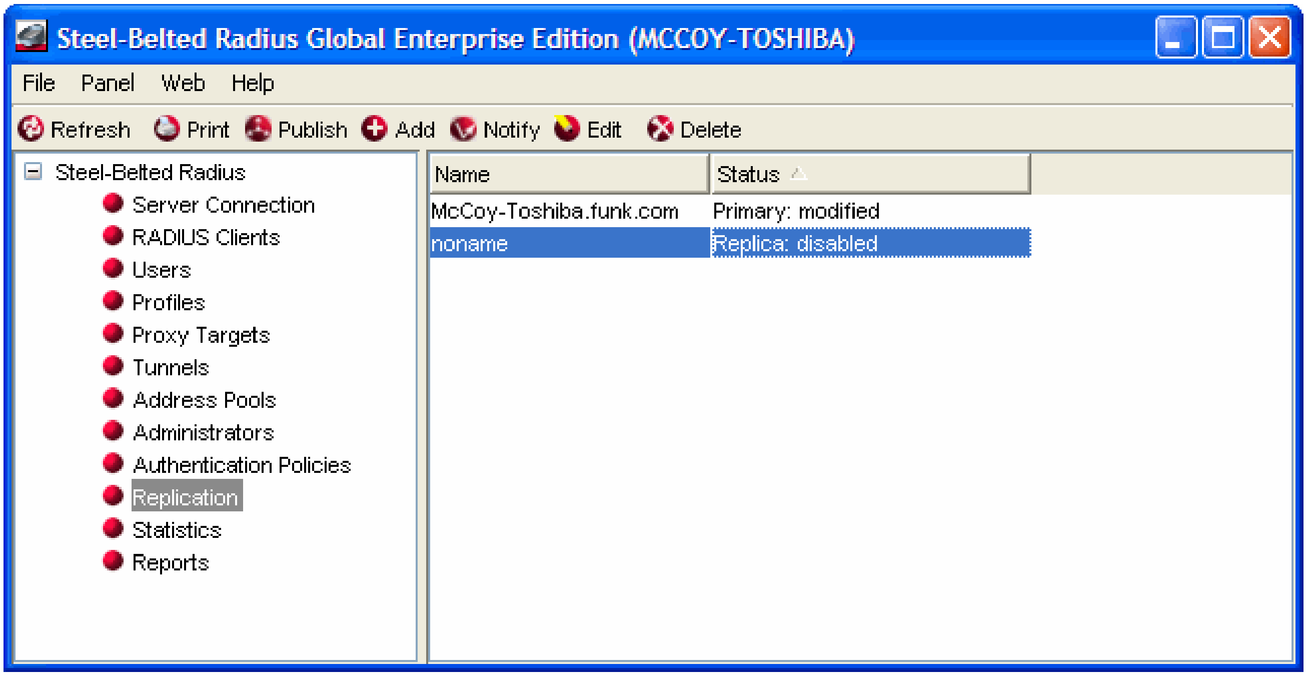
Servers in a replication realm must comply with the following requirements.

- All servers in a replication realm must be running the same operating system (Windows or Linux).
- All servers in a replication realm must be running the same version and edition (Global Enterprise Edition, Service Provider Edition, or Enterprise Edition) of Steel-Belted Radius.
- All servers in a replication realm must be configured to support the same types of users (domain, host, RSA SecurID, TACACS+, or UNIX).
- If RSA SecurID is enabled on the primary server, RSA SecurID must be enabled on the replica servers, and all servers in the realm must have consistent sdconf.rec files.
- The system clocks on servers in a replication realm must by synchronized to within 10 minutes of one another and their time zones must be configured correctly. Steel-Belted Radius uses the system clock value and time zone setting to convert local time to Universal Time Coordinated (also known as Greenwich Mean Time) when evaluating synchronization. If possible, you should use a Network Time Protocol (NTP) server to set the system clock on all servers automatically.
- All servers in a replication realm must use the same TCP port to exchange replication information. The default port for replication communication is TCP 1812, though you can specify another port for replication traffic by modifying the radius.ini file.
- If a firewall stands between servers in a replication realm, the firewall must be configured to pass traffic on the port used for replication communication.

Configuring Replica Servers

The Replication panel (Figure 170: Replication Panel) lets you add servers to a replication realm, initiate publication of a replication package by a primary server, and notify replica servers that they should download and install a new replication package.

Figure 170: Replication Panel



Name	Status
McCoy-Toshiba.funk.com	Primary: modified
noname	Replica: disabled

Adding a Replica Server

In most situations, you add a replica server to a realm as follows:

1. Copy the replica.ccmpkg configuration package file from the primary server to a directory on the host you want to add as a replica server.

   Note: The replica.ccmpkg file contains sensitive information, and should not be stored in a publicly accessible location, such as a file server or shared directoryInstall the Steel-Belted Radius server software on the host you want to add as a replica server.

2. When the installer (Windows) or configuration script (Linux) asks what kind of server you are installing, choose **Replica** and, when prompted, enter the path to the replica.ccmpkg file.

3. Restart the host you want to add as a replica server.

   The replica server registers itself with the primary server automatically after it is restarted. Thereafter, the replica server automatically connects to the primary server once an hour to check whether an updated configuration package is available.

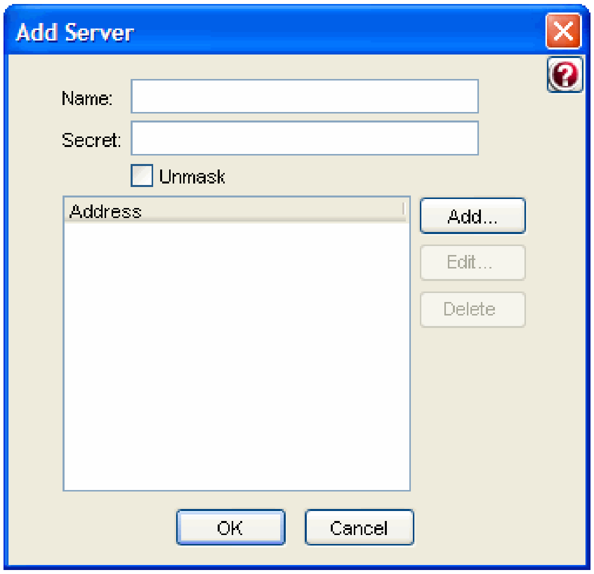
In some circumstances, however, you may want to add a replica server to the server list on the primary server manually so that it shows up immediately. To register a replica server manually:

1. Run SBR Administrator and connect to the primary server.

2. Open the Replication panel.

3. Click the **Add** button.
   The Add Server dialog opens.

Figure 171: Add Server Dialog



4. Enter the name of the RADIUS server in the **Name** field.

   Although you can assign any name to a RADIUS server, you should use the device's hostname to avoid confusion.

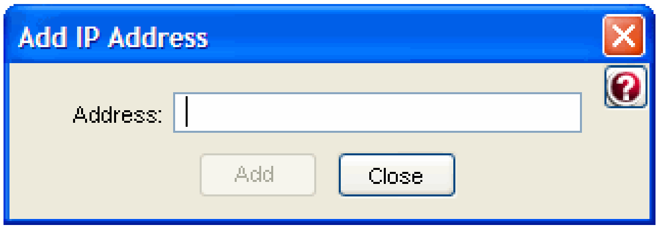
5. Enter the replication secret for the RADIUS server in the **Secret** field.

   For privacy, asterisks are echoed as you type. You can click the **Unmask** check box to display the characters in the shared secret.

6. Enter one or more IP addresses for your server.
   a. Click the **Add** button.

   b. When the Add IP Address dialog opens, enter an IP address you want to associate with the server in the **Address** field and click **Add**.

Figure 172: Add IP Address Dialog



   c. Repeat Step 5b until you have finished adding IP addresses for the server.

   d. Click **Close.**

Enabling a RADIUS Server

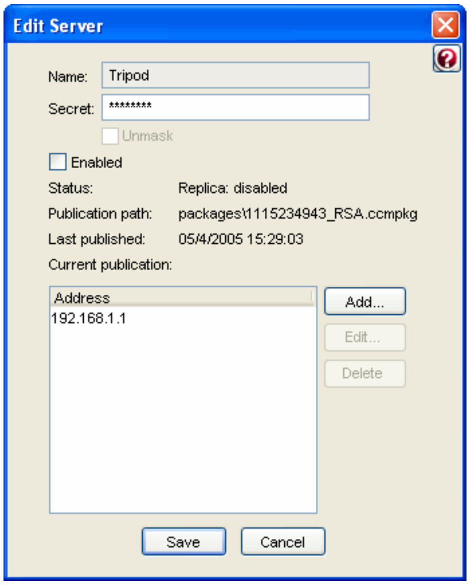
To enable a RADIUS server:

1. Open the Replication panel.

2. Select the RADIUS server you want to enable and click the **Edit** button (or double-click the RADIUS server entry).

The Edit Server dialog opens.

Figure 173: Edit Server Dialog



Edit Server

Name: Tripod

Secret: ********

□ Unmask

□ Enabled

Status: Replica: disabled

Publication path: packages\1115234943_RSA.ccmpkg

Last published: 05/4/2005 15:29:03

Current publication:

Address
192.168.1.1

3. Click the **Enabled** check box.

4. Click the **Save** button.

Deleting a RADIUS Server

To delete a replica server from a realm:

1. Open the Replication panel.

2. Select the RADIUS server entry you want to delete.

3. Click the Delete button on the SBR Administrator toolbar.

4. When you are prompted to confirm the deletion request, click Yes.

Publishing Server Configuration Information

If you change the configuration of your primary server, you must publish the modified configuration so that your replica servers can download the modified settings.

To publish server configuration information:

1. Open the Replication panel.

2. Click the Publish button on the toolbar.

   This creates a file called /radius/packages/timestamp_RSA.ccmpkg (Linux) or \Radius\Service\packages\timestamp_RSA.ccmpkg (Windows), where timestamp reflects the date and time the package was created.

Notifying Replica RADIUS Servers

Under normal circumstances, a replica server connects to its primary server every hour to check whether a new replica.ccmpkg file has been published. If necessary, a network administrator can manually notify a replica server to download and install the current configuration package from the primary server. Manual notification is useful when network issues prevent the automatic download and installation of a configuration package when it is first published, and the configuration on the replica no longer matches the configuration on the primary server.

Note: You can display the Replication panel to determine the status of your replica servers.

To notify replica servers that new configuration information has been published:

1. Open the Replication panel.
2. Select the replica server you want to notify.
3. Click the Notify button on the toolbar.

   The replica server downloads and installs its configuration package from the primary server. After the package is installed, the replica server is resynchronized with the primary server.

Designating a New Primary RADIUS Server

You can change which server within a realm is designated as the primary server for that realm.

To designate a new primary server:

1. Stop the RADIUS service on the replica server.
2. Log into the replica server as root.
3. Open a command window and navigate to the \Radius\Service directory (Windows) or /opt/PSsbr/radius directory (Linux).
4. Run the sbrsetuptool utility with the promote option.

   ```
   # sbrsetuptool -promote
   ```

   The utility creates a configuration package to change this server to the primary server.
5. Restart the updated replica server to make it the new primary server.
6. Publish a new configuration package administratively to configure all replica servers to use the new primary server.

After you designate a new primary server for a realm, the old primary server becomes a replica server automatically.

Recovering a Replica After a Failed Download

If a replica server fails during the download of a configuration package, its configuration may be corrupted or it may have a stale secret.

To recover after a failed download:

1. Stop the RADIUS service on the replica server.
2. Log into the replica server as root.
3. Open a command window and navigate to the \Radius\Service directory (Windows) or /opt/PSsbr/radius directory (Linux).

4. Run the sbrsetuptool utility with the identity option and information on where to download configuration information.

   To obtain configuration from a configuration package, issue the following command:

   # sbrsetuptool -identity REPLICA -reppkg pathname

   where pathname specifies the path to a replica.ccmpkg package.

   To obtain configuration from the primary server for the realm, issue the following command:

   # sbrsetuptool -identity REPLICA -primary name address secret

   where name specifies the DNS name of the primary server, address specifies the IP address of the primary server, and secret specifies the shared secret used to authenticate configuration downloads.

5. Restart the updated replica server so that it can load its new configuration.

After the replica server is restarted, it will be re-synchronized with the current primary server.

Changing the Name or IP Address of a Server

You may need to change the DNS name or IP address assigned to a primary or replica server if your network changes.

To change the DNS name or IP address of a primary or replica server:

1. Stop the RADIUS service on the RADIUS server you want to change.
2. Log into the RADIUS server as root.
3. Open a command window and navigate to the \Radius\Service directory (Windows) or /opt/PSsbr/radius directory (Linux).
4. Run the sbrsetuptool (Linux) utility with the identity option. To

   rename a primary server, enter the following command:

   # sbrsetuptool -identity PRIMARY

   To rename a replica server, enter the following command:

   # sbrsetuptool -identity REPLICA

5. Restart the updated server so that it can load its new configuration.
6. Run the SBR Administrator and modify the DNS name or IP address for the server you want to rename. Verify that the secret on the renamed server is correct.

   You may need to use the Replication panel to delete the old server name from the list of servers in the realm.

7. Publish the modified configuration to propagate the name change to the replica servers.

Replication Error Messages

The following tables list possible causes for error messages caused by replication issues.

Error Messages on Replica Servers

Table 36 lists possible causes for error messages on replica servers in a replication realm.

Table 36: Error Messages on Replica Servers

Error Type	Error Message	Possible Cause
Post Errors (Errors with Notification from Primary)	CRadManagedServerNotifyPost:: ExecutePost invalid signature!	Mismatched replication secret.
Post Errors (Errors with Notification from Primary)	CRadManagedServerNotifyPost:: ExecutePost invalid sequence number	Two posts have the same sequence number. The clocks on the primary and replica are more then 10 minutes apart.
	CRadManagedServerNotifyPost:: ExecutePost decrypt failed	Shared secret failed to decrypt. Bad Replication Secret secret.
	CRadManagedServerNotifyPost:: ExecutePost invalid <body> missing parameters	Post had an invalid xml request.
Update Errors (Errors with Published package from Primary)	CRadManagedServerUpdate:: DoStart Failed to open 'file_name' for writing	Temp directory does not exist. Temp directory or file have incorrect permissions.
	CRadManagedServerUpdate:: StartUpdates has already started update	Update is already in progress.
	CRadManagedServerUpdate:: DownloadPackage HTTP POST error:errCode Primary ID	Error in transmitting request to Primary (timeout during transmit).
	CRadManagedServerUpdate:: DownloadPackage HTTP headers parsing error	Error in receiving package. Typically caused by a timeout during receive resulting from an invalid package.
	CRadManagedServerUpdate:: DownloadPackage connection primary IP Addr error: errCode Primary ID	Replica failed to connect with Primary. Primary not running.
	CRadManagedServerUpdate:: DownloadPackage exceeded iterations limit while communicating with CCM	Update failed after three attempts.
	CRadManagedServerUp date:: ProcessPackage signature mismatch	Secrets on Replica and Primary do not match.
	CRadManagedServerUpdate:: ProcessPackage CCM error: 'Error String' 'Pararmeter'	Error parsing downloaded packages.
	CRadManagedServerUpdate:: ProcessPackage Failed to open \'" << file_name << "\' for writing	Temp directory does not exist. Temp directory or file has incorrect permissions.
	CRadManagedServerUpdate:: ProcessPackage thumbprint mismatch	Invalid package. Republish the package.

Error Type	Error Message	Possible Cause
Proxy Errors (Statistics retrieve errors)	CRadProxyPost:: ExecutePost invalid signature!	Mismatched replication secret.
	(Statistics retrieve errors)	Two posts have the same sequence number. The clocks on the primary and replica are more than 10 minutes apart.
		Shared secret failed to decrypt. Bad Replication Secret secret.
	CRadProxyPost:: ExecutePost invalid <body> missing parameters	Post had an invalid XML request.

Error Messages on Primary Servers

Table 37 lists possible causes for error messages on primary servers in a replication realm.

Table 37: Error Messages on Primary Servers

Error Type	Error Message	Possible Cause
Notify Target (Both Notify and Publish send a notification)	CRadConfigManagedServerHTTP Notification::NotifyTarget failed to fetch	Replica does not exist in database. This can occur if two administrators are running instances of SBR Administrator, one administrator deletes a replica, then the other administrator tries
	CRadConfigManagedServerHTTP Notification::NotifyTarget failed replicaId	Notify failed to communicate with replica, Replica is not running, or check Replica DCF log for more information.
Publication Provider (requests from GUI to Notify or Publish)	CRadConfigPublicationProvider:: UpdateResource notify invoked when not Primary	Attempted to Notify as a Replica, this is only allowed from a Primary.
	CRadConfigPublicationProvider:: UpdateResource publish invoked when not Primary	Replica attempted to publish; publication is permitted only from a Primary.
Publication Post (parsing of Post from replica to get data)	CRadConfigServerProviderPost:: ExecutePost signature mismatch with server:	Replica and Primary have different replication secrets.
	CRadConfigServerProvider:: GetResource invoked when not Primary	Another Replica is requesting a download from this server which is a replica. Replica that is requesting a download needs to be reconfigured (see disaster recovery in docs).
Proxy Errors (Statistics retrieve errors)	CRadProxyClient:: Send failed to fetch	Replica does not exist in database. This can occur if two administrators are running instances of SBR Administrator, one administrator deletes a replica, then the other administrator tries to publish to that replica.
	CRadProxyClient:: SendData HTTP POST error:	Connection error with replica.

Chapter 29
Configuring Replication via WebGUI

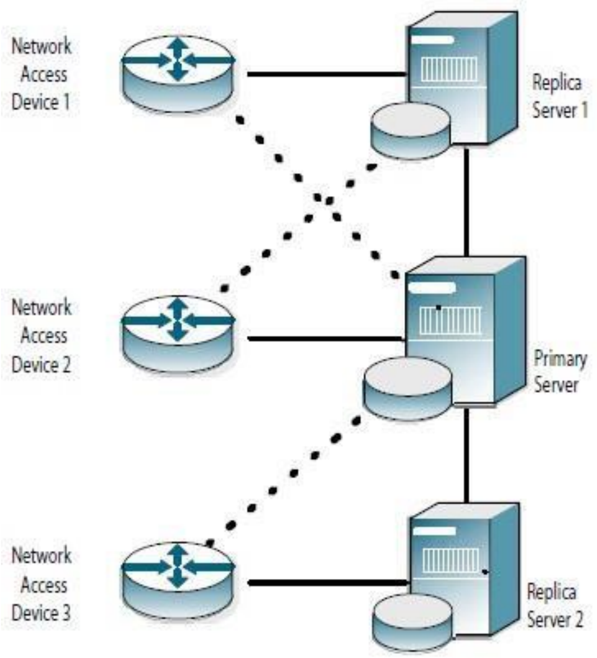
This chapter describes how to configure and use the centralized configuration management (CCM) feature to coordinate Steel-Belted Radius server settings in a replication environment via WebGUI.

About Replication

Steel-Belted Radius supports the replication of RADIUS configuration data from a primary server to one or more replica servers within a replication realm. Replication provides administrators with an easy way to configure multiple servers that require the same information. Depending on network configuration, you can use replication to increase AAA capacity, balance AAA traffic across RADIUS servers, or ensure that authentication services are not interrupted if access to a primary or replica server is interrupted (redundancy).

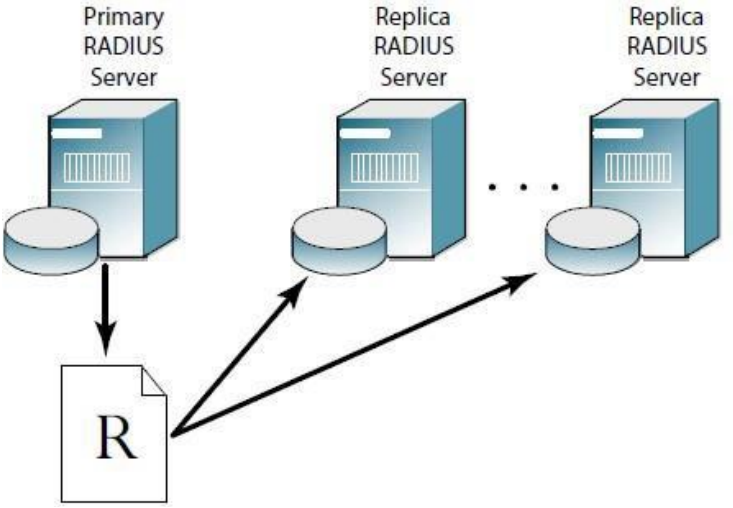
Figure 174: Using Replication for Load Balancing and Redundancy illustrates an environment where RADIUS traffic is load-balanced by configuring each network access device to authenticate users through a different RADIUS server (solid line). If a RADIUS server becomes unavailable, the NAD can fail over to its backup RADIUS server (dotted line).

Figure 174: Using Replication for Load Balancing and Redundancy



All the servers within a realm reflect the current configuration specified by the network administrator: the network administrator modifies the configuration on the primary server, and the primary server propagates the new configuration to its replica servers. For example, after a network administrator configures a new RADIUS client or profile on the primary server, the network administrator tells the primary server to publish a date-stamped configuration package file that contains the updated configuration information. After publication, the primary server notifies each replica server that a new configuration package is ready. Each replica server then downloads and installs the configuration package to update its settings.

Figure 175: Publication and Distribution of Replication Packages



The primary server maintains a list of the replica servers that have registered with it. The primary server use this list to track which servers to notify after it publishes an updated configuration package to resynchronize the configuration of replica servers.

Note: You should limit access to the directory in which you store configuration packages on Windows servers to the CREATOR OWNER, SYSTEM, and Administrators. To set file access permissions for the \Radius\Service directory, right-click the directory icon, click the Security tab, click the Allow and Deny check boxes to limit access to authorized users.

By default, file permissions for configuration packages on Linux servers are set to rw-rw----, which excludes users other than the file owner and the owner's group from displaying the contents of file packages.

If the primary server needs to be taken out of service for an extended period, the network administrator promotes one of the replica servers to be the new primary server. Thereafter, the other replica servers copy the configuration package from the promoted primary server.

The following types of information are included in a replication package.

Server information

- RADIUS client information
- User information
- Profile information
- Proxy target information
- EAP method configurations
- Filters
- RADIUS tunnel information
- Name parsing information
- Authentication method information
- Authentication realm information
- Rejection messages
- Javascript (.jsi) files

You administer this information by connecting the SBR Administrator to the primary server: the information is propagated to the replica servers in the domain. (If you connect the SBR Administrator to a replica server, you can view this information, but you cannot modify it.)

The following types of information are not included in a replication package:

- Address pool information—You administer address pools for a server by connecting the SBR Administrator to that server. Because an address must not be assigned to two users at the same time, each server in a realm must have its own address pools, and these pools must not overlap.
- Administrator information—Administrator information must be configured for each primary and replica server separately.
- Statistics information—Server statistics are not replicated. You can view statistics for replica servers when you connect SBR Administrator to the primary server.
- Report information—Report information is not replicated. To obtain report information for a primary or replica server, connect SBR Administrator to the applicable server.
- Steel-Belted Radius configuration files— Configuration files (*.ini files (other than filter.ini and eap.ini), *.aut files (other than peapauth.aut, ttlsauth.aut, tlsauth.aut, and talsauth.eap), and *.dir files are not replicated. When you change configuration files on the primary server, you must copy the modified files to the appropriate directory on each replica server.

  Note: Configuration packages are retained until they are replaced. An old configuration package is automatically deleted 24 hours after a new configuration package is published.

Replication Requirements

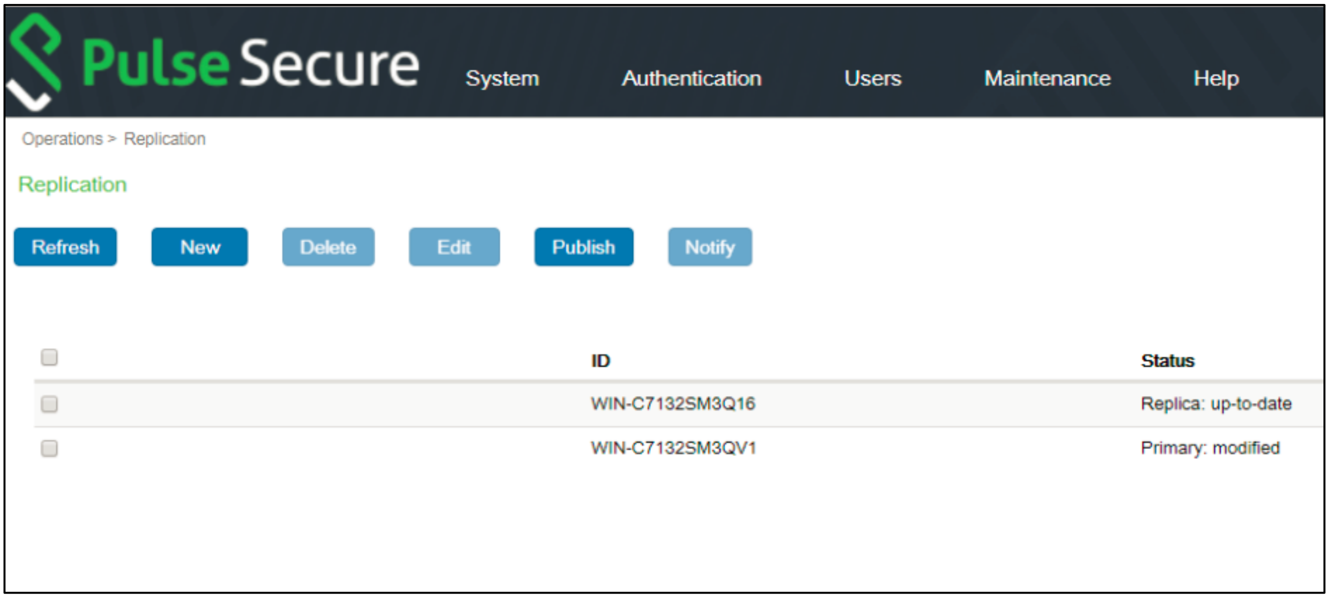
Servers in a replication realm must comply with the following requirements.

- All servers in a replication realm must be running the same operating system (Windows or Linux).
- All servers in a replication realm must be running the same version and edition (Global Enterprise Edition, Service Provider Edition, or Enterprise Edition) of Steel-Belted Radius.
- All servers in a replication realm must be configured to support the same types of users (domain, host, RSA SecurID, TACACS+, or UNIX).
- If RSA SecurID is enabled on the primary server, RSA SecurID must be enabled on the replica servers, and all servers in the realm must have consistent sdconf.rec files.
- The system clocks on servers in a replication realm must by synchronized to within 10 minutes of one another and their time zones must be configured correctly. Steel-Belted Radius uses the system clock value and time zone setting to convert local time to Universal Time Coordinated (also known as Greenwich Mean Time) when evaluating synchronization. If possible, you should use a Network Time Protocol (NTP) server to set the system clock on all servers automatically.
- All servers in a replication realm must use the same TCP port to exchange replication information. The default port for replication communication is TCP 1812, though you can specify another port for replication traffic by modifying the radius.ini file.
- If a firewall stands between servers in a replication realm, the firewall must be configured to pass traffic on the port used for replication communication.

Configuring Replica Servers

The Replication page (Figure 176: Replication Page) lets you add servers to a replication realm, initiate publication of a replication package by a primary server, and notify replica servers that they should download and install a new replication package.

Figure 176: Replication Page



Operations > Replication

Replication

□	ID	Status
□	WIN-C7132SM3Q16	Replica: up-to-date
□	WIN-C7132SM3QV1	Primary: modified

Adding a Replica Server

In most situations, you add a replica server to a realm as follows:

1. Copy the replica.ccmpkg configuration package file from the primary server to a directory on the host you want to add as a replica server.

   Note: The replica.ccmpkg file contains sensitive information, and should not be stored in a publicly accessible location, such as a file server or shared directoryInstall the Steel-Belted Radius server software on the host you want to add as a replica server.

2. When the installer (Windows) or configuration script (Linux) asks what kind of server you are installing, choose **Replica** and, when prompted, enter the path to the replica.ccmpkg file.

3. Restart the host you want to add as a replica server.

   The replica server registers itself with the primary server automatically after it is restarted. Thereafter, the replica server automatically connects to the primary server once an hour to check whether an updated configuration package is available.

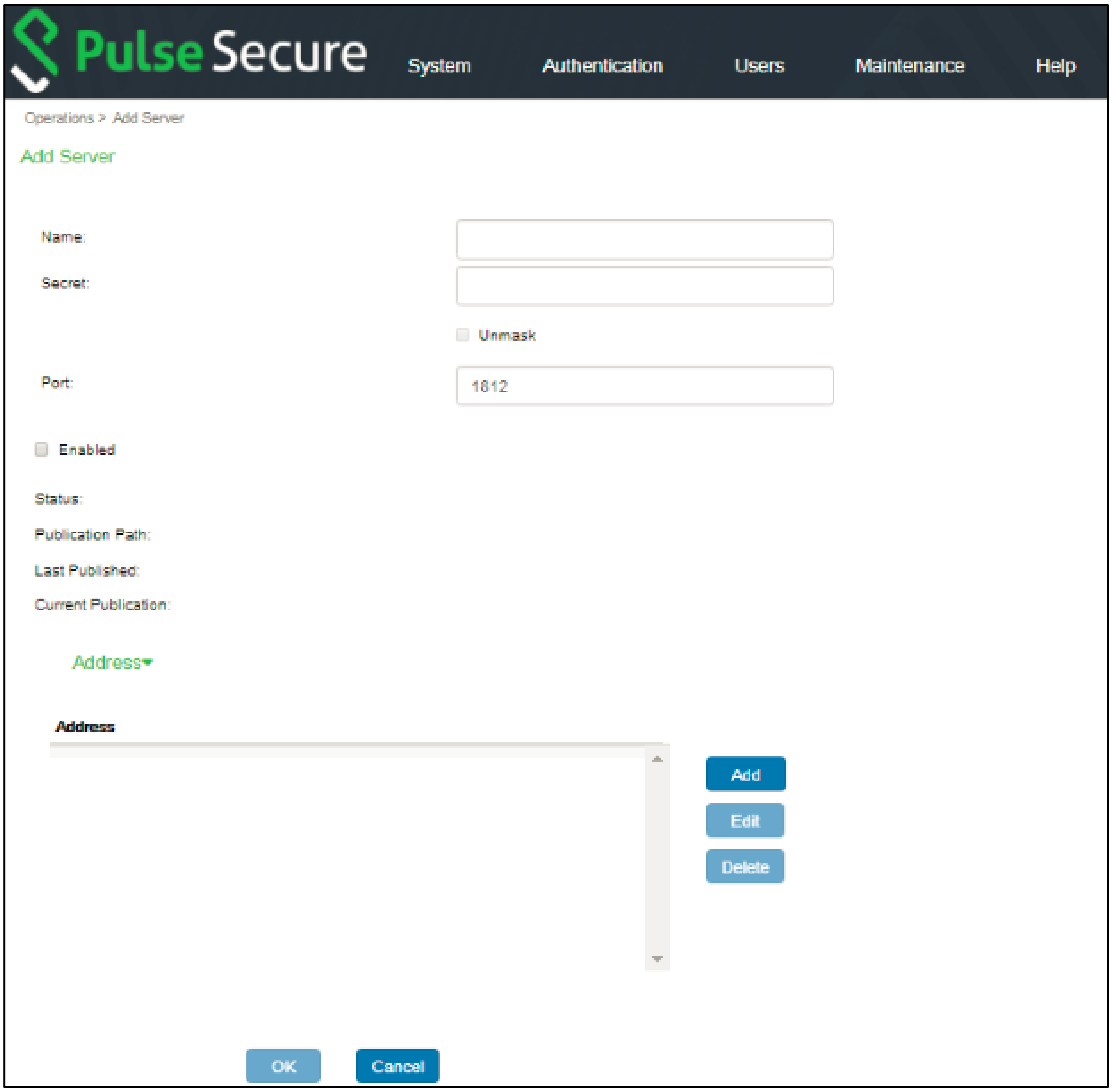
In some circumstances, however, you may want to add a replica server to the server list on the primary server manually so that it shows up immediately. To register a replica server manually:

1. Run SBR Administrator and connect to the primary server.

2. Choose Maintenance > Operations > Replication to open the Replication page.

3. Click the **New** button.
   The New Server page opens.

Figure 177: New Server Page



4. Enter the name of the RADIUS server in the **Name** field.

   Although you can assign any name to a RADIUS server, you should use the device's hostname to avoid confusion.

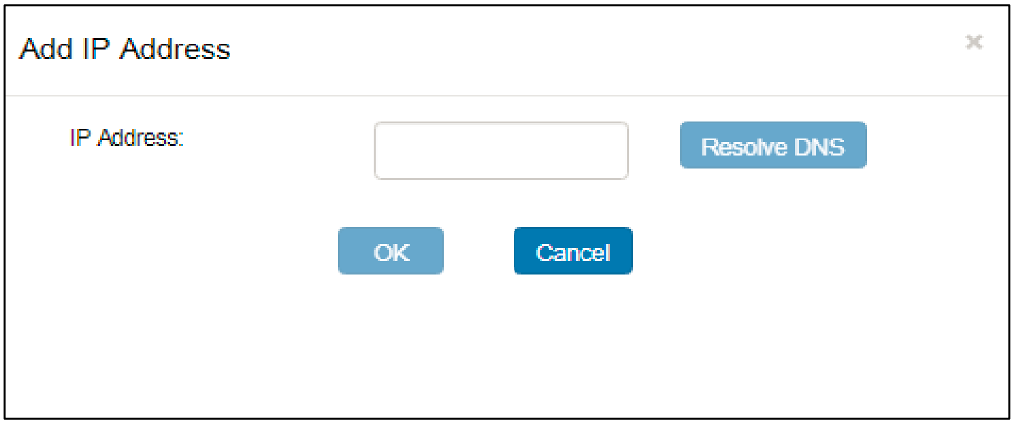
5. Enter the replication secret for the RADIUS server in the **Secret** field.

   For privacy, asterisks are echoed as you type. You can click the **Unmask** check box to display the characters in the shared secret.

6. Enter one or more IP addresses for your server.
   a. Click the **Add** button.

   b. When the Add IP Address page opens, enter an IP address you want to associate with the server in the **Address** field and click **Add**.

Figure 178: Add IP Address Page



c. Repeat Step 5b until you have finished adding IP addresses for the server.

d. Click **Close.**

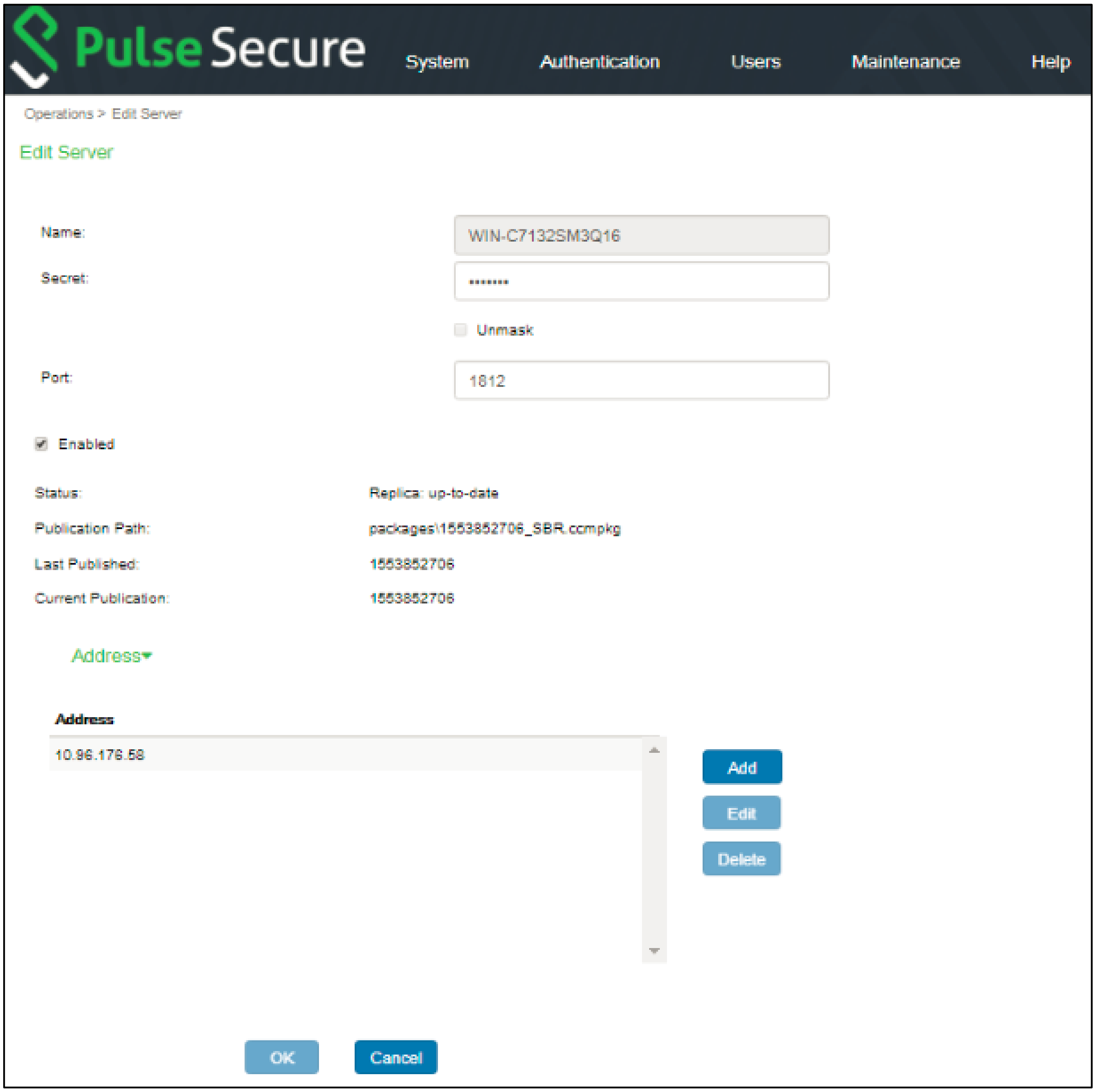
Enabling a RADIUS Server

To enable a RADIUS server:

1. Choose Maintenance > Operations > Replication to open the Replication page.
2. Select the RADIUS server you want to enable and click the **Edit** button.

   The Edit Server page opens.

Figure 179: Edit Server Page



Name: WIN-C7132SM3Q16

Secret: •••••••

□ Unmask

Port: 1812

☑ Enabled

Status: Replica: up-to-date

Publication Path: packages\1553852706_SBR.ccmpkg

Last Published: 1553852706

Current Publication: 1553852706

Address
10.96.176.58

3. Click the **Enabled** check box.

4. Click the **Save** button.

Deleting a RADIUS Server

To delete a replica server from a realm:

1. Choose Maintenance > Operations > Replication to open the Replication page.
2. Select the RADIUS server entry you want to delete.
3. Click the Delete button.
4. When you are prompted to confirm the deletion request, click Yes.

Publishing Server Configuration Information

If you change the configuration of your primary server, you must publish the modified configuration so that your replica servers can download the modified settings.

To publish server configuration information:

1. Choose Maintenance > Operations > Replication to open the Replication Page.
2. Click the Publish button.

   This creates a file called /radius/packages/timestamp_RSA.ccmpkg (Linux) or \Radius\Service\packages\timestamp_RSA.ccmpkg (Windows), where timestamp reflects the date and time the package was created.

Notifying Replica RADIUS Servers

Under normal circumstances, a replica server connects to its primary server every hour to check whether a new replica.ccmpkg file has been published. If necessary, a network administrator can manually notify a replica server to download and install the current configuration package from the primary server. Manual notification is useful when network issues prevent the automatic download and installation of a configuration package when it is first published, and the configuration on the replica no longer matches the configuration on the primary server.

ℹ Note: You can display the Replication page to determine the status of your replica servers.

To notify replica servers that new configuration information has been published:

1. Choose Maintenance > Operations > Replication to open the Replication page.
2. Select the replica server you want to notify.
3. Click the Notify button.

   The replica server downloads and installs its configuration package from the primary server. After the package is installed, the replica server is resynchronized with the primary server.

Designating a New Primary RADIUS Server

You can change which server within a realm is designated as the primary server for that realm.

To designate a new primary server:

1. Stop the RADIUS service on the replica server.
2. Log into the replica server as root.
3. Open a command window and navigate to the \Radius\Service directory (Windows) or /opt/PSsbr/radius directory (Linux).
4. Run the sbrsetuptool utility with the promote option.

   ```
   # sbrsetuptool -promote
   ```

   The utility creates a configuration package to change this server to the primary server.
5. Restart the updated replica server to make it the new primary server.
6. Publish a new configuration package administratively to configure all replica servers to use the new

primary server.

After you designate a new primary server for a realm, the old primary server becomes a replica server automatically.

Recovering a Replica After a Failed Download

If a replica server fails during the download of a configuration package, its configuration may be corrupted or it may have a stale secret.

To recover after a failed download:

1. Stop the RADIUS service on the replica server.
2. Log into the replica server as root.
3. Open a command window and navigate to the \Radius\Service directory (Windows) or /opt/PSsbr/radius directory (Linux).
4. Run the sbrsetuptool utility with the identity option and information on where to download configuration information.

   To obtain configuration from a configuration package, issue the following command:

   ```
   # sbrsetuptool -identity REPLICA -reppkg pathname
   ```

   where pathname specifies the path to a replica.ccmpkg package.

   To obtain configuration from the primary server for the realm, issue the following command:

   ```
   # sbrsetuptool -identity REPLICA -primary name address secret
   ```

   where name specifies the DNS name of the primary server, address specifies the IP address of the primary server, and secret specifies the shared secret used to authenticate configuration downloads.

5. Restart the updated replica server so that it can load its new configuration.

After the replica server is restarted, it will be re-synchronized with the current primary server.

Changing the Name or IP Address of a Server

You may need to change the DNS name or IP address assigned to a primary or replica server if your network changes.

To change the DNS name or IP address of a primary or replica server:

1. Stop the RADIUS service on the RADIUS server you want to change.
2. Log into the RADIUS server as root.
3. Open a command window and navigate to the \Radius\Service directory (Windows) or /opt/PSsbr/radius directory (Linux).
4. Run the sbrsetuptool (Linux) utility with the identity option. To

   rename a primary server, enter the following command:

   ```
   # sbrsetuptool -identity PRIMARY
   ```

   To rename a replica server, enter the following command:

```
# sbrsetuptool -identity REPLICA
```

5. Restart the updated server so that it can load its new configuration.
6. Run the SBR Administrator and modify the DNS name or IP address for the server you want to rename. Verify that the secret on the renamed server is correct.

   You may need to use the Replication page to delete the old server name from the list of servers in the realm.
7. Publish the modified configuration to propagate the name change to the replica servers.

Replication Error Messages

The following tables list possible causes for error messages caused by replication issues.

Error Messages on Replica Servers

Table 36 lists possible causes for error messages on replica servers in a replication realm.

Table 36: Error Messages on Replica Servers

Error Type	Error Message	Possible Cause
Post Errors (Errors with Notification from Primary)	CRadManagedServerNotifyPost:: ExecutePost invalid signature!	Mismatched replication secret.
Post Errors (Errors with Notification from Primary)	CRadManagedServerNotifyPost:: ExecutePost invalid sequence number	Two posts have the same sequence number. The clocks on the primary and replica are more then 10 minutes apart.
	CRadManagedServerNotifyPost:: ExecutePost decrypt failed	Shared secret failed to decrypt. Bad Replication Secret secret.
	CRadManagedServerNotifyPost:: ExecutePost invalid <body> missing parameters	Post had an invalid xml request.
Update Errors (Errors with Published package from Primary)	CRadManagedServerUpdate:: DoStart Failed to open 'file_name' for writing	Temp directory does not exist. Temp directory or file have incorrect permissions.
	CRadManagedServerUpdate:: StartUpdates has already started update	Update is already in progress.
	CRadManagedServerUpdate:: DownloadPackage HTTP POST error:errCode Primary ID	Error in transmitting request to Primary (timeout during transmit).
	CRadManagedServerUpdate:: DownloadPackage HTTP headers parsing error	Error in receiving package. Typically caused by a timeout during receive resulting from an invalid package.
	CRadManagedServerUpdate:: DownloadPackage connection primary IP Addr error: errCode Primary ID	Replica failed to connect with Primary. Primary not running.
	CRadManagedServerUpdate::	Update failed after three attempts.

Error Type	Error Message	Possible Cause
	DownloadPackage exceeded iterations limit while communicating with CCM	
	CRadManagedServerUp date:: ProcessPackage signature mismatch	Secrets on Replica and Primary do not match.
	CRadManagedServerUpdate:: ProcessPackage CCM error: ‘Error String’ ‘Pararmeter’	Error parsing downloaded packages.
	CRadManagedServerUpdate:: ProcessPackage Failed to open \”’ << file_name << “\’ for writing	Temp directory does not exist. Temp directory or file has incorrect permissions.
	CRadManagedServerUpdate:: ProcessPackage thumbprint mismatch	Invalid package. Republish the package.
Proxy Errors (Statistics retrieve errors)	CRadProxyPost:: ExecutePost invalid signature!	Mismatched replication secret.
	(Statistics retrieve errors)	Two posts have the same sequence number. The clocks on the primary and replica are more than 10 minutes apart.
		Shared secret failed to decrypt. Bad Replication Secret secret.
	CRadProxyPost:: ExecutePost invalid <body> missing parameters	Post had an invalid XML request.

Error Messages on Primary Servers

Table 37 lists possible causes for error messages on primary servers in a replication realm.

Table 37: Error Messages on Primary Servers

Error Type	Error Message	Possible Cause
Notify Target (Both Notify and Publish send a notification)	CRadConfigManagedServerHTTP Notification::NotifyTarget failed to fetch	Replica does not exist in database. This can occur if two administrators are running instances of SBR Administrator, one administrator deletes a replica, then the other administrator tries
	CRadConfigManagedServerHTTP Notification::NotifyTarget failed replicaId	Notify failed to communicate with replica, Replica is not running, or check Replica DCF log for more information.
Publication Provider (requests from GUI to Notify or Publish)	CRadConfigPublicationProvider:: UpdateResource notify invoked when not Primary	Attempted to Notify as a Replica, this is only allowed from a Primary.
	CRadConfigPublicationProvider:: UpdateResource publish invoked when not Primary	Replica attempted to publish; publication is permitted only from a Primary.

Error Type	Error Message	Possible Cause
Publication Post (parsing of Post from replica to get data)	CRadConfigServerProviderPost:: ExecutePost signature mismatch with server:	Replica and Primary have different replication secrets.
	CRadConfigServerProvider:: GetResource invoked when not Primary	Another Replica is requesting a download from this server which is a replica. Replica that is requesting a download needs to be reconfigured (see disaster recovery in docs).
Proxy Errors (Statistics retrieve errors)	CRadProxyClient:: Send failed to fetch	Replica does not exist in database. This can occur if two administrators are running instances of SBR Administrator, one administrator deletes a replica, then the other administrator tries to publish to that replica.
	CRadProxyClient:: SendData HTTP POST error:	Connection error with replica.

Chapter 30
LDAP Configuration Interface

This chapter describes:

- The file used to enable and configure the LDAP configuration interface (LCI)
- An overview of the LCI and LDAP utilities
- A description of the LDAP virtual schema
- Information about how to use LDAP utilities to configure the Steel-Belted Radius database
- Sample LDIF files that control the execution of LDAP utilities
- Information about how to view rate statistics variables with LDAP utilities

LDAP Configuration Interface File

The radius.ini file establishes settings for the LDAP configuration interface. For more information about radius.ini, refer to the *Steel-Belted Radius Reference Guide*.

Table 38: LDAP Configuration Interface Files

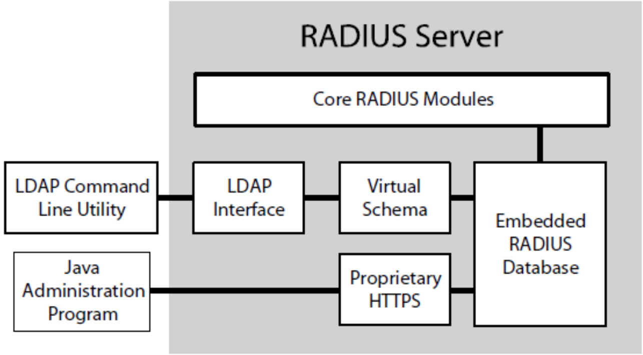
File Name	Function
radius.ini	Specifies whether the LCI is enabled, the port used for LCI communication, and the interfaces on which Steel-Belted Radius listens for LCI requests.

About the LDAP Configuration Interface

The LDAP Configuration Interface (LCI) provided by Steel-Belted Radius consists of an LDAP interface in the Steel-Belted Radius server and an LDAP virtual schema. The LDAP virtual schema presents the structure of the Steel-Belted Radius database in a manner that can be understood by the LDAP client utilities. The LCI uses the virtual schema to retrieve, modify, and delete entries in the database.

Figure 180: LDAP Components illustrates the relationship between LDAP components, the Administrator, and the configuration database.

Figure 180: LDAP Components



LDAP Utilities

Freeware LDAP utilities, such as ldapsearch, ldapdelete, and ldapmodify, act as clients of the LDAP interface. LDAP utilities let you read and modify an LDAP database.

- ldapsearch—The ldapsearch utility locates and retrieves LDAP directory entries. The ldapsearch utility opens a connection to an LDAP interface using the specified distinguished name and password, binds, and locates entries based on the specified search filter. A search can return a single entry, an entry's immediate subentries, or an entire tree or subtree. Search results are returned in LDAP Data Interchange Format (LDIF) format.
- ldapdelete—The ldapdelete utility deletes entries from an existing LDAP directory. ldapdelete opens a connection to the specified server using the distinguished name and password you provide, binds, and deletes the entry or entries.
- ldapmodify—The ldapmodify utility adds or modifies entries in an existing LDAP directory. ldapmodify opens a connection to an LDAP interface using the distinguished name and password you supply, binds, and adds or modifies the entries based on the LDIF update statements contained in a specified file. Because ldapmodify uses LDIF update statements, ldapmodify can do everything ldapdelete can do.

LDAP Requests

LDAP requests are submitted in two ways:

- By specifying options on the LDAP configuration interface command line.
- By placing instructions and data into an LDIF file, which you then process by invoking an LDAP command line utility using the -f option.

Because communication between the LDAP client and server is unencrypted, the LDAP utilities should be run on the same computer as Steel-Belted Radius.

Downloading the LDAP Utilities (Windows)

To use the LCI, you need the ldapsearch, ldapmodify, and ldapdelete utilities. You can download freeware Windows LDAP utilities as follows:

1. Use a browser to navigate to http://www.sun.com/download/products.xml?id=/3ec28dbd.
2. When the Sun ONE Directory SDK (software development kit) download page appears, click the Download link at the bottom of the page.
3. If you are prompted to register yourself, complete the registration form.
4. When you are prompted to accept the license agreement, click the Accept button and then click Continue.
5. Download the SDK by clicking the link for the version of the SDK that is appropriate for your computer. Versions of the SDK are available for Linux and Windows.
6. When the download is completed, extract the files from the compressed image to a directory on your computer.

To run the LDAP utilities, execute them from this directory. Note that, if you set the path environment variable to point to this directory, you can run them from any location on the system.

Note: The examples that follow assume you are using the LDAP utilities provided as part of the Sun ONE

Directory SDK. If you are using LDAP utilities from another source, the command options you use may be different. Consult the documentation for your LDAP utilities for more information.

LDAP Version Compliance

The LDAP interface in Steel-Belted Radius complies with version 2 of the LDAP specification. You should use the -V 2 command option to direct the utilities to use version 2 features. For example:

```
ldapmodify -c -V 2 -p 354 -D “cn=admin,o=radius” -w radius -f filename
```

Configuring the LDAP TCP Port

To avoid conflicts with LDAP services that may already be installed, the default port number for communication between Steel-Belted Radius and the LDAP client is 667. You can configure Steel-Belted Radius to use a different TCP port to communicate with an LDAP client. For example, you can change this port number to 389, the standard LDAP TCP port, if you are certain doing so will not create port number conflicts with other applications.

The following example configures Steel-Belted Radius to use TCP port 354.

1. In the radius.ini file, uncomment the [LDAP] section, set Enable to 1, and set the TCPPort field to the port number you want to use. For example:

   ```
   [LDAP]

   Enable = 1

   TCPPort = 354
   ```

2. If you want to specify the interfaces on which Steel-Belted Radius listens for LCI requests, add a [LDAPAddresses] section to the radius.ini file. This section should contain a list of IP addresses, one per line. For example:

   ```
   [LDAPAddresses]

   192.168.12.45

   10.10.10.25
   ```

   If the [LDAPAddresses] section is omitted or empty, Steel-Belted Radius listens for LCI requests on all bound IP interfaces.

You must specify the port number (by means of the -p option) when you run the LDAP utilities. For example:

```
ldapsearch -V 2 -p 354 -D “cn=admin,o=radius” -w radius -s sub -T -b “radiusclass=Client,o=radius”

radiusname=*
```

Example

The Steel-Belted Radius server at the Good Times Clock Company has two network interfaces. The first interface (192.168.10.40) connects to the corporate network. The second interface (192.168.20.50) connects to a dedicated administrative VLAN accessible only from the local subnet. To limit access to the LCI to network administrators, the [LDAPAddresses] section of radius.ini specifies that LCI requests must come through the administrative interface (192.168.20.50). LCI requests coming through the corporate network interface (192.168.10.40) are ignored.

Configuring the LCI Password

After you enable the LCI, you should change the default LCI password to prevent unauthorized LDAP clients from accessing your database. After you install the LDAP utilities and verify that they work, perform the following steps:

1. Create a text file called temp.ldif with the following contents:

   ```
   dn:   radiusclass=server,o=radius

   changetype: modify

   replace:   server-password

   server-password:   new-password
   ```

   where new-password is the LCI password you want to use.

2. Change the radius.ini [LDAP] setting to Enable=1.

3. Restart Steel-Belted Radius.

4. Execute the following command:

   ```
   ldapmodify -V 2 -h ip-address -p port -D “cn=admin,o=radius” -w oldpassword -f temp.ldif
   ```

   where:

   -h ip-address specifies the IP address of the Steel-Belted Radius server.

   -p port specifies the port number specified in the [LDAP] section of the radius.ini file.

   -w oldpassword specifies the current password (which is radius by default).

5. Verify that the password change was successful by executing the following command:

   ```
   ldapsearch -V 2 -h ip-address -p port -D “cn=admin,o=radius” -w newpassword -s sub -T -b “o=radius” radiusclass=server
   ```

   where:

   -h ip-address specifies the IP address of the Steel-Belted Radius server.

   -p port specifies the port number specified in the [LDAP] section of the radius.ini file.

   -w newpassword specifies the password configured in the temp.ldif file.

After you verify that the password change has been successful, delete the temp.ldif file and any other file that contains a cleartext copy of the modified LCI password.

Note: The LDAP Configuration Interface does not support Secure Sockets Layer (SSL).

LDAP Virtual Schema

The LDAP interface uses the virtual schema (illustrated in Figure 181: LDAP Schema (Slide 1 of 5) – Figure 185: LDAP Schema (Slide 5 of 5)) to represent the structure of the Steel- Belted Radius database. LDAP clients use the virtual schema to exchange configuration data over the LDAP configuration interface.

Note: Your edition of Steel-Belted Radius may not support all branches of the schema illustrated in

Figure 181: LDAP Schema (Slide 1 of 5) through Figure 185: LDAP Schema (Slide 5 of 5).

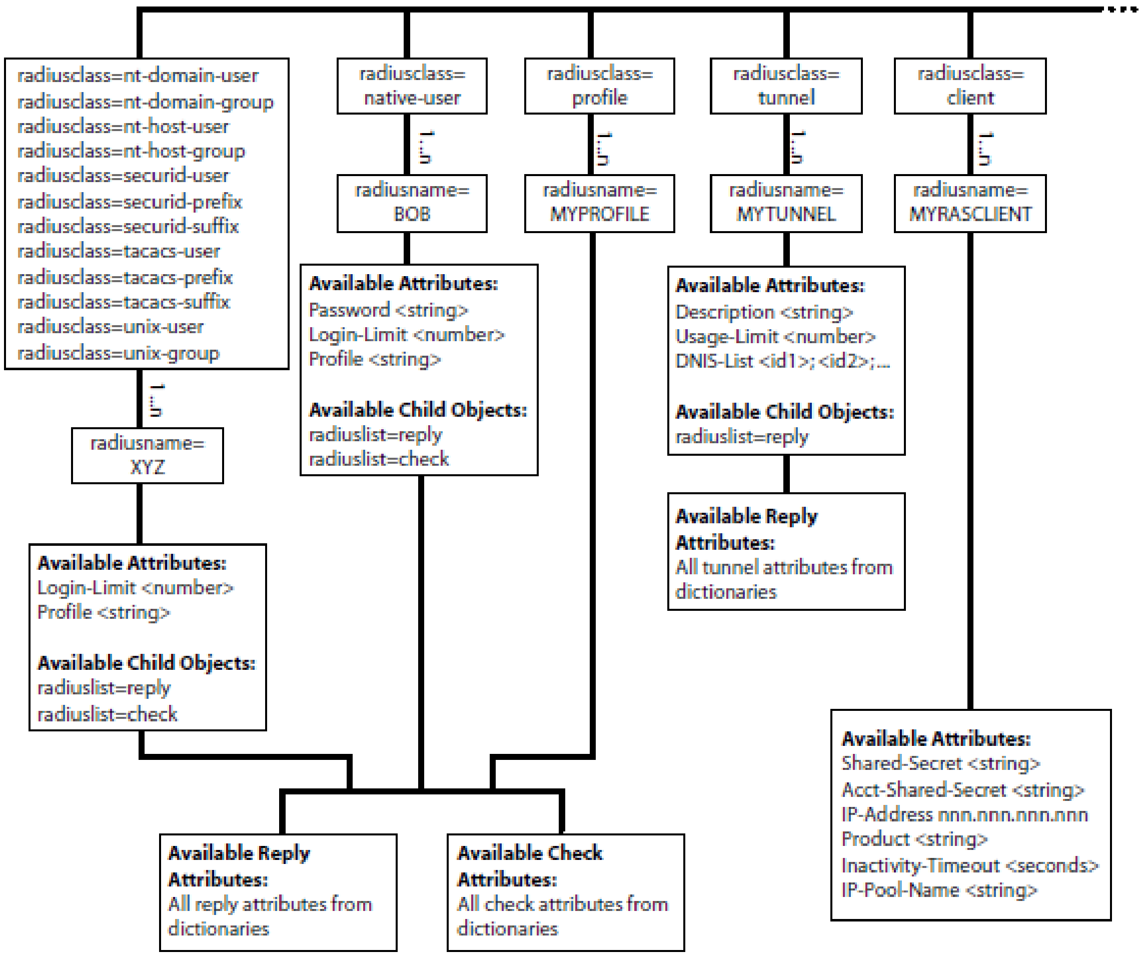
Many of the top-level items in the LDAP virtual schema correspond to windows and panels in the SBR Administrator.

Table 39: LDAP Schema and SBR Administrator Dialogs

Item	See
radiusclass=client	"**Administering RADIUS Clients via Legacy SBR Administrator**"
radiusclass=native-user, securid-user, ...	"**Administering Users via Legacy SBR**"
radiusclass=profile	"**Administering Profiles via Legacy SBR Administrator**"
radiusclass=proxy	"**Administering Proxy RADIUS via Legacy SBR**"
radiusclass=tunnel	"**Administering RADIUS Tunnels via Legacy SBR**"
radiusclass=server	"**Setting Up EAP Authentication Policies**"
radiusclass=ip-addr-pool	"**Setting Up IP Address Pools**"
radiusclass=ipx-addr-pool	"**Setting Up IPX Address Pools**"
radiusstatus=statistics	"**Displaying Statistics**"
radiusstatus=sessions	"**Displaying the Current Sessions List**" Note: LDAP searches that call radiusstatus=sessions can adversely affect Steel Belted Radius performance. When possible, you should search using the sessions_ by keywords.

Note: radiusstatus items can be read, but they cannot be modified.

Figure 181: LDAP Schema (Slide 1 of 5)



radiusclass=nt-domain-user
radiusclass=nt-domain-group
radiusclass=nt-host-user
radiusclass=nt-host-group
radiusclass=securid-user
radiusclass=securid-prefix
radiusclass=securid-suffix
radiusclass=tacacs-user
radiusclass=tacacs-prefix
radiusclass=tacacs-suffix
radiusclass=unix-user
radiusclass=unix-group

radiusname=
XYZ

Available Attributes:
Login-Limit <number>
Profile <string>

Available Child Objects:
radiuslist=reply
radiuslist=check

radiusclass=
native-user

radiusname=
BOB

Available Attributes:
Password <string>
Login-Limit <number>
Profile <string>

Available Child Objects:
radiuslist=reply
radiuslist=check

radiusclass=
profile

radiusname=
MYPROFILE

radiusclass=
tunnel

radiusname=
MYTUNNEL

Available Attributes:
Description <string>
Usage-Limit <number>
DNIS-List <id1>;<id2>;...

Available Child Objects:
radiuslist=reply

Available Reply Attributes:
All tunnel attributes from dictionaries

radiusclass=
client

radiusname=
MYRASCLIENT

Available Attributes:
Shared-Secret <string>
Acct-Shared-Secret <string>
IP-Address nnn.nnn.nnn.nnn
Product <string>
Inactivity-Timeout <seconds>
IP-Pool-Name <string>

Available Reply Attributes:
All reply attributes from dictionaries

Available Check Attributes:
All check attributes from dictionaries

Figure 182: LDAP Schema (Slide 2 of 5)

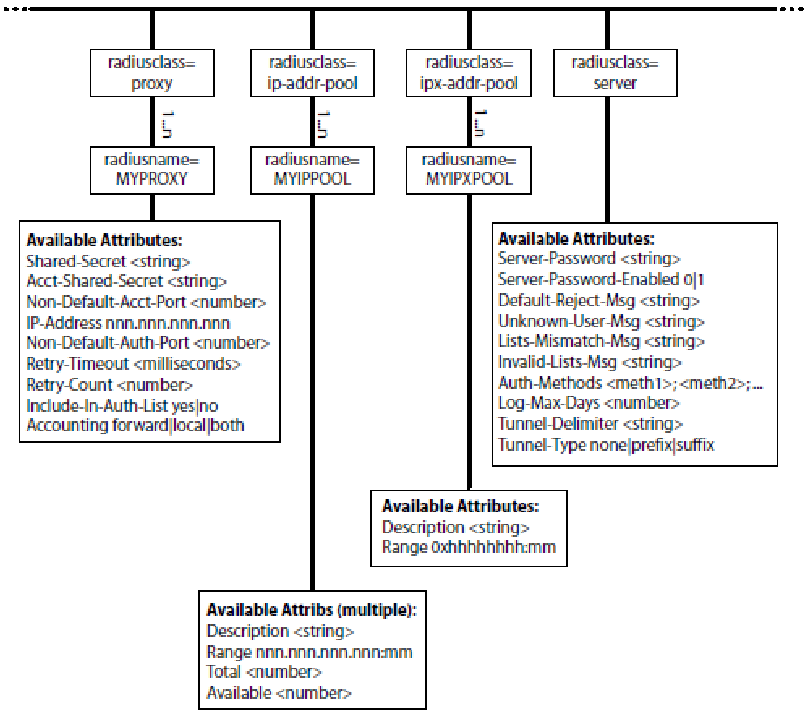

radiusclass=
proxy
radiusclass=
ip-addr-pool
radiusclass=
ipx-addr-pool
radiusclass=
server
1..5
1..5
1..5
radiusname=
MYPROXY
radiusname=
MYIPPOOL
radiusname=
MYIPXPOOL
Available Attributes:
Shared-Secret <string>
Acct-Shared-Secret <string>
Non-Default-Acct-Port <number>
IP-Address nnn.nnn.nnn.nnn
Non-Default-Auth-Port <number>
Retry-Timeout <milliseconds>
Retry-Count <number>
Include-In-Auth-List yes|no
Accounting forward|local|both
Available Attributes:
Server-Password <string>
Server-Password-Enabled 0|1
Default-Reject-Msg <string>
Unknown-User-Msg <string>
Lists-Mismatch-Msg <string>
Invalid-Lists-Msg <string>
Auth-Methods <meth1>;<meth2>;...
Log-Max-Days <number>
Tunnel-Delimiter <string>
Tunnel-Type none|prefix|suffix
Available Attributes:
Description <string>
Range 0xhhhhhhhh:mm
Available Attribs (multiple):
Description <string>
Range nnn.nnn.nnn.nnn:mm
Total <number>
Available <number>

Figure 183: LDAP Schema (Slide 3 of 5)

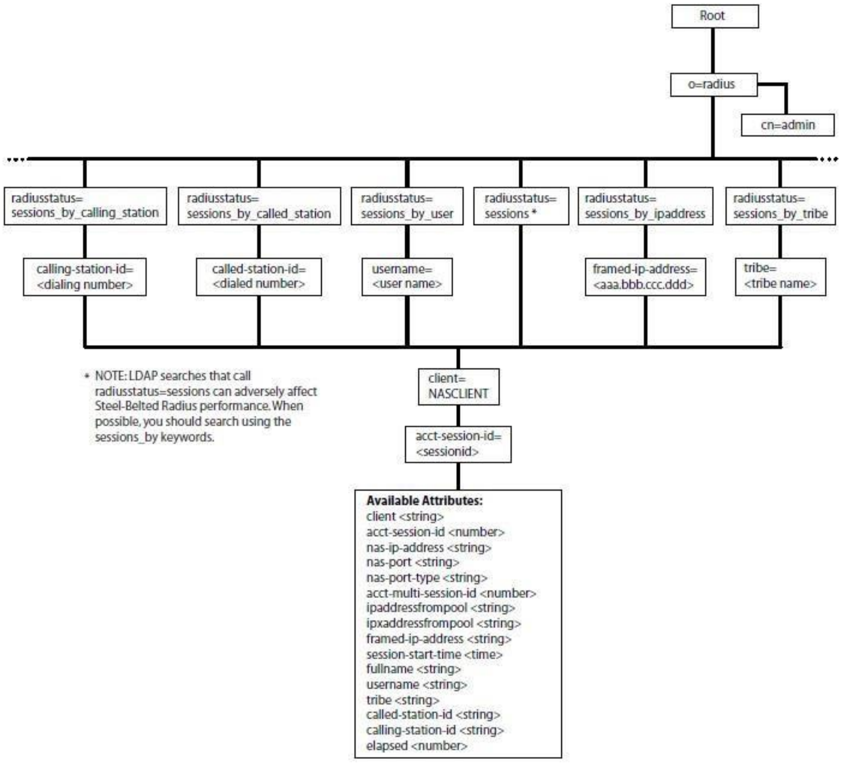
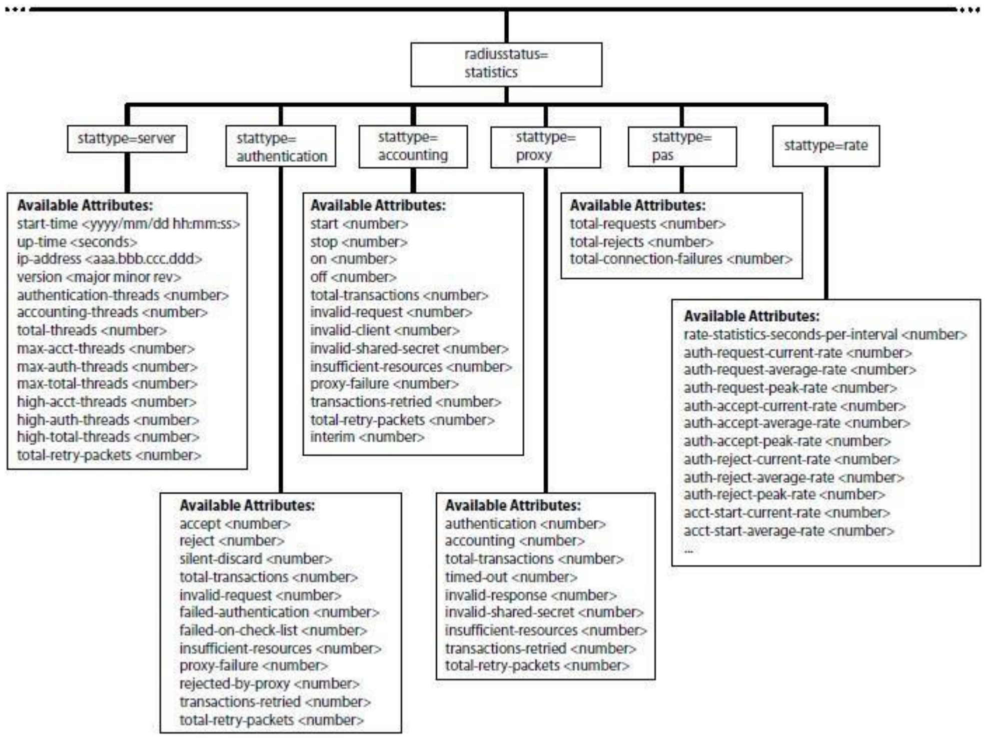
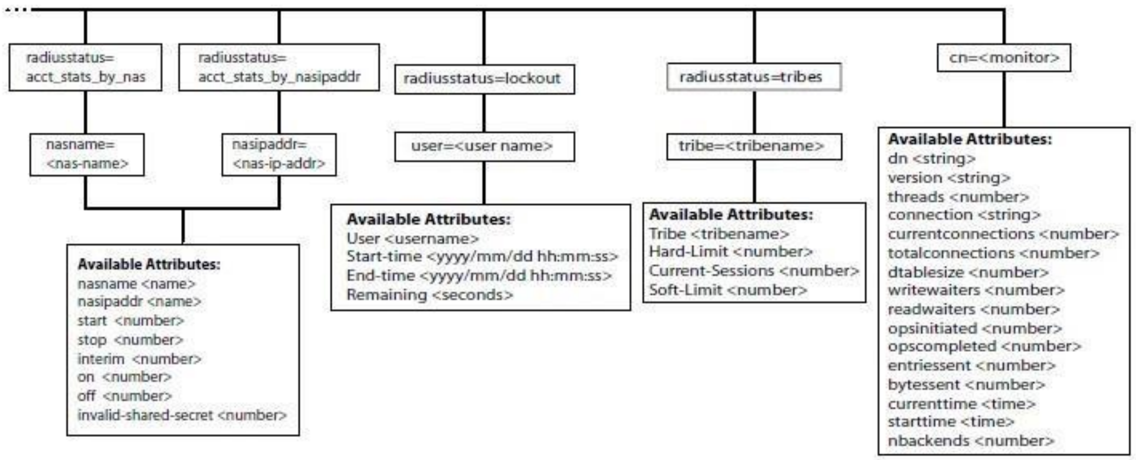


Figure 184: LDAP Schema (Slide 4 of 5)


radiusstatus=
statistics
stattype=server
stattype=
authentication
stattype=
accounting
stattype=
proxy
stattype=
pas
stattype=rate
Available Attributes:
start-time <yyyy/mm/dd hh:mm:ss>
up-time <seconds>
ip-address <aaa.bbb.ccc.ddd>
version <major minor rev>
authentication-threads <number>
accounting-threads <number>
total-threads <number>
max-acct-threads <number>
max-auth-threads <number>
max-total-threads <number>
high-acct-threads <number>
high-auth-threads <number>
high-total-threads <number>
total-retry-packets <number>
Available Attributes:
start <number>
stop <number>
on <number>
off <number>
total-transactions <number>
invalid-request <number>
invalid-client <number>
invalid-shared-secret <number>
insufficient-resources <number>
proxy-failure <number>
transactions-retried <number>
total-retry-packets <number>
interim <number>
Available Attributes:
total-requests <number>
total-rejects <number>
total-connection-failures <number>
Available Attributes:
rate-statistics-seconds-per-interval <number>
auth-request-current-rate <number>
auth-request-average-rate <number>
auth-request-peak-rate <number>
auth-accept-current-rate <number>
auth-accept-average-rate <number>
auth-accept-peak-rate <number>
auth-reject-current-rate <number>
auth-reject-average-rate <number>
auth-reject-peak-rate <number>
acct-start-current-rate <number>
acct-start-average-rate <number>
...
Available Attributes:
accept <number>
reject <number>
silent-discard <number>
total-transactions <number>
invalid-request <number>
failed-authentication <number>
failed-on-check-list <number>
insufficient-resources <number>
proxy-failure <number>
rejected-by-proxy <number>
transactions-retried <number>
total-retry-packets <number>
Available Attributes:
authentication <number>
accounting <number>
total-transactions <number>
timed-out <number>
invalid-response <number>
invalid-shared-secret <number>
insufficient-resources <number>
transactions-retried <number>
total-retry-packets <number>

Note: The Enterprise Edition of Steel-Belted Radius with the optional LCI add-on does not support the Statistics items.

Figure 185: LDAP Schema (Slide 5 of 5)



LDAP Rules and Limitations

While the LDAP virtual schema diagram shows as much of the detail of the LDAP virtual schema as possible, the following rules and limitations should be considered.

- **Bind request—**All attempts to perform operations on the virtual schema must be preceded by an LDAP Bind request that authenticates the administrator to the Steel-Belted Radius server. The Bind request must reference a Steel-Belted Radius administrative account and must provide the password that authenticates that account. This translates into the following command line options for each invocation of the LDAP utilities:

```
-D"cn=AdminName,o=radius" -w AdminPassword
```

where AdminName is the administrative account name and AdminPassword is its password.

- **Uppercase and lowercase—**The uppercase/lowercase rules for object names are the same as in the SBR Administrator; that is, almost all object names are stored in the database in uppercase format. The exception to this rule is that UNIX User/Group, SecurID User/Prefix/Suffix and TACACS+ User/Prefix/Suffix names are maintained in the case specified in the LDIF files.
- **Attributes—**When you enter attributes, make sure that the attribute name matches the name found in the dictionary and that the attribute's value is consistent with the syntax type for the attribute. Note that the LDAP virtual schema does not list all the dictionary attributes that are available in Steel-Belted Radius.
- **IP addresses—T**he ipaddr-pool type in the dictionary can represent an IP address or a pool name. If the value specified begins with the marker string [pool], the token that follows the marker string is assumed to be an IP pool name; otherwise, it must be a valid IP address. If it is neither, the operation fails.

  Address ranges in IP address pool objects are specified in the form IPaddress:NumberOfAddresses. An example of a valid range is 128.22.12.45:34.
- **IPX addresses—**The ipxaddr-pool syntax type in the dictionary lets you enter an IPX network address (up to 8 hexadecimal digits using the format 0xhhhhhhhh) or choose a pool name. If the value specified begins with the string [pool], the token that follows the marker is assumed to be an IPX pool name; otherwise, it must be a valid IPX address. If it is neither, the operation fails.

  Address ranges in IPX address pool objects are specified in the form IPXNetAddress:NumberOfAddresses. An example of a valid range is 0xa020443b:34.
- **Substrings—**Some attribute may have a value that consists of a list of strings. For example, the DNIS list in a tunnel entry and the authentication method list may consist of multiple substrings. The rule for specifying the data portion for these lists is that semicolons must delimit substrings. For example, a DNIS list for a tunnel entry might be specified as 555-1212;5551212. If a semicolon needs to appear inside a substring, it must be escaped by placing a backslash character (\) before it.
- **Hexadecimal values—**Hexadecimal numbers (for attributes of syntax type hex1, hex2, or hex4) require a 0x prefix before the hexadecimal digits; for example 0x0000149a.
- Password syntax—Passwords that are retrieved from the database may consist of one of the following:
  - A clear-text password of the form {x-clear} clear-text-password-string if the password is weakly encrypted in the database.
  - A string of the form {x-md5}xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx if the password is stored as a one-way md5 hash.
  - A string of the form {x-md5}[encrypt]clear-text-password-string indicates that, although the password is specified in clear-text form, it is to be stored as a hash.
  - SBR 6.2 also supports SHA, SSHA, SHA-256, SSHA-256, SHA-512, SSHA-512 password formats also. Password prefix:

SHA-1
{SHA}XXXXXXXXXXXXXXXX
{SSHA}XXXXXXXXXXXXXXX

(Following feature exclusively added in 6.2)
SHA256
{SHA256}XXXXXXXXXXXXXXXX
{SSHA256}XXXXXXXXXXXXXXX

SHA512
{SHA512}XXXXXXXXXXXXX
{SSHA512}XXXXXXXXXXXXXX

White space in a password is treated as follows:

- When clear-text passwords are specified, the password is assumed to begin immediately after the right brace or right bracket. Adding a white space character, such as a space or tab, after the right brace or right bracket causes the white space to be considered part of the password.
- White space entered at the beginning of the attribute (before the left brace or left bracket) is ignored.
- White space entered between the right brace of {x-md5} and the left bracket of [encrypt] is ignored.
- All white space specified in the hexadecimal sequence describing a password hash is ignored.

- **Profiles, checklists, and return lists—**Steel-Belted Radius supports user definitions that include attribute subtractions of profile entries. To specify that a user attribute is to be considered a subtraction of a profile attribute, preface the attribute value with the string %subtract%.

  Steel-Belted Radius permits user and profile checklists to include default values for attributes. Configuring a default value for an attribute means that, if a RADIUS request does not include this attribute, the request should not be rejected. Instead, the value supplied as the default should be used as if it were received as part of the request. To specify that a checklist attribute is to be considered a default attribute, preface the attribute value with the string %default%.

  Steel-Belted Radius permits user and profile return lists to include attributes whose values are set by copying the contents of received attributes. This feature is referred to as “attribute echoing.” To specify that a return list attribute is to be treated as an echo attribute, enter %echo% for the attribute value.

- **Unspecified or 0.0.0.0 RAS IP address—**When you display acct_stats_by_nasipaddr information, any RAS entries with an unspecified IP address or an IP address of 0.0.0.0 are omitted. Similarly, when you display acct_stats_by_nas information, any RAS entries with an unspecified IP address or an IP address of 0.0.0.0 will have their nasipaddr attribute omitted.

- **Duplicate RAS IP addresses—**When displaying acct_stats_by_nasipaddr information, two RAS entries that contain the same (non-zero) IP address cause information about one of the entries to be displayed twice. This is the result of the ambiguity of the query and is not a bug.

- **RADIUS client information displayed after deletion—**If you define a RADIUS client entry, send some accounting traffic to it, and then delete the entry, the output of ldapsearch queries will continue to list the deleted RADIUS client so that the per-RAS statistics add up to the total RAS statistics.

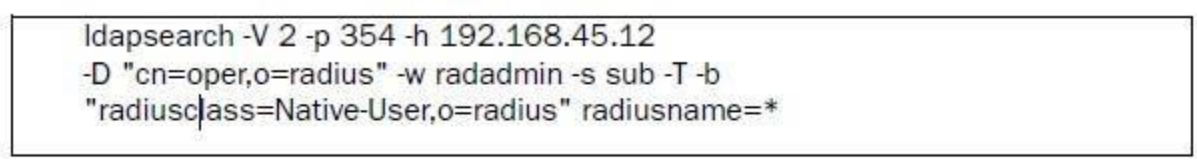
LDAP Command Examples

This section explains how to use the ldapdelete, ldapmodify, and ldapsearch utilities to configure the server.

Searching for Records

You can use the ldapsearch command to extract information from the LDAP tree. The command shown in Figure 186: ldapsearch Command lets you extract information about all RADIUS Native Users.

Figure 186: ldapsearch Command



```
ldapsearch -V 2 -p 354 -h 192.168.45.12
-D "cn=oper,o=radius" -w radadmin -s sub -T -b
"radiusclass=Native-User,o=radius" radiusname=*
```

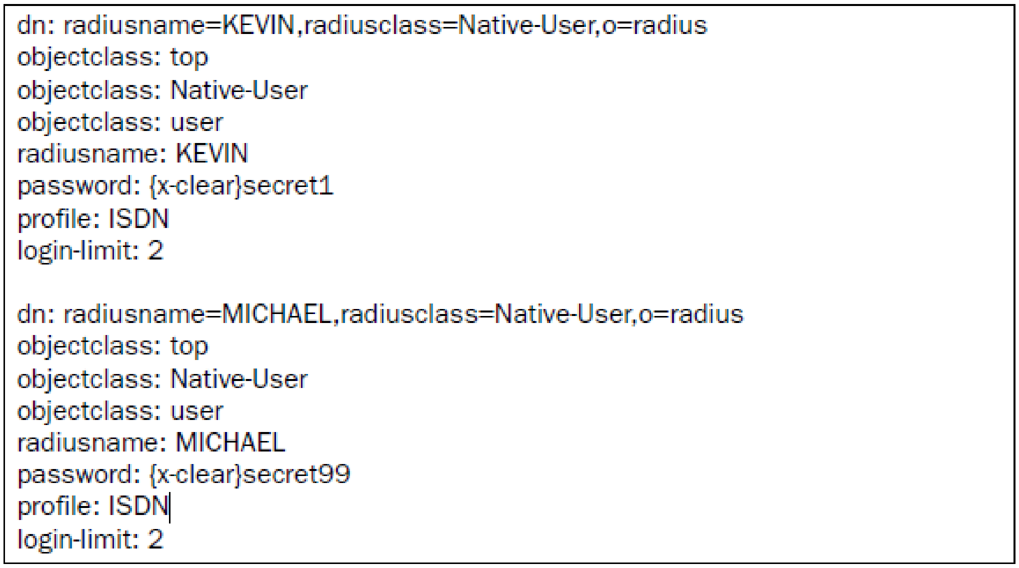
Note there must be a blank space between each option (for example, -p) and its value (for example, 354). Command syntax is case-sensitive.

Table 40: Searching for Records Using the ldapsearch Command

ldapsearch Option	Meaning
-V 2	Use LDAP Version 2 to communicate with the server. This option is not required, but it improves the performance of the transaction.
-p 354	Use TCP port 354 to communicate with the LDAP interface of the server. The -p value must match the TCPPort setting in the [LDAP] section of radius.ini. If the -p option is not specified, the LDAP utilities contact Steel-Belted Radius on the default port number (TCP port 389).
-h 192.168.45.12	Contact a remote host at the specified address or name. By default, ldapsearch tries to connect to the local host.
-D "cn=oper,o=radius"	Use the oper administrative account to authenticate the command. Note: You can use any administrative account name in place of oper in this example. Do not change the o=radius argument.
-w radadmin	Use an authentication password of radadmin. Note: The -w parameter value (in this case, radadmin) must match the password of the account named by the -D parameter.
-s sub	Perform a recursive subtree search from the base.
-T	Do not wrap long output lines to the next line.
-b "radiusclass=Client,o=radius"	Specifies the base from which the search operation starts.
radiusname=*	Specifies the selection criteria for the search.

Executing the ldapsearch command shown in Figure 186: ldapsearch Command against a Steel-Belted Radius server containing two Native User definitions would produce an LDIF file similar to the output shown in Figure 187: Search Results.

Figure 187: Search Results



```
dn: radiusname=KEVIN,radiusclass=Native-User,o=radius
objectclass: top
objectclass: Native-User
objectclass: user
radiusname: KEVIN
password: {x-clear}secret1
profile: ISDN
login-limit: 2

dn: radiusname=MICHAEL,radiusclass=Native-User,o=radius
objectclass: top
objectclass: Native-User
objectclass: user
radiusname: MICHAEL
password: {x-clear}secret99
profile: ISDN
login-limit: 2
```

Modifying Records

You can use the ldapmodify utility to update the Steel-Belted Radius configuration.

```
ldapmodify -c -V 2 -h example.host.com -p 354

-D “cn=oper,o=radius” -w radadmin -f filename
```

Note: There must be a blank space between each option (for example, -p) and its value (for example, 354). Command syntax is case-sensitive.

Table 41: Modifying Records Using the ldapmodify Command

ldapmodify Option	Meaning
-c	Run the command in continuous mode; do not stop on errors.
-V 2	Use LDAP Version 2 to communicate with the server. This option is not required, but it improves the performance of the transaction.
-h example.host.com	Contact a remote host at the specified address or name. If the -h option is not used, ldapsearch connects to the local database.
-p 354	Use TCP port 354 to communicate with the LDAP interface of the server. The -p value must match the TCPPort setting in the [LDAP] section of radius.ini. If the -p option is not specified, the LDAP utilities contact Steel-Belted Radius on the default port number (TCP port 389).
-D “cn=oper,o=radius”	Use the oper administrative account to authenticate the command. Note: You can use any administrative account name in place of oper in this example. Do not change the o=radius argument.

ldapmodify Option	Meaning
-w radadmin	Use an authentication password of radadmin. Note: The -w parameter value (in this case, radadmin) must match the password of the account named by the -D parameter.
-f filename	Specifies the input LDIF file to process.

The LDIF files generated by ldapsearch differ from those required for input to ldapmodify. The ldapmodify input files must contain a changetype entry immediately after each dn entry. The changetype entry specifies how to use the data to change the LDAP database.

The full syntax for changetype within each transaction is as follows:

```
dn:   distinguished-name-of-entry
changetype: keyword
subkeyword:attribute
attribute: value
changetype: keyword
subkeyword:attribute
attribute: value
:
```

where:

- keyword can be add, modify, or delete.
- subkeyword can be (respectively): add, replace, or delete.
- attribute can be any LDAP attribute in the entry.
- value is the value to assign to the attribute.

Repeated changetype: keyword entries are not required within a transaction unless you change the keyword. From top to bottom within the transaction, the latest keyword applies until another changetype: keyword entry is provided. The following syntax is valid if the same keyword applies throughout the transaction:

```
dn:   distinguished-name-of-entry
changetype: keyword
subkeyword:attribute
attribute: value
subkeyword:attribute
attribute: value
subkeyword:attribute
```

attribute: value

:

subkeyword: attribute entries are optional and indicate that you want to apply the change to a specific attribute within the entry. If no subkeyword: attribute entries in the transaction are found, the change applies to the entire entry. For example, it is faster to delete an entire entry:

```
dn:     radiusname=TINYCO.COM,radiusclass=Proxy,o=radius
changetype: delete
```

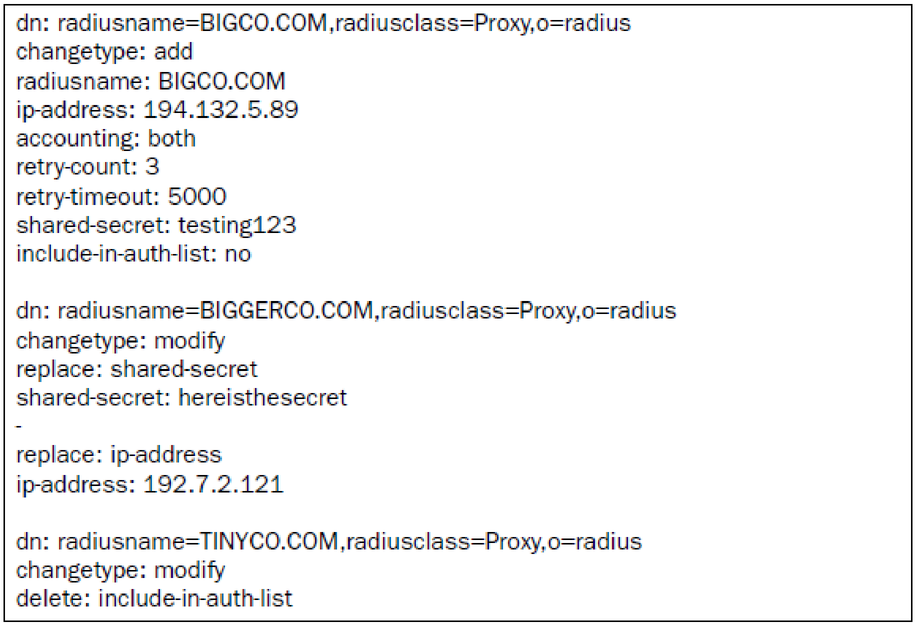
but if you want to delete only a few attributes from the entry, you can do so:

```
dn:     radiusname=TINYCO.COM,radiusclass=Proxy,o=radius
changetype: delete
delete: retry-count
-
delete:  include-in-auth-list
```

If the subkeyword is add or replace, an attribute: value entry must appear immediately following the subkeyword: attribute entry. If the subkeyword is delete, the attribute: value entry does not apply and should be omitted.

The following sample LDIF file could be used with an ldapmodify command.

Figure 188: Sample LDIF File



```
dn: radiusname=BIGCO.COM,radiusclass=Proxy,o=radius
changetype: add
radiusname: BIGCO.COM
ip-address: 194.132.5.89
accounting: both
retry-count: 3
retry-timeout: 5000
shared-secret: testing123
include-in-auth-list: no

dn: radiusname=BIGGERCO.COM,radiusclass=Proxy,o=radius
changetype: modify
replace: shared-secret
shared-secret: hereisthesecret
-
replace: ip-address
ip-address: 192.7.2.121

dn: radiusname=TINYCO.COM,radiusclass=Proxy,o=radius
changetype: modify
delete: include-in-auth-list
```

Note: To delete the proxy entry for TINYCO.COM, issue the following command:
dn: radiusname=TINYCO.COM,radiusclass=Proxy,o=radius
changetype: delete

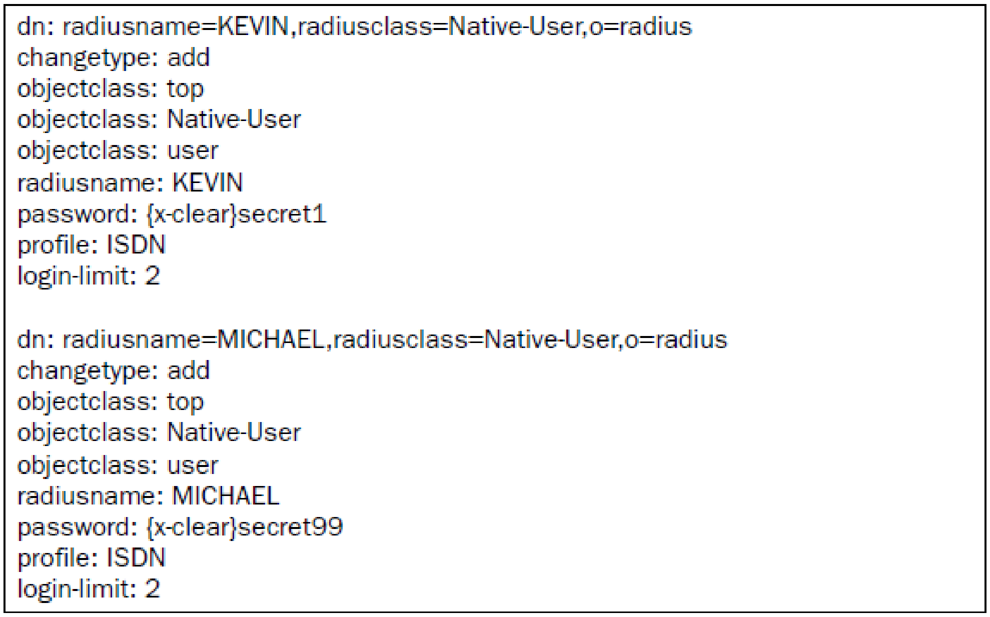
Importing Records From Another LDAP Database

To import entries from one LDAP database into another, run the ldapsearch command on the first database.

Request only the attributes you want for the new database. When ldapsearch completes processing, edit the output LDIF file. After each line that begins with dn:, add a single line containing the text changetype: add. Once your editing is complete, run an ldapmodify -f command that references the new LDIF file. After the ldapmodify command is executed, your new database is populated with the records you extracted from the old database.

The LDIF file shown in Figure 189: Adding Records with an LDIF File is derived from the output of the ldapsearch command. When specified as the input to an ldapmodify -f command, the contents of the file are added to the target database.

Figure 189: Adding Records with an LDIF File

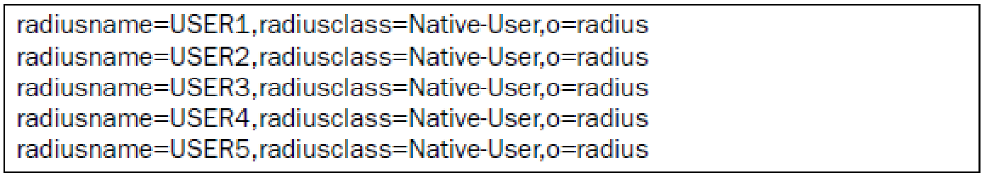


```
dn: radiusname=KEVIN,radiusclass=Native-User,o=radius
changetype: add
objectclass: top
objectclass: Native-User
objectclass: user
radiusname: KEVIN
password: {x-clear}secret1
profile: ISDN
login-limit: 2

dn: radiusname=MICHAEL,radiusclass=Native-User,o=radius
changetype: add
objectclass: top
objectclass: Native-User
objectclass: user
radiusname: MICHAEL
password: {x-clear}secret99
profile: ISDN
login-limit: 2
```

Deleting Records

The ldapdelete command allows you to remove records from the LDAP database. For example, to delete entries names USER1 through USER5, add the information shown in Figure 190: Deleting Records with an LDIF File to a file called deletedemo.ldf.

Figure 190: Deleting Records with an LDIF File



```
radiusname=USER1,radiusclass=Native-User,o=radius
radiusname=USER2,radiusclass=Native-User,o=radius
radiusname=USER3,radiusclass=Native-User,o=radius
radiusname=USER4,radiusclass=Native-User,o=radius
radiusname=USER5,radiusclass=Native-User,o=radius
```

Now, pass the deletedemo.ldf file to the ldapdelete command.

```
ldapdelete -V2 -h hostname -p 667

-D "cn=admin,o=radius" -w password -f deletedemo.ldf
```

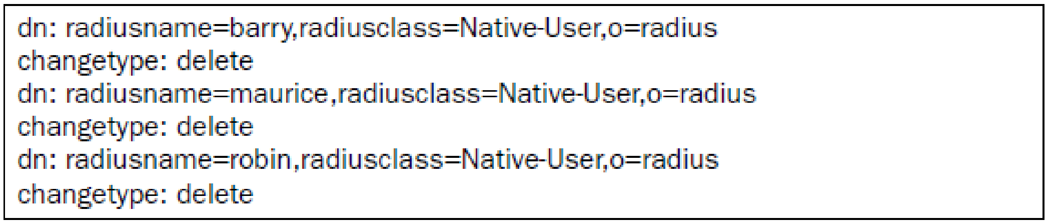
Note: Verify that the dn: values that usually appear in these entries are not a part of the entries in your file, because this will cause the command to fail.

You can use ldapdelete to remove records from the LDAP database without having to supply a file. For example, to delete the native user record identified as USER1, you would enter the following:

```
ldapdelete -V2 -h hostname -p 667
        -D "cn=admin,o=radius" -w password
        "radiusname=USER1,radiusclass=native-user,o=radius"
```

You can cause records to be deleted by means of the ldapmodify command, if the entries in the text file contain the line changetype: delete. Consider the sample LDIF file named deletemodify.ldf shown in Figure 191: deletemodify.ldf Example.

Figure 191: deletemodify.ldf Example



```
dn: radiusname=barry,radiusclass=Native-User,o=radius
changetype: delete
dn: radiusname=maurice,radiusclass=Native-User,o=radius
changetype: delete
dn: radiusname=robin,radiusclass=Native-User,o=radius
changetype: delete
```

The deletemodify.ldf file can be passed to the ldapmodify command as follows:

```
        ldapmodify -V2 -h hostname -p 667 -D"cn=admi,o=radius"
        -w password -f deletemodify.ldf
```

Note: On some LDAP servers, an error could cause the deletion of a container without prompting for confirmation. This could, in turn, cause the entire directory server to fail.

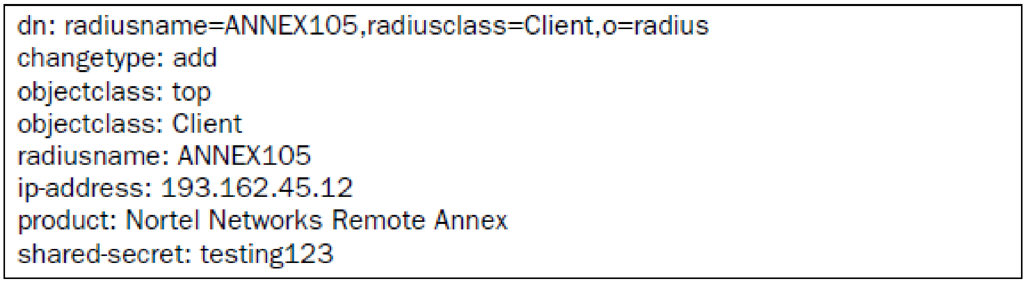
LDIF File Examples

This section explains how to construct LDIF files that, when input to the ldapmodify command, add entries to the Steel-Belted Radius database.

Adding RADIUS Clients with LDIF

The sample LDIF entry shown in Figure 192: Adding RADIUS Clientsadds a RADIUS client named ANNEX105 to the Steel-Belted Radius database.

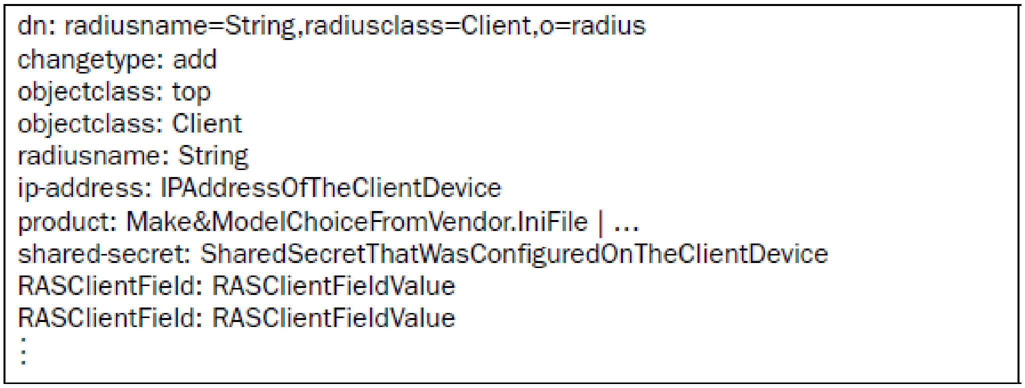
Figure 192: Adding RADIUS Clients



```
dn: radiusname=ANNEX105,radiusclass=Client,o=radius
changetype: add
objectclass: top
objectclass: Client
radiusname: ANNEX105
ip-address: 193.162.45.12
product: Nortel Networks Remote Annex
shared-secret: testing123
```

The syntax in this LDIF entry is shown in Figure 193: LDIF Syntax

Figure 193: LDIF Syntax

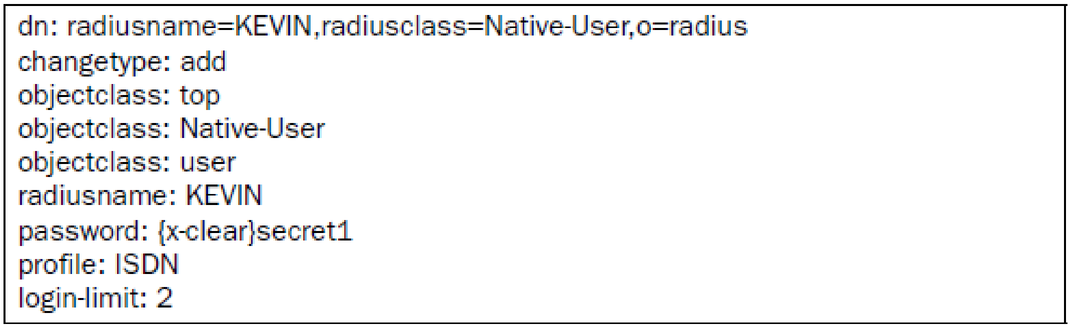


```
dn: radiusname=String,radiusclass=Client,o=radius
changetype: add
objectclass: top
objectclass: Client
radiusname: String
ip-address: IPAddressOfTheClientDevice
product: Make&ModelChoiceFromVendor.IniFile | ...
shared-secret: SharedSecretThatWasConfiguredOnTheClientDevice
RASClientField: RASClientFieldValue
RASClientField: RASClientFieldValue
⋮
```

Adding Users with LDIF

The sample LDIF entry shown in Figure 194: Adding Users adds a Local (Native) User named KEVIN to the Steel-Belted Radius   database.

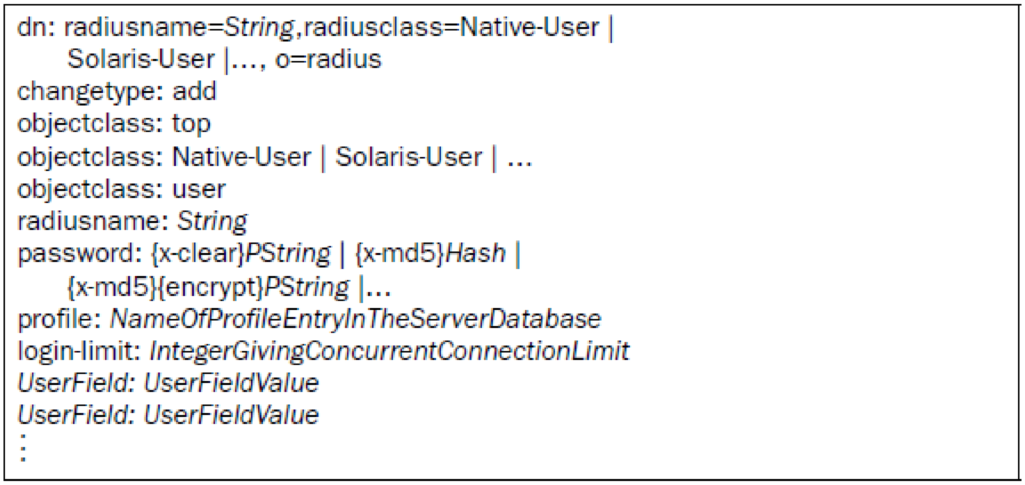
Figure 194: Adding Users



```
dn: radiusname=KEVIN,radiusclass=Native-User,o=radius
changetype: add
objectclass: top
objectclass: Native-User
objectclass: user
radiusname: KEVIN
password: {x-clear}secret1
profile: ISDN
login-limit: 2
```

The syntax in this LDIF entry is shown in Figure 195: LDIF Syntax.

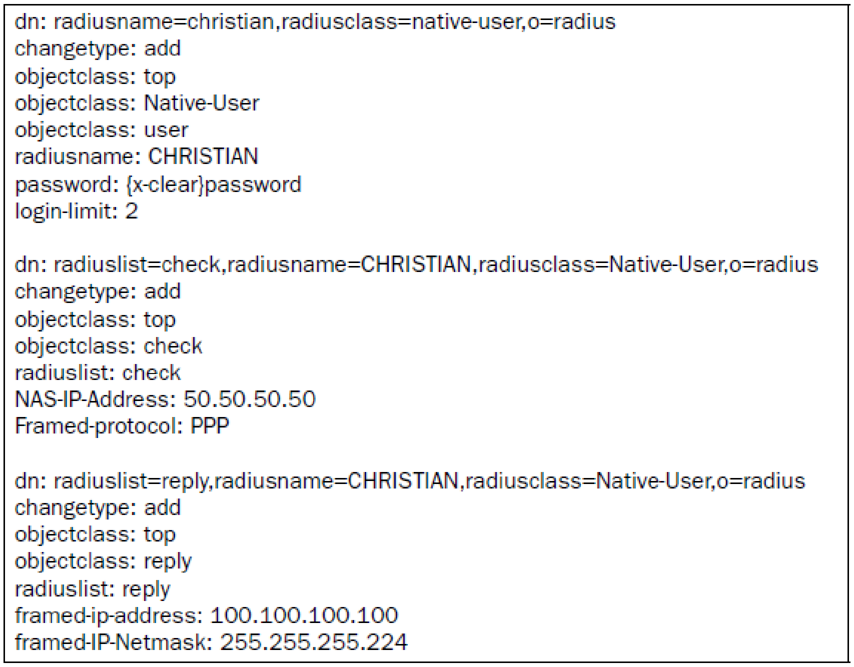
Figure 195: LDIF Syntax



```
dn: radiusname=String,radiusclass=Native-User |
     Solaris-User |..., o=radius
changetype: add
objectclass: top
objectclass: Native-User | Solaris-User | ...
objectclass: user
radiusname: String
password: {x-clear}PString | {x-md5}Hash |
     {x-md5}{encrypt}PString |...
profile: NameOfProfileEntryInTheServerDatabase
login-limit: IntegerGivingConcurrentConnectionLimit
UserField: UserFieldValue
UserField: UserFieldValue
⋮
```

The LDIF file shown in Figure 196: Adding a Native User add a local (native) user named CHRISTIAN, who has various attribute/value pairs assigned to his checklist and return list.

Figure 196: Adding a Native User



```
dn: radiusname=christian,radiusclass=native-user,o=radius
changetype: add
objectclass: top
objectclass: Native-User
objectclass: user
radiusname: CHRISTIAN
password: {x-clear}password
login-limit: 2

dn: radiuslist=check,radiusname=CHRISTIAN,radiusclass=Native-User,o=radius
changetype: add
objectclass: top
objectclass: check
radiuslist: check
NAS-IP-Address: 50.50.50.50
Framed-protocol: PPP

dn: radiuslist=reply,radiusname=CHRISTIAN,radiusclass=Native-User,o=radius
changetype: add
objectclass: top
objectclass: reply
radiuslist: reply
framed-ip-address: 100.100.100.100
framed-IP-Netmask: 255.255.255.224
```

Checklists and return lists are objects in the LDAP virtual schema, but the individual RADIUS attributes are not. Therefore, you must use a separate LDIF entry for each checklist and return list object, but each LDIF entry can name multiple attribute/value pairs.

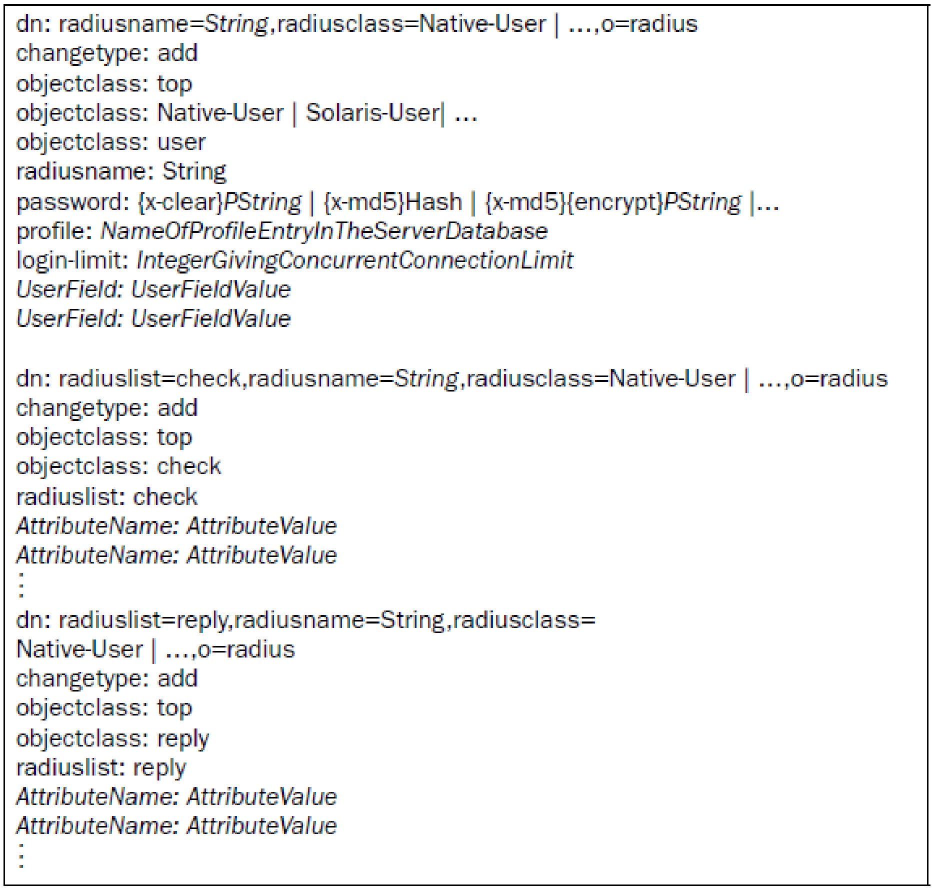
To indicate that a transaction applies to the user’s checklist (rather than to the user entry itself), use the keyword check as the value for radiuslist and objectclass within the transaction. You must assign this value to radiuslist in the distinguished name, and again before the list of attributes. You must also assign the value to objectclass, above the second radiuslist entry.

To indicate the return list, use the keyword reply.

The LDIF syntax to add a user entry, complete with a checklist and return list, is shown in Figure 197: Adding a User with Checklist and Return List Attributes.

Note: that the radiusname and radiusclass values for all of the transactions that apply to the same User entry must be the same.

Figure 197: Adding a User with Checklist and Return List Attributes

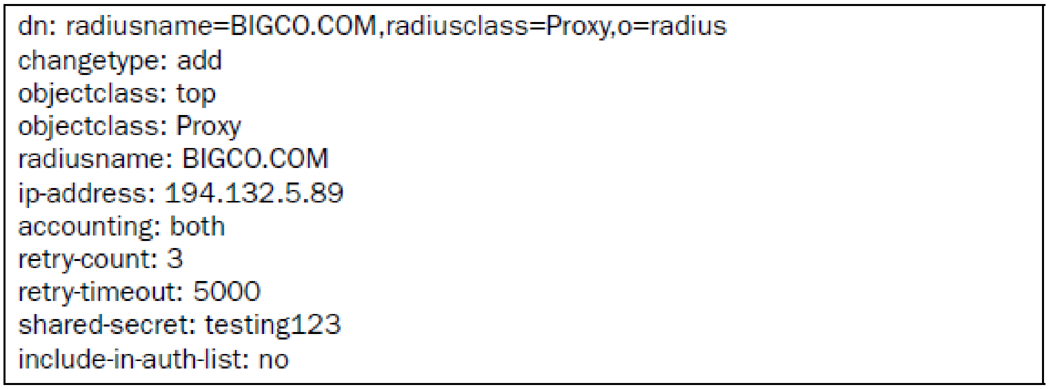


```
dn: radiusname=String,radiusclass=Native-User | …,o=radius
changetype: add
objectclass: top
objectclass: Native-User | Solaris-User| …
objectclass: user
radiusname: String
password: {x-clear}PString | {x-md5}Hash | {x-md5}{encrypt}PString |…
profile: NameOfProfileEntryInTheServerDatabase
login-limit: IntegerGivingConcurrentConnectionLimit
UserField: UserFieldValue
UserField: UserFieldValue

dn: radiuslist=check,radiusname=String,radiusclass=Native-User | …,o=radius
changetype: add
objectclass: top
objectclass: check
radiuslist: check
AttributeName: AttributeValue
AttributeName: AttributeValue
⋮
dn: radiuslist=reply,radiusname=String,radiusclass=
Native-User | …,o=radius
changetype: add
objectclass: top
objectclass: reply
radiuslist: reply
AttributeName: AttributeValue
AttributeName: AttributeValue
⋮
```

Adding Proxy Targets with LDIF

The sample LDIF entry shown in Figure 198: Adding Proxy Targets adds the proxy RADIUS target BIGCO.COM to the Steel-Belted Radius database.

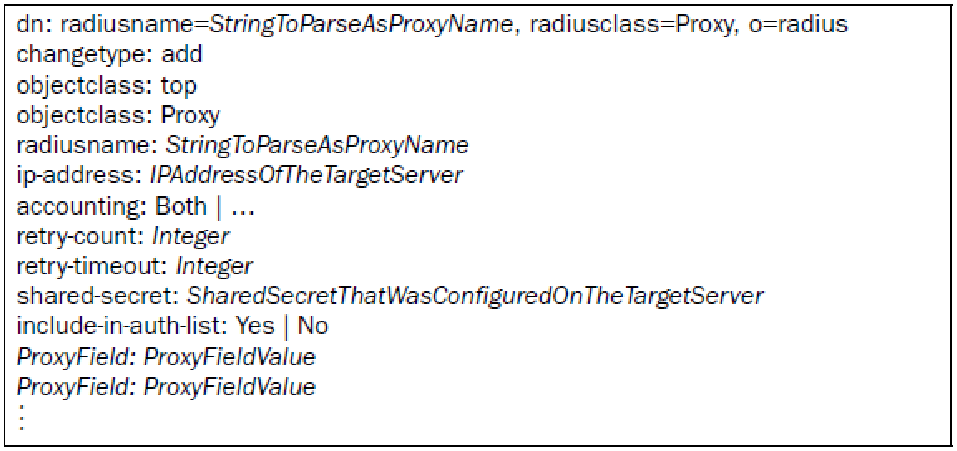
Figure 198: Adding Proxy Targets



```
dn: radiusname=BIGCO.COM,radiusclass=Proxy,o=radius
changetype: add
objectclass: top
objectclass: Proxy
radiusname: BIGCO.COM
ip-address: 194.132.5.89
accounting: both
retry-count: 3
retry-timeout: 5000
shared-secret: testing123
include-in-auth-list: no
```

The syntax in this LDIF entry is shown in Figure 199: LDIF Syntax.

Figure 199: LDIF Syntax

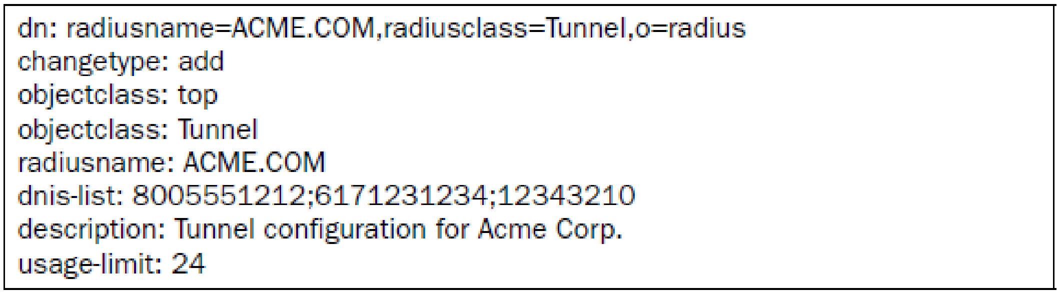


```
dn: radiusname=StringToParseAsProxyName, radiusclass=Proxy, o=radius
changetype: add
objectclass: top
objectclass: Proxy
radiusname: StringToParseAsProxyName
ip-address: IPAddressOfTheTargetServer
accounting: Both | ...
retry-count: Integer
retry-timeout: Integer
shared-secret: SharedSecretThatWasConfiguredOnTheTargetServer
include-in-auth-list: Yes | No
ProxyField: ProxyFieldValue
ProxyField: ProxyFieldValue
:
```

Adding Tunnels with LDIF

The sample LDIF entry shown in Figure 200: Adding Tunnels adds the tunnel ACME.COM to the Steel-Belted Radius database.

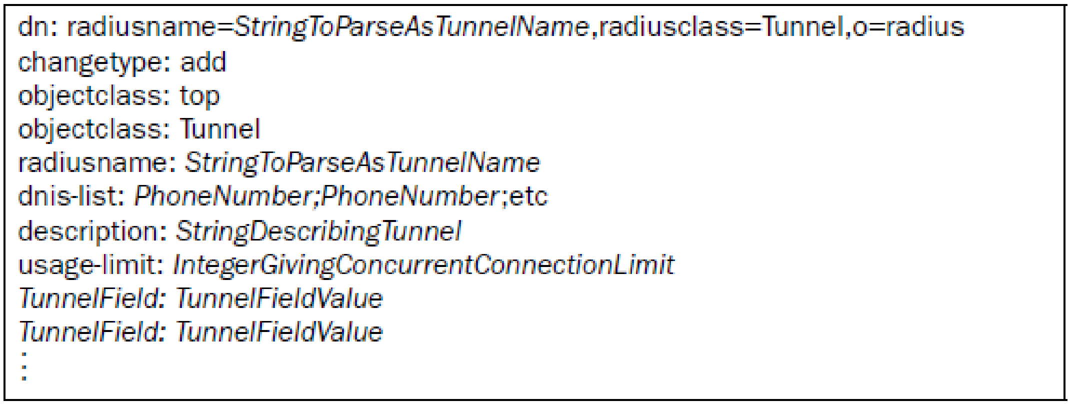
Figure 200: Adding Tunnels



```
dn: radiusname=ACME.COM,radiusclass=Tunnel,o=radius
changetype: add
objectclass: top
objectclass: Tunnel
radiusname: ACME.COM
dnis-list: 8005551212;6171231234;12343210
description: Tunnel configuration for Acme Corp.
usage-limit: 24
```

The syntax in this LDIF entry is shown in Figure 201: LDIF Syntax.

Figure 201: LDIF Syntax

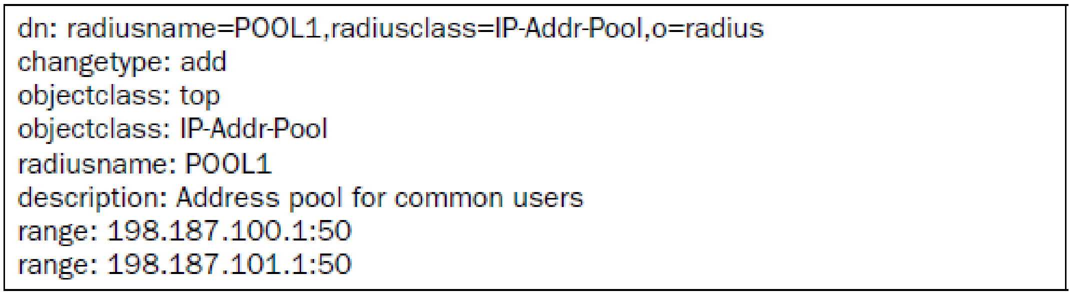


```
dn: radiusname=StringToParseAsTunnelName,radiusclass=Tunnel,o=radius
changetype: add
objectclass: top
objectclass: Tunnel
radiusname: StringToParseAsTunnelName
dnis-list: PhoneNumber;PhoneNumber;etc
description: StringDescribingTunnel
usage-limit: IntegerGivingConcurrentConnectionLimit
TunnelField: TunnelFieldValue
TunnelField: TunnelFieldValue
:
```

Adding IP Address Pools with LDIF

The sample LDIF entry shown in Figure 202: Adding IP Address Pools adds an IP address pool named POOL1 to the Steel-Belted Radius database.

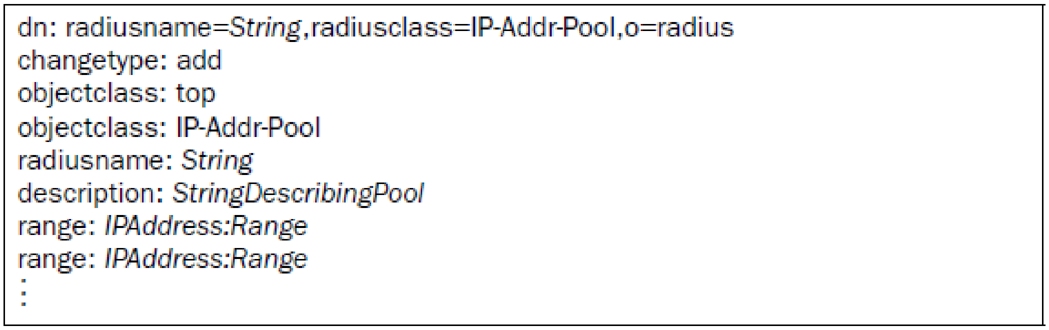
Figure 202: Adding IP Address Pools



```
dn: radiusname=POOL1,radiusclass=IP-Addr-Pool,o=radius
changetype: add
objectclass: top
objectclass: IP-Addr-Pool
radiusname: POOL1
description: Address pool for common users
range: 198.187.100.1:50
range: 198.187.101.1:50
```

The syntax in this LDIF entry is shown in Figure 203: LDIF Syntax.

Figure 203: LDIF Syntax

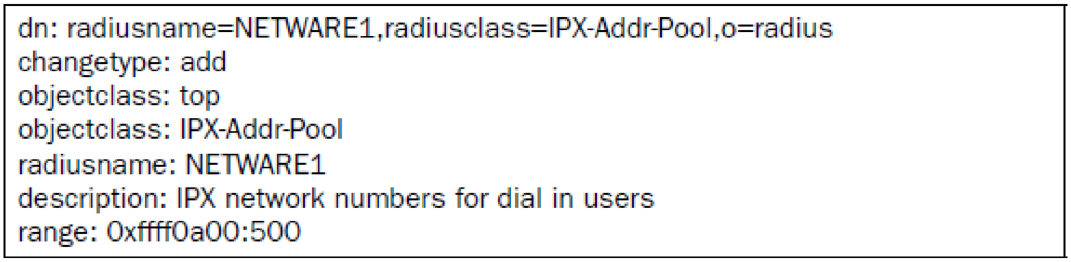


```
dn: radiusname=String,radiusclass=IP-Addr-Pool,o=radius
changetype: add
objectclass: top
objectclass: IP-Addr-Pool
radiusname: String
description: StringDescribingPool
range: IPAddress:Range
range: IPAddress:Range
:
```

Adding IPX Address Pools with LDIF

The sample LDIF entry shown in Figure 204: Adding IPX Address Pools adds an IPX address pool named NETWARE1 to the Steel-Belted Radius database.

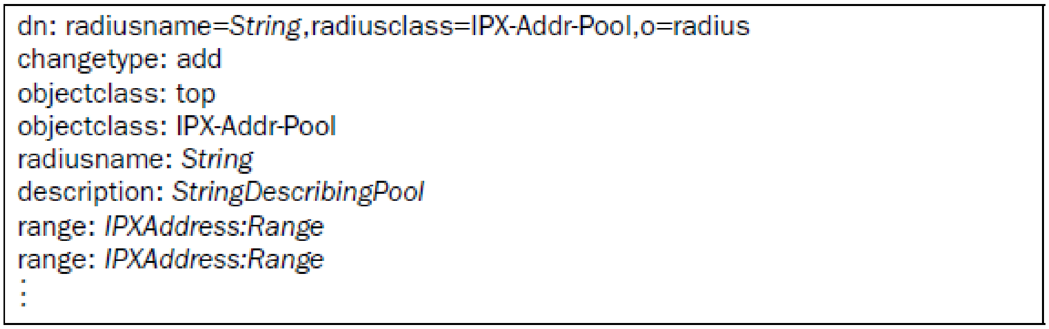
Figure 204: Adding IPX Address Pools



```
dn: radiusname=NETWARE1,radiusclass=IPX-Addr-Pool,o=radius
changetype: add
objectclass: top
objectclass: IPX-Addr-Pool
radiusname: NETWARE1
description: IPX network numbers for dial in users
range: 0xffff0a00:500
```

The syntax in this LDIF entry is shown in Figure 205: LDIF Syntax. You may provide multiple IPX address ranges using the range field.

Figure 205: LDIF Syntax

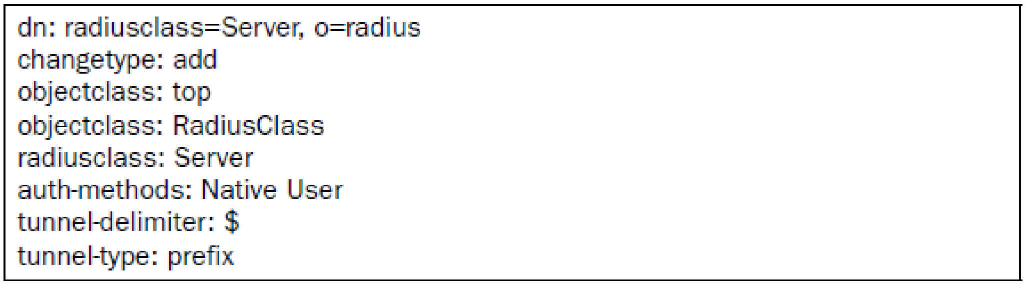


```
dn: radiusname=String,radiusclass=IPX-Addr-Pool,o=radius
changetype: add
objectclass: top
objectclass: IPX-Addr-Pool
radiusname: String
description: StringDescribingPool
range: IPXAddress:Range
range: IPXAddress:Range
⋮
```

Configuring a RADIUS Server with LDIF

The sample LDIF entry shown in Figure 206: Adding a RADIUS Server lets you configure your Steel-Belted Radius server by adding the Native User authentication method and defining conventions for tunnel name parsing.

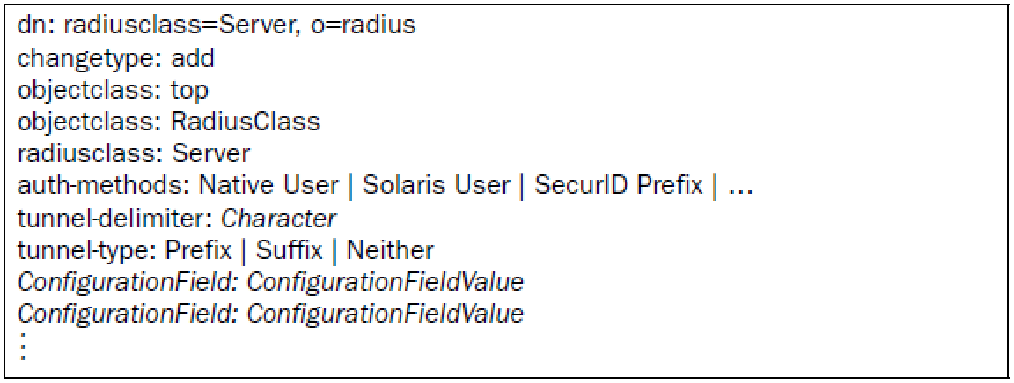
Figure 206: Adding a RADIUS Server



```
dn: radiusclass=Server, o=radius
changetype: add
objectclass: top
objectclass: RadiusClass
radiusclass: Server
auth-methods: Native User
tunnel-delimiter: $
tunnel-type: prefix
```

The syntax in this LDIF entry is shown in Figure 207: LDIF Syntax.

Figure 207: LDIF Syntax



```
dn: radiusclass=Server, o=radius
changetype: add
objectclass: top
objectclass: RadiusClass
radiusclass: Server
auth-methods: Native User | Solaris User | SecurID Prefix | ...
tunnel-delimiter: Character
tunnel-type: Prefix | Suffix | Neither
ConfigurationField: ConfigurationFieldValue
ConfigurationField: ConfigurationFieldValue
⋮
```

Statistics Variables

Server statistics counters record the number of certain types of events. The LCI allows you to read these statistics to monitor the performance of your Steel-Belted Radius server.

Note: The Enterprise Edition of Steel-Belted Radius with the optional LCI add-on does not support the Statistics items.

Counter Statistics

The statistics counters can be accessed via the LCI by executing the following one line command:

```
        ldapsearch -V 2 -h 127.0.0.1 -p 667 -D “cn=admin,o=radius” -w radius -s sub -T -b
        “radiusstatus=statistics,o=radius”    stattype=typeofstatus
```

The following sections illustrate the variables displayed for different settings of the stattype parameter.

stattype: server

```
dn:   stattype=server,radiusstatus=statistics,o=radius
        objectclass: top
        objectclass: radiusstatus
        radiusstatus:statistics
        stattype: server
        start-time: 2002/05/08 13:29:08
        up-time: 26188
        ip-address:   192.168.21.142
        version: v 2.20.33
        authentication-threads:   0
        accounting-threads: 0
        total-threads: 0
        max-auth-threads: 100
        max-acct-threads: 100
        max-total-threads: 200
        high-auth-threads: 2
        high-acct-threads: 0
        high-total-threads: 2
```

stattype: authentication

```
        dn:   stattype=authentication,radiusstatus=statistics,o=radius
        objectclass: top
        objectclass: radiusstatus
        radiusstatus:statistics
        stattype:authentication
        accept: 1
        reject: 0
        silent-discard: 0
        total-transactions: 8
```

```
invalid-request: 0
failed-authentication:   0
failed-on-check-list: 0
insufficient-resources: 0
proxy-failure: 0
rejected-by-proxy: 0
transactions-retried: 0
total-retry-packets: 0
```

stattype: accounting

```
dn:     stattype=accounting,radiusstatus=statistics,o=radius
objectclass: top
objectclass: radiusstatus
radiusstatus: statistics
stattype: accounting
start: 0
stop: 0
on: 0
off: 0
total-transactions: 0
invalid-request: 0
invalid-client: 0
invalid-shared-secret: 0
insufficient-resources: 0
proxy-failure: 0
transactions-retried: 0
total-retry-packets: 0
```

stattype: proxy

```
dn:     stattype=proxy,radiusstatus=statistics,o=radius
objectclass: top
objectclass: radiusstatus
radiusstatus: statistics
stattype: proxy
authentication: 0
```

```
accounting: 0

total-transactions: 0

timed-out: 0

invalid-response: 0

invalid-shared-secret: 0

insufficient-resources: 0

transactions-retried: 0

total-retry-packets: 0
```

Rate Statistics Rate

statistics are derived from existing counter statistics by taking time into consideration. Rate values calculated for each of these counter statistics consist of the following:

- Current rate—The rate measured over the most recent rate interval.
- Average rate—The rate measured since the Steel-Belted Radius server was started or since the last time statistics were reset to zero.
- Peak rate—The highest rate observed since the Steel-Belted Radius server was started or since the last time statistics were reset to zero.

  Additionally, there is a (read-only) time value used in calculations:

- Rate statistics seconds-per interval—The duration (in seconds) of the interval over which the rate statistics are gathered.

To read rate statistics from the LCI, you must set stattype: rate. This results in output such as the following:

```
rate-statistics-seconds-per-interval:   1

auth-request-current-rate:   0

auth-request-average-rate:   0

auth-request-peak-rate:   7

auth-accept-current-rate:   0

auth-accept-average-rate:   0

auth-accept-peak-rate:   1

auth-reject-current-rate: 0

auth-reject-average-rate: 0

auth-reject-peak-rate: 0

acct-start-current-rate:   0

acct-start-average-rate:   0

acct-start-peak-rate:0

acct-stop-current-rate:   0
```

```
acct-stop-average-rate:  0
acct-stop-peak-rate:0
proxy-auth-request-current-rate:   0
proxy-auth-request-average-rate:   0
proxy-auth-request-peak-rate:   0
proxy-acct-request-current-rate:   0
proxy-acct-request-average-rate:   0
proxy-acct-request-peak-rate:   0
proxy-fail-timeout-current-rate:   0
proxy-fail-timeout-average-rate:   0
proxy-fail-timeout-peak-rate:   0
proxy-fail-badresp-current-rate:   0
proxy-fail-badresp-average-rate:   0
proxy-fail-badresp-peak-rate:   0
proxy-fail-badsecret-current-rate:   0
proxy-fail-badsecret-average-rate:  0
proxy-fail-badsecret-peak-rate:  0
proxy-fail-missingresr-current-rate:   0
proxy-fail-missingresr-average-rate:   0
proxy-fail-missingresr-peak-rate:   0
proxy-retries-current-rate:   0
proxy-retries-average-rate:   0
proxy-retries-peak-rate:   0
proxy-auth-rej-proxy-current-rate:0
proxy-auth-rej-proxy-average-rate:0
proxy-auth-rej-proxy-peak-rate:0
proxy-acct-fail-prox-current-rate:   0
proxy-acct-fail-prox-average-rate:   0
proxy-acct-fail-prox-peak-rate:   0
proxy-auth-rej-proxy-error-current-rate:    0
proxy-auth-rej-proxy-error-average-rate:    0
proxy-auth-rej-proxy-error-peak-rate:0
```

Chapter 31
Configuring SQL Authentication

This chapter presents an overview of SQL authentication and describes how to configure SQL authentication in Steel-Belted Radius.

About SQL Authentication

Steel-Belted Radius can authenticate against records stored in an external SQL database. Any attribute or set of attributes, such as username and password, can be used to query the database.

External database authentication is typically used when an organization already has a large amount of user information stored in a SQL database, and this information is to be used to authenticate these users using RADIUS. Authentication against an existing database extends authentication services to user accounts without requiring an administrator to enter user information into the Steel-Belted Radius database.

Steel-Belted Radius offers the SQL authentication feature as a plug-in software module. Key features of the SQL plug-in include:

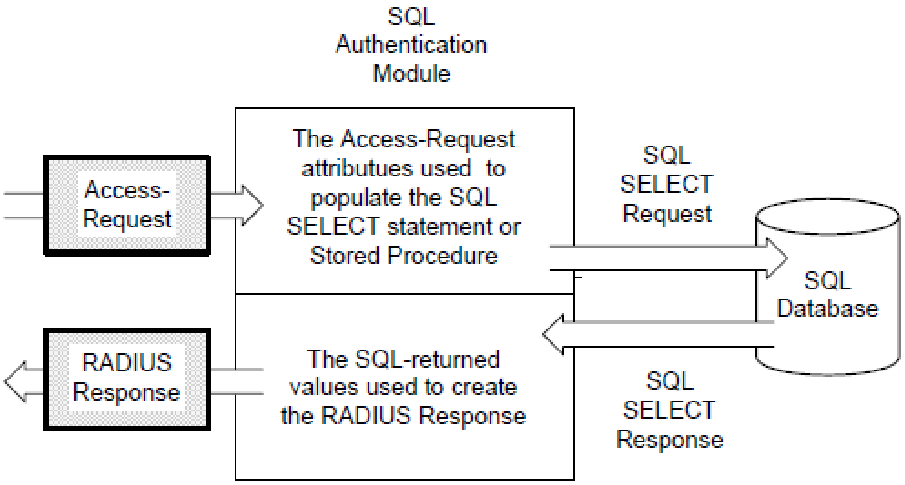
- The SQL statement is completely user-specified, allowing support of existing tables with existing field names and formats.
- The SQL statement supports a wide range of arithmetic and string expressions as part of the statement.
- The SQL statement is parameterized, so it is compiled once, and each execution uses variable data without need for recompilation.
- Multiple authentications may be overlapped at the same time.
- The SQL authentication method, which appears in the Authentication Policies panel in SBR Administrator, can be activated/deactivated and ordered with respect to other authentication methods.
- Multiple instances of the SQL authentication module can operate simultaneously, allowing authentication to multiple databases.
- If the database connection drops, it is automatically reestablished after a configurable timeout without Steel-Belted Radius being restarted.
- Data from the database can be returned as attributes in the Access-Accept message.

Note: While Steel-Belted Radius does its best to provide uniformity in the operation of databases from different vendors, differences occur, particularly in the way SQL statements are interpreted. The capabilities of the SQL authentication module depend on the capabilities of the underlying databases and their clients; things that work with one database may not work with another.

SQL Authentication Process

Any RADIUS attribute (or Steel-Belted Radius request variable) from the request can be used in an SQL SELECT statement. Any return list attribute (that is, a Steel-Belted Radius response variable) can be retrieved from a SQL database and returned in a RADIUS access response message.

Figure 208: SQL Authentication Process


SQL Authentication Module
Access-Request
The Access-Request attributues used to populate the SQL SELECT statement or Stored Procedure
SQL SELECT Request
SQL Database
RADIUS Response
The SQL-returned values used to create the RADIUS Response
SQL SELECT Response

Stored Procedures

A stored procedure is a sequence of SQL statements that form a logical unit and perform a particular task. You can use stored procedures to encapsulate a set of queries or operations that can be executed repeatedly on a database server. For example, you can code operations on an employee database, such as password lookup, as stored procedures that can be executed by application code.

Stored procedures can be compiled and executed with different parameters and results. Stored procedures can use any combination of input parameters (the values passed to the stored procedure at execution time) and output parameters (the values set or returned by the stored procedure to the calling application or environment).

You can write stored procedures for SQL that communicate with Steel-Belted Radius via input and output parameters to implement custom functions. Stored procedures let you use server-side processing on the SQL server to manipulate the information specified by variables. How you use these stored procedures depends on details specific to the implementation of SQL that you are using.

Note: Do not configure a stored procedure to call the same attribute more than once. Doing so may cause Steel-Belted Radius to fail.

For information on using stored procedures with the Oracle SQL database, see "Working with Stored Procedures in Oracle". For information on using stored procedures with the Microsoft SQL database, see "Working with Stored Procedures in MS-SQL".

Connectivity Issues

Steel-Belted Radius may encounter serious problems if the connection between Oracle and Steel-Belted Radius becomes unstable. The most common reasons for a connection becoming unstable are:

- Slow or unreliable network response times
- Interruptions in connectivity caused by intervening network devices, such as a firewall timing out the connection

  To prevent connectivity problems, consider implementation of one of the following solutions:

- To minimize problems caused by intervening firewalls, configure your firewall to pass traffic on the Oracle communications ports between the Steel-Belted Radius server and the Oracle server without restriction.

- To minimize network latency and firewall-related problems, move the Steel-Belted Radius server to the same network segment as the Oracle server.
- If moving your Steel-Belted Radius server is not feasible, locate a second Steel-Belted Radius server on the same network segment as your Oracle server, and configure your current Steel-Belted Radius server to proxy all authentication requests to this new device. This configuration will allow you to open RADIUS ports on the firewall only for the Steel-Belted Radius server (instead of opening RADIUS ports for all network access devices). Because proxy functions in Steel-Belted Radius do not require an uninterrupted connection to process requests, this solution allows you to retain your current firewall timeout settings.

Configuring SQL Authentication

You must configure both Steel-Belted Radius and the SQL database to support SQL authentication. The configuration procedure must be tailored to the database that you use. However, all procedures must give the following results:

- The required transport must be in place between SQL client software and the SQL server.
- The SQL server must be configured via a plug-in to coordinate with SQL client software.
- The Steel-Belted Radius server must be configured to communicate with the SQL client software to interact with the back-end SQL server to perform stored procedures or SQL queries.

Files

The following files establish settings for configuring SQL authentication in Steel-Belted Radius. For more information about these files, refer to the ***Steel-Belted Radius Reference Guide.***

Table 42: SQL Authentication Files

File Name	Function
sqlauth.aut	Configures settings for SQL authentication (Windows).
radsqljdbc.aut	Configures settings for SQL authentication using JDBC (Linux).

Using the SQL Authentication Header File

To configure SQL authentication, you must edit the authentication header files, radsqljdbc.aut (Linux with JDBC), or sqlauth.aut (Windows/ODBC), which are located in the same directory that contains the Steel-Belted Radius service or daemon. Most of these options may be left at their original settings; however, you must modify certain options to accommodate your own database.

After you complete your changes to the authentication header files and restart Steel-Belted Radius, the InitializationString value that you entered in the [Bootstrap] section of the header file appears in the Authentication Methods tab in the Authentication Policies panel. You can then enable, disable, or prioritize your SQL database like any other authentication method in the list.

Using Multiple SQL Authentication Methods

You can configure Steel-Belted Radius to authenticate users against more than one SQL database. Each database that you set up in this way becomes a separate selection in the Authentication Methods tab in the

Authentication Policies panel.

To add an additional database, create a new header file with extension .aut in the same directory as radsqljdbc.aut (Linux with JDBC), or sqlauth.aut (Windows/ODBC). You can give this file any name you like, provided its extension is .aut. At startup, Steel-Belted Radius enumerates all .aut files to create its list of authentication methods.

When creating the new file, start by copying the original .aut file. Be sure to change its InitializationString entry to a unique authentication method name; otherwise, Steel-Belted Radius has no way of distinguishing between the different methods in the authentication methods list.

Connecting to the SQL Database

Upon startup, the SQL authentication module connects to the database, using settings specified by a connect string specified in the header file. The connect string contains information such as the name and location of the database, and the password required to connect.

The connect string is passed to the database client to establish the connection. While a sample connect string is provided in the original header file, you must configure the Connect entry of the header file with a connect string appropriate to your database.

The password for database access must be provided as part of the connect string. If it is not:

- Linux: The connection fails.
- Windows: A pop-up window prompts you to enter the password before making the connection at startup and each time a reconnect is required.

If the initial attempt to connect to the database fails, or if a processing error occurs that the SQL authentication module interprets as a database connection failure, the SQL authentication module drops the connection and attempts to establish a new connection after a period of time. In the interim, all authentication requests are ignored.

The SQL authentication module uses an exponential back-off strategy in determining how long to wait before attempting a new connection, as well as how frequently this attempt should be made. After the first dropped connection, it waits a certain amount of time before attempting to reconnect. If this attempt to reconnect also fails, it waits for twice the amount of time before trying again; and so on, up to some maximum wait time. The initial and maximum wait times are configurable.

Note: “PWD” (password for DB) value of "Connect" string will be overwritten by their encrypted equivalents after restart.

Note: (Linux only): Detailed error information may not be available if there is an error processing the database logon at connect time. A numeric result code is displayed in the log. You may need to refer to product-specific documentation to decode this result code.

SQL Statement Construction

The authentication transaction is based on an SQL query that returns a password (and possibly other information) based on the name entered by the user attempting to log in.

While a sample SQL query is provided in the original header file, you must configure the SQL entry of the header file with a query appropriate to your database. The query you enter must be either a SQL SELECT or SQL EXECUTE statement that contains additional syntax elements which are preprocessed by the SQL authentication module.

The SQL authentication module executes SQL statements in parameterized form. This means that the SQL statement is compiled once, with parameter markers (usually question marks) as placeholders for data items that vary from one execution to the next. Only upon execution of the statement are the actual data values supplied.

The SQL statement you compose must not include parameter markers directly. Instead, the names of the parameters should be included where parameter markers would appear, in a format described below. The SQL authentication module translates the SQL statement provided, replacing parameter names with parameter markers prior to passing the SQL statement to the database engine.

The SQL statement can be very simple. Basically, all that is required is to look up a password and possibly some optional information based on a username. The SQL statement can also be quite complex; it can include inner joins, and it can contain expressions. The underlying database engine is responsible for handling the SQL statement; the SQL authentication module performs no interpretation of the SQL statement other than to translate parameter names to parameter markers.

Example:

```
SELECT password, profile, fullname FROM usertable WHERE username = %name/63s
```

As shown in this example, a parameter consists of a percent sign (%), the name of the parameter and a format specifier. **Table 43** lists SQL statement parameter names.

Table 43: SQL Statement Parameters

Item	Meaning for SQL Authentication
%OriginalUserName	The original full identification of the user, prior to any processing (that is, user@realm).
%User	The user portion of OriginalUserName (the section before '@').
%UserName	The full user identification (user and realm strings) after all stripping and processing has been performed.
%Name	Synonym for UserName.
%EffectiveUser	The name of the user (the section before '@') as presented to the authentication method. This may be a modified version of the original user name.
%Realm	The realm portion of the original user identification (the section after '@') as presented to the authentication method. This may be a modified version of the original realm name.
%EffectiveRealm	The realm portion of the user identification as presented to the method. This may be a modified version of the original realm name.
%NASName	The name of the network access device that originated the request. This may be the name of the RADIUS clients entry in the database or the value of the NAS-Identifier or NAS-IP-Address attribute.
%NASAddress	The address of the network access device, in dotted notation.

Item	Meaning for SQL Authentication
%NASModel	The make/model of the network access device, as specified in the Steel-Belted Radius database.
%Password	The PAP password.
%AllowedAccessHours	The times that the user is allowed to be logged in.
%RADIUSClientName	The name of the network access device, as specified in a RADIUS clients entry in the Steel-Belted Radius database.

Along with these parameters, any RADIUS attribute received in the Access-Request can be referred to by using an at-sign ('@') followed by the name of the attribute. If you need to specify a literal at-sign character in an SQL statement, such as in a UserName, you must use two at-signs in a row. For example:

```
SELECT foo FROM bar WHERE field = 'abc@@xyz'
```

Likewise, if you need to specify a literal percent character ('%') in an SQL statement you must use a two percent characters in a row.

The format specifier should describe the database storage format of the column that corresponds to the parameter. It consists of a slash ('/'), a length, and a type, which for SQL authentication is always 's' for string. For example, if the user's name is stored in the database as a string of up to 63 bytes, you would enter:

```
%name/63s
```

Note: Be sure to specify a length no greater than the actual field size in the database. The compilation of the SQL statement may fail if a parameter size greater than the actual field size is specified.

Password Parameters

Normally, the only parameter you'd include in the SQL statement is %name. The %password parameter is available to support databases containing non-unique usernames. For example, your database might allow two people named "George"; one with password "swordfish", and the other with password "martha". You can authenticate them correctly with the following query:

```
SELECT password, profile, fullname FROM usertable

WHERE username = %name/63s and password = %password/63s
```

You must return the password as the first column of the result to perform authentication. If the password is not returned in a password column or as an output parameter, no password authentication is performed.

In the following statement, %name is an input parameter used to look up a record.

```
SELECT profile FROM database WHERE username = %name
```

Since there's no password output parameter, no password authentication is performed. The [Results] section of the .aut file should look something like the following to work with the above SELECT statement:

```
[Results]

Password=0
```

```
Profile=1/50

Alias=0
```

If the record cannot be found in the database, the authentication attempt fails.

Note: If you are not using password checking for authentication, the Password parameter must be set to 0 in the [Results] section.

Overlapped Execution of SQL Statements

The SQL authentication module is multi-threaded. SQL authentication can be configured with a maximum number of simultaneous executions of any SQL statement, using the MaxConcurrent entry in the .aut file's [Settings] section.

If MaxConcurrent is set to 1, SQL execution occurs serially, and the SQL execution for each authentication request must complete before execution for the next request may begin.

By increasing MaxConcurrent, it may be possible to increase throughput by overlapping operations, especially if the database server is remote and a large part of the time to complete a statement execution is taken up by network latency. If the database server is local, the point of diminishing returns may be reached at a small value of MaxConcurrent, possibly even at 1 or 2. The optimum value is a matter of experiment.

Note: A setting of MaxConcurrent = 1 should be sufficient for all but the most demanding environments. Increase this value only slowly and conservatively.

You might expect that databases that are licensed by number of connections would debit a single connection regardless of how many SQL statements are active. This is not necessarily the case; some databases count each open compiled SQL statement against the licensed number of connections. So another factor that determines how MaxConcurrent should be set might be the database license.

%result Parameter

The %result parameter is a string value that can be returned as a column or stored procedure output parameter. The %result parameter can be used with or without password authentication.

The value expected to be returned in this parameter when authenticating a user can be specified in the SuccessResult entry of the [Settings] section. For example, if a user is successfully authenticated by the SQL authentication method, the result signifying success is the text string “okay”. This can be automatically checked by the following setting.

```
[Settings]

SuccessResult = okay
```

Note: The string comparison is case insensitive.

If the SQL statement succeeds but the SuccessResult value does not match the expected value returned from the database, Steel-Belted Radius issues a reject response, which can include any attributes and values configured in the [FailedSuccessResultAttributes] section of the *.aut file.

If PerformSuccessResultCheckAfterPasswordCheck=1 is specified and the SQL statement performs a password check that fails, Steel-Belted Radius does not process the SuccessResult and does not return the attributes from the [FailedSuccessResultAttributes] section in the reject response. If PerformSuccessResultCheckAfterPasswordCheck=1 is specified and the SQL statement performs a password check that succeeds but the SuccessResult value does not match the value returned from the database, Steel-Belted Radius issues a reject response that contains the attributes from the [FailedSuccessResultAttributes] section.

In the following statement, %password is passed to a stored procedure, which returns a %result of either "okay" or something else (that signifies a rejection):

```
BEGIN CheckUser(%name, %password, %result!o); END;
```

Another example might be a database of usernames, passwords, and account status. The administrator can enable a user by setting the user's account status to "okay"or disable the user by setting the account status to some other value, without having to delete the record. In the following statement, both password and result columns are checked:

```
SELECT password, result FROM database WHERE username = %name

    [Results]

    Password=1/50

    %Result=2/50

    Profile=0

    Alias=0
```

SQL Authentication and Password Format

Steel-Belted Radius supports the authentication of users residing in a SQL database, in which password values for the users are stored in one of the following formats: clear text, UNIXcrypt, Secured Hash Algorithm (SHA1+Base64 hash), MD4 hash, or enc-md5 reversibly-encoded password.

Hashed Passwords

Values in the Password column include a prefix that indicates how the password has been processed. The prefix is in clear text between curly braces { } and is immediately followed by a hash value computed from the password. If no prefix is present in the value retrieved from the table Password column, the entire password is assumed to be in clear text format. In summary:

- PasswordText indicates clear text format (no encryption)
- {crypt}HashHash indicates UNIXcrypt format
- {SHA}HashHashHash indicates SHA1+Base64 hash
- {SSHA}HashHashHashSalt indicates salted SHA1+Base64 hash
- {md4}HashHash indicates MD4 hash of the Unicode form of password
- {enc-md5}EncryptedEncrypted indicates a reversibly encrypted password

ℹ Note: Refer to RFC 2759 for details about how MS-CHAP-V2 produces an MD4 hash value.

ℹ Note: Although Steel-Belted Radius reads passwords encoded in enc-md5 format, you must purchase the Software Developer's Kit to convert clear-text passwords to this format.

UNIXcrypt is the standard hash algorithm that is used for the /etc/passwd file on Linux systems. This may be necessary if, for example, the standard user database on a Linux machine (the /etc/passwd file) is migrated to a SQL database, so that the values in the Password column of the SQL table are processed with UNIXcrypt.

Steel-Belted Radius may be configured to expect that the values retrieved from the SQL table Password column during authentication have been run through UNIXcrypt by adding the following entry into the [Settings] section of the SQL authentication header file:

```
PasswordFormat=3
```

Automatic Parsing

If PasswordFormat is set to 0, Steel-Belted Radius attempts to determine the password format automatically by parsing it. This is the recommended setting. Automatic parsing expects the password to be stored in one of the formats described in this section.

Note: The setting for automatic password parsing in older versions of Steel-Belted Radius (auto) has been deprecated.

Working with Stored Procedures in Oracle

The following notes discuss some considerations specific to Oracle, which uses the term ***package and package body*** when referring to stored procedures.

Assume you have a SELECT statement that extracts a user’s name, password, and profile from the table usertable when it receives the user’s name as an input parameter:

```
SELECT fullname, password, profile FROM usertable WHERE username =

%name/63s
```

To write a package called myPack1 that performs the equivalent function, you would enter the following sequence of commands:

```
Package myPack1

is

PROCEDURE myProc

(

name IN VARCHAR2,

passOUTVARCHAR2,

prof OUTVARCHAR2,

fName OUT VARCHAR2

);

End myPack1;
```

When referencing the package from sqlauth.aut, you would point to the package name myPack1 (not the procedure name myProc):

```
Package Body myPack1

is

PROCEDURE myProc

(
name IN VARCHAR2,
pass OUT VARCHAR2,
prof OUT VARCHAR2,
fName OUT VARCHAR2
)
IS
BEGIN
SELECT fullname INTO fName, password INTO pass, profile INTO prof FROM
usertable WHERE username = name;
END myProc;
End myPack1;
```

When you invoke the stored procedure, you would delineate each parameter as an input (!i), output (!o), or input/output (!io) variable. The presence of a !io or !o keyword indicates that the value returned from the database is to be included in the Access-Accept response as if it had been coded in the [Results] section. If a r value is included in the suffix (for example, !ir, !r, or !or), the parameter is expected to be an output parameter, and the attribute is to be treated as if it were included in the [FailedSuccessResultAttributes] section. Variables that are not specifically marked are considered input parameters by default.

Note : Do not configure a stored procedure to call the same attribute more than once. Doing so may cause Steel-Belted Radius to fail.

Correct: SQL= {call joeproc2 (@class!o)}

Incorrect: SQL= {call joeproc2 (@class!o, @class!o)}

You could replace the SELECT statement by invoking myProc as follows:

```
SQL=BEGIN myPack1.myProc(%name!i, %password!o, %profile!o, %fullname!o ); END;
```

When using input-output parameters with Oracle, you must set the DefaultResults setting to 0. Any other variables that need to be returned (such as Reply-Message) must be identified by the “!o” marker within the SQL statement.

Working with Stored Procedures in MS-SQL

A simple example of a stored procedure returns a result set in the same way as a Select statement. For example, assume you have a table with the following fields: username, password, Alias, and active, where all fields have the datatype varchar. You want a stored procedure that will return a password and alias when the username and password received in the request match entries in the database, provided that active field has a value of yes’.

Example 1

To create a simple stored procedure, run the following command sequence from MS Query Analyzer to create a stored procedure called rsp_getpword.

```
CREATE PROCEDURE rsp_getpword
@Unamevarchar(21),
@pword varchar(21)
AS
SELECT password, alias FROM authentication WHERE username = @Uname
AND password = @pword AND active = 'yes'
GO
```

This stored procedure can then be executed from a *.aut file as follows:

```
SQL= Execute rsp_getpword %username, %password
[results]
Password=1
Alias=2
```

Example 2

More complex stored procedures take input and output parameters in a manner similar to that used by Oracle. For example, assume you have a table with the following fields: username, password, profile, and active, where all fields have a datatype of varchar. You want a stored procedure that returns a password and profile when the username and password received in the request match a username and password in the database, provided that the active field has a value of yes.

First, to create the stored procedure, run the following command from MS query analyzer:

```
CREATE PROCEDURE rsp_authuser
@uname as varchar(20),
@pword as varchar(21) OUTPUT,
@profile as varchar(21)OUTPUT
AS
SELECT @pword =password,@profile=profile FROM authentication WHERE
username = @uname AND active = 'yes'
GO
```

This stored procedure can then be executed from a *.aut file as follows:

```
SQL= {call rsp_authuser (%username!i, %password!o, %profile!o)}
[results]
; No entries should be specified in results, everything but the header should be
commented out.
```

Chapter 32
Configuring SQL Accounting

This chapter presents an overview of SQL accounting and describes how to configure SQL accounting in Steel-Belted Radius.v

About SQL Accounting

Steel-Belted Radius can write RADIUS accounting information to an external SQL database, independently of the Steel-Belted Radius accounting log.

To set up an external database for use as a repository for RADIUS accounting data, you must place an .acc database configuration file in the same directory that contains the Steel-Belted Radius service (normally C:\RADIUS\Service) or daemon. This file must be modified to contain specialized information about your enterprise database.

Steel-Belted Radius offers the SQL accounting feature as a plug-in software module. Key features of the SQL plug-in include:

- The SQL statement is completely user-specified, allowing support of existing tables with existing field names and formats.
- The SQL statement can include a wide variety of arithmetic and string expressions.
- The SQL statement is parameterized, so it is compiled once, and each execution uses variable data without need for recompilation.
- Attribute and other data from the accounting request can be mapped to any parameter of the SQL statement (and hence to any field in the table) by means of a simple syntax.
- Different request types can be mapped to different SQL statements that may operate against distinct tables within the database.
- Multiple instances of a SQL statement can be overlapped for simultaneous execution.
- Multiple instances of the SQL accounting module can operate simultaneously, allowing logging to multiple databases.
- If the database connection drops, it is automatically reestablished after a configurable timeout without restarting Steel-Belted Radius.
- SQL accounting responses can return information.
- Stored procedures invoked by SQL accounting can make use of input parameters, record results, and return output parameters.

Note: While Steel-Belted Radius tries to provide uniformity in the operation of databases from different vendors, differences exist, particularly in the way SQL statements are interpreted. The capabilities of the SQL Authentication module depend on the capabilities of the underlying databases and their clients; things that work with one database may not work with another.

Stored Procedures

A stored procedure is a sequence of SQL statements that form a logical unit and perform a particular task. You can use stored procedures to encapsulate a set of queries or operations that can be executed repeatedly on a database server. For example, you can code operations on an employee database, such as password lookup, as stored procedures that can be executed by application code.

Stored procedures can be compiled and executed with different parameters and results. Stored procedures can use any combination of input parameters (the values passed to the stored procedure at execution time) and output parameters (the values set or returned by the stored procedure to the calling application or environment).

You can write stored procedures for SQL that communicate with Steel-Belted Radius via input and output parameters to implement custom functions. Stored procedures let you use server-side processing on the SQL server to manipulate the information specified by variables. How you use these stored procedures depends on details specific to the implementation of SQL that you are using.

For information on using stored procedures with the Oracle SQL database, see "Working with Stored Procedures in Oracle". For information on using stored procedures with the Microsoft SQL database, see "Working with Stored Procedures in MS-SQL".

Connectivity Issues

Steel-Belted Radius may encounter serious problems if the connection between Oracle and Steel-Belted Radius becomes unstable. The most common reasons for a connection becoming unstable are:

- Slow or unreliable network response times
- Interruptions in connectivity caused by intervening network devices, such as a firewall timing out the connection

To prevent connectivity problems, consider implementation of one of the following solutions:

- To minimize problems caused by intervening firewalls, configure your firewall to pass traffic on the Oracle communications ports between the Steel-Belted Radius server and the Oracle server without restriction.
- To minimize network latency and firewall-related problems, move the Steel-Belted Radius server to the same network segment as the Oracle server.
- If moving your Steel-Belted Radius server is not feasible, locate a second Steel-Belted Radius server on the same network segment as your Oracle server, and configure your current Steel-Belted Radius server to proxy all authentication requests to this new device. This configuration will allow you to open RADIUS ports on the firewall only for the Steel-Belted Radius server (instead of opening RADIUS ports for all network access devices). Because proxy functions in Steel-Belted Radius do not require an uninterrupted connection to process requests, this solution allows you to retain your current firewall timeout settings.

Configuring SQL Accounting

You must configure both Steel-Belted Radius and the SQL database to support SQL accounting. The configuration procedure must be tailored to the database that you use. However, all procedures must give the following results:

- The SQL server must be configured to be listening for client requests. Note that for SQL purposes, the Steel-Belted Radius server must be a client of the SQL server.
- The Steel-Belted Radius server must know the machine where the SQL server software runs, and it

must know the protocol and port used in communicating with that machine.

- The required transport must be in place between SQL client and server.

Files

The following files establish settings for configuring SQL accounting in Steel-Belted Radius. For more information about these files, refer to the Steel-Belted Radius Reference Guide.

Table 44: SQL Accounting Files

File Name	Function
radsqljdbc.acc	Configures settings for SQL accounting (Linux and JDBC).
sqlacct.acc	Configures settings for SQL accounting (Windows/ODBC).

Using the SQL Accounting Header File

To configure SQL accounting, you must edit the accounting header file (radsqljdbc.acc (Linux and JDBC) or sqlacct.acc (Windows/ODBC)), located in the same directory that contains the Steel- Belted Radius service (normally C:\RADIUS\Service) or daemon.

You must modify certain options in the accounting header file to accommodate your own database. After you update your accounting header file and restart Steel-Belted Radius, accounting proceeds as you have configured it.

Using Multiple SQL Databases

You can configure Steel-Belted Radius to log accounting transactions against more than one SQL database.

To add an additional database, create a new header file with extension .acc in the same directory as radsqljdbc.acc (Linux and JDBC), or sqlacct.acc (Windows/ODBC). You can give this file any name you like, provided its extension is .acc. At startup, Steel-Belted Radius enumerates all .acc files to create its list of accounting modules.

Note: When creating the new file, start by duplicating the original .acc file, then make whatever modifications are necessary.

Connecting to the SQL Database

Upon startup, the SQL accounting module connects to the database, based on a connect string specified in your accounting header file. The connect string contains information such as the name and location of the database, and the password required to connect.

The connect password will be overwritten by their encrypted value in accounting header file.

The connect string is passed to the database client to establish the connection. While a sample connect string is provided in the original header file, you must configure the Connect entry of the header file with a connect string appropriate to your database.

The password for database access must be provided as part of the connect string or the following results occur:

- Linux: The connection fails.
- Windows: A pop-up window prompts the user to enter a password before making the connection at

startup and each time a reconnect is required.

If the initial attempt to connect to the database fails, or if a processing error occurs that the SQL accounting module interprets as a database connection failure, the SQL accounting module drops the connection and attempts to establish a new connection after a period of time. In the interim, all authentication requests are ignored.

The SQL accounting module uses an exponential back-off strategy in determining how long to wait before attempting a new connection, as well as how frequently this attempt should be made. After the first dropped connection, it waits a certain amount of time before attempting to reconnect. If this attempt to reconnect also fails, it waits for twice the amount of time before trying again; and so on, up to some maximum wait time. The initial and maximum wait times are configurable.

Note: (Linux only): Detailed error information may not be available if there is an error processing the database logon at connect time. A numeric result code appears in the log. You may need to refer to product-specific documentation to decode this result code.

SQL Statement Construction

For each accounting request whose Acct-Status-Type is mapped to a SQL statement, that accounting request is logged to the backend database by executing the associated SQL statement.

While a sample SQL statement is provided in the original header file, you must configure one or more SQL entries of the header file with a statement appropriate to your database. Each SQL statement is typically an INSERT INTO statement and may contain additional syntax elements that are preprocessed by the SQL accounting module.

The SQL accounting module executes SQL statements in parameterized form. This means that the SQL statement is compiled once, with parameter markers (usually question marks) as placeholders for data items that vary from one execution to the next. Only upon execution of the statement are the actual data values supplied.

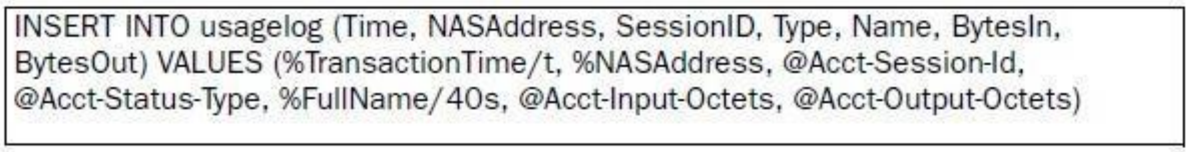
The SQL statement you compose must not include parameter markers directly. Instead, the names of the parameters should be included where parameter markers would appear, in a format described below. The SQL authentication module translates the SQL statement provided, replacing parameter names with parameter markers prior to passing the SQL statement to the database engine.

A SQL statement can be very simple. Basically, all that is required is to set fields of the database record with values from the request. The SQL statement can also be quite complex; it can include inner joins, and it can contain expressions. The underlying database engine is responsible for handling the SQL statement; The SQL accounting module performs no interpretation of the SQL statement other than to translate parameter names to parameter markers.

INSERT Statement and VALUES Section

The following is an example of a SQL INSERT statement that might be found in a Steel-Belted Radius .acc file.

Figure 209: INSERT Statement Example



```
INSERT INTO usagelog (Time, NASAddress, SessionID, Type, Name, BytesIn,
BytesOut) VALUES (%TransactionTime/t, %NASAddress, @Acct-Session-Id,
@Acct-Status-Type, %FullName/40s, @Acct-Input-Octets, @Acct-Output-Octets)
```

In the VALUES section, the names (between parentheses) represent the values inserted into the SQL table columns. To support the SQL accounting module, each item in the VALUES section must be prefixed with a @ sign or a % sign.

- @ indicates a RADIUS accounting attribute. The attribute name must also be listed in the account.ini file. This remains true even if the account.ini file is disabled.
- % indicates an item associated with the INSERT request that is not a RADIUS accounting attribute. **Table 45** lists the Steel-Belted Radius items that may be provided.

Table 45: Insert Statement Syntax

Item	Data Type	Meaning
%TransactionTime	Time	The date/time that the event occurred that is the subject of the request. Note: You should include the /t (timestamp) data type qualifier with the %TransactionTime argument in SQL statements. If you do not, the %TransactionTime output is formatted as character, with differing results on JDBC and Oracle.
%Time	Time	The date/time when the request is being processed. (This is later than %TransactionTime if the request is a retry.) Note: You should include the /t (timestamp) data type qualifier with the %Time argument in SQL statements. If you do not, the %Time output is formatted as character, with differing results on JDBC and Oracle.
%Type	String	The RADIUS accounting request type.
%NASAddress	IP address	The IP address of the requesting RAS.
%NASName	String	The name of the network access device that originated the request. This may be the name of the RADIUS client entry in the database or the value of the NAS-Identifier or NAS-IP-Address attribute.
%NASModel	String	The RAS make/model.
%FullName	String	The full name of the logged in user.
%AuthType	String	The method by which the user was authenticated.
%RADIUSClientName	String	The name of the network access device, as specified in a RADIUS client entry in the Steel-Belted Radius database.

A format specifier may appear immediately following each parameter. The format specifier should describe the database storage format of the column that corresponds to the parameter. It consists of a slash ('/'), possibly a length, and a data type. **Table 46** lists the available data types.

Table 46: Data Types

Format Specifier	Meaning
/xs	A text string of length x. /s indicates a string with the default length of 256.

Format Specifier	Meaning
/xb	A binary data string of length x. A binary string is different from an text string in that it is not NULL-terminated and is not restricted to ASCII characters. /b indicates a binary data string with the default length of 256.
/n	32-bit integer
/n8	8-bit integer
/n16	16-bit integer
/n32	32-bit integer (same as /n)
/nxx	Integer xx bits in length. For example, /n64 indicates a number with a length of 64 bits.
/t	Timestamp

Note: Steel-Belted Radius supports integers larger than 32 bits by manipulating them as binary data strings. Other database/operating-system combinations may not allow for integers larger than 32 bits.

If a format specifier is not present in the SQL statement syntax, Steel-Belted Radius automatically defaults to an appropriate specifier based on the actual parameter type. For example, @Acct-Input-Octets is a number, and defaults to /n.

Note: For strings, always include a format specifier, and be sure to specify a length no greater than the actual field size in the database. The compilation of the SQL statement may fail if a length greater than the actual field size is specified. If no format specifier is present, the length defaults to 256 characters, which may cause the compilation to fail.

Steel-Belted Radius automatically attempts to convert between the internal format of a parameter and its format in the database, as described by the format specifier. In most cases, the formats are equivalent; if not, Steel-Belted Radius performs reasonable conversions.

Table 47 lists the internal formats and their compatible database formats:

Table 47: Internal Formats and Compatible Database Formats

Internal Format	Compatible Database Formats
Binary data string	/b, /xb, /n, /n8, /n16, /n32
Number	/n, /n8, /n16, /n32, /xs, /s
String	/xs, /s

Internal Format	Compatible Database Formats
Time (seconds since 1/1/70)	/t, /n, /n32, /xs, /s
IP address	/n, /n32, /xs, /s

As you write the INSERT statement for your SQL accounting header file (.acc), we recommend the following syntax checklist:

- The column names and their corresponding attributes in the VALUES section are order-dependent. In the example shown in Figure 209: INSERT Statement Example, the %TransactionTime/t value would be inserted into the Time column (and formatted as a timestamp), the %NASAddress value would be inserted into
  the NASAddress column, and so forth. The ordering of these settings is critical to proper RADIUS accounting data insertion, since each column in the SQL table may be a specific data type, such as varchar or int.
- The use of left and right parentheses '( )', the backslash '\', the forward slash '/' and even blank spaces are all extremely important and must be exact. You can add as many columns and attributes as you want for your RADIUS accounting needs; however, be sure to model your INSERT statement syntax on the example shown in Figure 209: INSERT Statement Example.
- An attribute listed incorrectly in the VALUES section, such as @Acct_Session-Id rather than @Acct-Session-Id, causes the SQL statement to fail during a RADIUS accounting transaction. The attribute's syntax must match its corresponding attribute name in the account.ini file, which in turn matches the attribute's name in the appropriate dictionary file, which allows Steel-Belted Radius to process the attribute correctly when it is received from the RAS (the RADIUS client).
- An attribute listed in the VALUES section that is missing its prefix of '@' or '%' causes the SQL statement to fail during a RADIUS accounting transaction.
- If a carriage return is present within the INSERT statement without the backslash '\' to indicate the end of the line, the SQL statement fails during a RADIUS accounting transaction.
- Do not make the lines in the .acc file too long. There is a line length limit of 255 characters. Use the backslash '\' to indicate the end of the line before that limit is reached. If a line exceeds this limit, the SQL statement fails during a RADIUS accounting transaction.

Using Multiple SQL Statements

The most common use of accounting is to track user sessions. However, accounting requests are generated when the RAS starts up and shuts down; and, vendor-specific uses of accounting are used to track other RAS phenomena. Clearly, it might be advisable to log different types of accounting events to different tables.

The Acct-Status-Type attribute of an accounting request indicates the request type. You may, if you like, create multiple SQL statements, and map each Acct-Status-Type to one of these SQL statements. The different statements may update different tables in the database, but they all share the single database connection.

Overlapped Execution of SQL Statements

The SQL accounting module is multi-threaded. SQL accounting can be configured with a maximum number of simultaneous executions of any SQL statement, using the MaxConcurrent entry in the .acc file's [Settings] section.

If MaxConcurrent is set to 1, SQL execution occurs serially, and the SQL execution for each accounting request must complete before execution for the next request may begin.

By increasing MaxConcurrent, it may be possible to increase throughput by overlapping operations, especially if the database server is remote and a large part of the time to complete a statement execution is taken up by network latency. If the database server is local, the point of diminishing returns may be reached at a small value of MaxConcurrent, possibly even at 1 or 2. You can find the optimum value for your system by experimentation.

Note: A setting of MaxConcurrent = 1 should be sufficient for all but the most demanding environments. Increase this value slowly and conservatively.

MaxConcurrent determines the maximum overlap for executing any single SQL statement. Multiple SQL statements for different request types are not interdependent, and executions of one statement do not affect executions of a different statement.

You might expect that databases that are licensed by number of connections would debit a single connection regardless of how many SQL statements are active. This is not necessarily the case; some databases count each open compiled SQL statement against the licensed number of connections. The database license may also have an influence on the optimum setting for MaxConcurrent.

SQL Accounting Return Values

SQL accounting statements can return information in RADIUS attributes in an accounting response. This is useful only if you are using a client that expects and supports attributes embedded in a RADIUS accounting response message.

Stored procedures can also return output parameters. The way in which these stored procedures are called depends on your operating system:

- To call an Oracle stored procedure in a Windows environment:

  call(storedProcedure(parameters...))

Accounting Stored Procedure Example

A simple stored procedure can return a result set in the same way as a Select statement. For example, assume you have a table with the following fields: username, password, Alias, and active, where all fields have the datatype varchar. You want a stored procedure that will return a password and alias when the username and password received in the request match entries in the database, provided that active field has a value of yes'.

The following example executes a stored procedure to update an accounting table in Steel-Belted Radius.

1. Create an accounting table by executing the following command:

   create table accounting

   (TransactionDate varchar(20), Username varchar(21), SessionID varchar(12),

   NASIPAddr varchar(15), NASPort varchar(5), UserIPAddr varchar(15), CallingNum

   varchar(12), CalledNum varchar(12),

   type varchar(4), Sessiontime varchar(14), Disconnect varchar(12))

2. Create a rsp_account stored procedure that can be called by a *.acc file.

```
create procedure rsp_account
@transactiontime   varchar(21),
@username   varchar(21),
@AcctSessionID   varchar(21),
@NASIPAddressvarchar(21),
@NASPORTTYPEvarchar(21),
@FRAMEDIPADDRESS    varchar(21),
@callingstationid   varchar(21),
@calledstationid   varchar(21),
@TYPEvarchar(21),
@ACCTSESSIONTIME    varchar(21),
@ACCTTERMINATIONCAUSE    varchar(21)
AS
INSERT INTO Accounting (TransactionDate, username, SessionID, NASIPAddr,
NASPort, UserIPAddr, CallingNum, CalledNum, type, Sessiontime, Disconnect)
VALUES (@transactiontime, @username, @AcctSessionID, @NASIPAddress,
@NASPORTTYPE,@FRAMEDIPADDRESS,@callingstationid,@calledstationid,
@TYPE,   @ACCTSESSIONTIME,   @ACCTTERMINATIONCAUSE)
```

3. Create the mysqlacct.acc file to call the rsp_account stored procedure.

   Note: The mysqlacct.acc file uses an SQL=EXECUTE procedure_name value1,...valueN statement instead of an SQL=INSERT into table (column1, ...columnN) Values (value1,...valueN), since the stored procedure will do the INSERT action. You would configure the CONNECT statement to reflect your operatingenvironment.

```
[Bootstrap]
LibraryName=sqlacct.dll
Enable=1
InitializationString=
[Settings]
Connect=DSN=<dsn_name_here>;UID=<username_for_dB>;PWD=<password_for
_dB>
ConnectTimeout=25
```

```
WaitReconnect=2

MaxWaitReconnect=360

ParameterMarker=?

loglevel=2

[Type]

1=User

2=User

3=User

[Type/User]

SQL=Execute rsp_account %transactiontime/t, \

        @user-name/21s, \

        @Acct-Session-ID/12s, \
        @NAS-IP-Address/15s, \

        @NAS-PORT-TYPE/5s, \

        @FRAMED-IP-ADDRESS/15s,  \

        @calling-station-id/12s,  \

        @called-station-id/12s, \

        %TYPE/4s, \

        @ACCT-SESSION-TIME/14s,  \

        @ACCT-TERMINATION-CAUSE/12s

ConcurrentTimeout=30

MaxConcurrent=2
```

Chapter 33
Configuring LDAP Authentication

This chapter presents an overview of LDAP authentication and describes how to configure LDAP authentication in Steel-Belted Radius.

About LDAP Authentication

Steel-Belted Radius can authenticate against records stored in an external LDAP database. Any attribute(s), such as username and password, can be used to query the database.

External database authentication is typically used when an organization has a large amount of user information stored in an LDAP database, and wants to authenticate these users using RADIUS. Authentication against an existing LDAP database extends authentication services to user accounts without requiring an administrator to enter user information into the Steel-Belted Radius database.

Steel-Belted Radius offers LDAP authentication as a plug-in software module. Key features of the LDAP plug-in include the following:

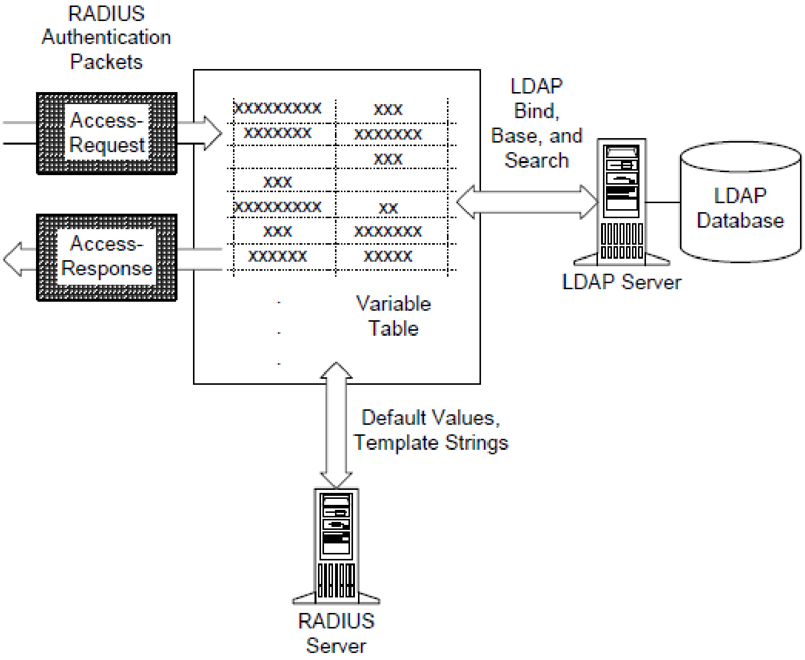
- LDAP Version 3 is supported.
- SSL/TLS is supported on Linux platform.
- TLS is supported on Windows platform.
- You can authenticate via LDAP Bind or via a password returned from an LDAP Search request (BindName).
- A single Search request or a sequence of Search requests can be specified.
- Bind, Base, and Search strings can include variables.
- New Bind parameters can be specified during a sequence of searches.
- Other authentication credentials can be specified in a string that can include variable values.
- Variables may be set from the RADIUS request packet and from LDAP Search results.
- Variables may be used to specify RADIUS response attributes and other response information.
- The RADIUS response can include RADIUS attributes found in the LDAP database, or it can reference a Steel-Belted Radius profile or user entry.
- Several features similar to SQL authentication are supported, such as round-robin load balancing, the “server of last resort,” and activation targets.
- Decorated usernames can be parsed into two variables within the variable table. For example, simon@xyz.com would be parsed into simon and xyz.com for use later in the authentication process.
- The variable table allows both attributes and %Profile in the [Response] section.
- Conditional search logic is supported by branching using the OnFound and OnNotFound fields.

LDAP Variable Table

The LDAP Variable Table lets you translate a RADIUS request into an LDAP lookup. At the beginning of each

LDAP authentication request, Steel-Belted Radius creates a Variable Table. Attributes and other information from the RADIUS request are entered in the Variable Table for use in LDAP Bind, Base, and Search strings. When attributes are returned by LDAP requests, they too are entered in the Variable Table. Finally, selected information from the Variable Table is returned to the RADIUS client in the RADIUS response packet.

Figure 210: Role of the Variable Table in LDAP Authentication



Types of LDAP Authentication

To design an LDAP authentication method, consider how you want to validate the username and password.

The LDAP plug-in offers two techniques for validating the username and password. Each header file that you write to control LDAP authentication must employ Bind or BindName. The differences between the two techniques have to do with how Steel-Belted Radius connects to the LDAP server and whether the username/ password validation is performed by the LDAP server or by Steel-Belted Radius.

BindName Authentication

When you use BindName authentication, your LDAP header file provides Steel-Belted Radius with the username and password of an account on the LDAP server. This must be an account that has privileges to access all of the information that you require to authenticate users. In the LDAP header file, you provide the username in the BindName parameter, and the password in the BindPassword parameter.

BindPassword parameter will be encrypted and overwritten in the LDAP header file after restart.

After you complete the LDAP header file, each time Steel-Belted Radius starts up, it executes a Bind request to the LDAP server using the BindName and BindPassword parameters as its credentials. If the LDAP server can validate these credentials, a connection is established between the two servers. This connection remains “up” all the time. It is disconnected only if the Steel-Belted Radius server or the LDAP server goes down, and it’s re-established as soon as possible after the “down” server comes back up. The LDAP header file offers a number of connection and re-connection timeouts and other parameters that regulate this relationship.

Any time authentication via LDAP is required, Steel-Belted Radius consults the corresponding LDAP header file. When you use BindName authentication, this file must contain a Search command that maps the username from the Access-Request to a password attribute in the LDAP database. The Search may retrieve other LDAP attributes as well. When the Search returns its results, Steel-Belted Radius compares the value of the password returned from the LDAP database with the password from the incoming Access-Request. If the two values are the same, the password is considered validated.

When the connection to the LDAP server is established using BindName, multiple authentications can be performed at the same time over the same connection.

Bind Authentication

When you use Bind authentication, Steel-Belted Radius authenticates connection requests by attempting to Bind to the LDAP server using the username and password from the incoming Access-Request or from a configured username and password. If this Bind request succeeds, the password is validated. This is essentially "pass-through" authentication; Steel-Belted Radius presents an LDAP user's credentials to the LDAP server and asks to have them validated.

In the simplest case, a single connection is established for each Access-Request and is kept open only long enough for the LDAP server to validate the password and respond to any Search requests. Then Steel-Belted Radius closes the connection and completes any processing that remains to generate an Access-Response.

A more sophisticated search technique can take advantage of flexible Bind, which allows you to allocate a sequence of connections for each Access-Request. Each in turn is kept open only long enough for the server to process each search criterion. Then Steel-Belted Radius closes the connection and completes any processing that remains to generate an Access-Response.

Attributes and LDAP Authentication

A username and password may be all the information that you require to authenticate users. However, the LDAP plug-in offers a number of techniques for working with checklist and/or return list attributes, should you need them.

Configuring LDAP Authentication

To configure an LDAP authentication method, you must edit the header file that controls the LDAP authentication sequence.

Table 48 summarizes the process of configuring an LDAP authentication method for Steel-Belted Radius. It lists the sections that you must edit in the header file to accomplish each step. No step may be omitted. You must at least consider the entries that you want to put in each section of the header file, even if you decide to leave most of that section blank.

Table 48: LDAP Authentication Header File Topics

Step	LDAP Configuration Task	.aut File Sections
1	Decide how you want Steel-Belted Radius to validate RADIUS access requests. Two major areas of choice are described above: (1) Bind or BindName; and (2) Profile, Alias, or attribute list.	All sections
2	Determine which incoming RADIUS attributes are required to perform the LDAP search.	[Response]
3	Determine which LDAP attributes support are required to perform the LDAP search.	[Attribute/name]

Step	LDAP Configuration Task	.aut File Sections
4	Design Search template(s) that can find the necessary data in your LDAP database schema.	[Search/name]
5	Extract the data from the incoming RADIUS packet that Steel-Belted Radius will use to perform the LDAP Bind and Search requests.	[Request]
6	Select defaults that you want Steel-Belted Radius to use when corresponding values are not provided.	[Defaults]
7	Enable connections between the Steel-Belted Radius server and LDAP server(s).	[Server] [Server/name] [Settings] [Failure]
8	Enable the LDAP plug-in and name the authentication method.	[Bootstrap]

The order in which you should edit header file sections is the reverse order in which Steel-Belted Radius processes them. The processing sequence is described in "LDAP Authentication Sequence".

Supporting Secure Sockets Layer/Transport Layer Security

Linux platform:

1. Set SSL in the [Settings] (or[Server/name]) section to 1.
2. Set the Certificates field in the [Settings] section to the path where the SHA1 and SHA2 certificate files are located. The certificate should be in .cer format. The name of the file should be included. For example: Certificates=/<Certificate path>/xyz.cer
3. Set the Host[server] section in the URI format.
   a. If SSL, the Host should be in URI format like ldaps://<LDAP Server IP>:<SSL Port of the LDAP server>
   b. If TLS, the host should be in URI format like ldap://<LDAP Server IP>:<TLS Port of the LDAP server>
4. While TLS/SSL support, Disable Port [server] Section in ldapauth.aut file. "

Windows platform:

1. Set SSL in the [Settings] (or [Server/name]) section to 1.

2. Certificate related configurations:

   a. If the certificate of the server is trusted, i.e. signed by a Trusted Certificate Authority (example: Verisign), then there is no configuration needed.
   b. If the certificate of the server is untrusted, then the certificate has to be installed as trusted in the store of Computer account option present in Microsoft Management Console.
      Go to MMC → File → add or remove snap-in. Select the certificates option and click Add. Then select Computer Account option and click Finish.
      Expand Certificates tree and expand Trusted CA / any other option according to the requirement. Right click Certificates and select All Tasks -> Import. Import the certificate which has to be trusted.

3. Set the port in the [Server] section to the TLS port of the LDAP server.

Files

The following file establishes settings for LDAP authentication. For more information about this file, refer to the *Steel-Belted Radius Reference Guide*.

Table 49: LDAP Authentication Files

File Name	Function
ldapauth.aut	Specifies settings for LDAP authentication in Steel-Belted Radius.

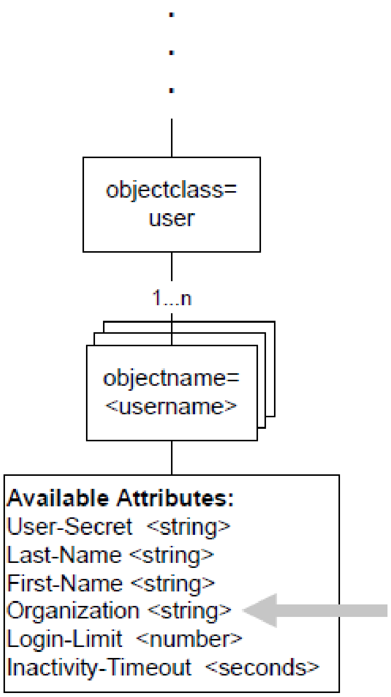
LDAP Database Schema

The most important factor in the success of your LDAP authentication methods is the design of your LDAP database schema. It's assumed that you already have a schema in place.

Often, you can use the LDAP plug-in without changing the LDAP database schema at all. In Figure 211: Capitalizing on an Existing Schema for LDAP Authentication, the user record already provides an LDAP attribute called Organization. If you intend to grant connection privileges according to the organization to which each user belongs, you can create profiles in the Steel-Belted Radius database whose names match the strings you are already using for the Organization attribute. You can then create an LDAP authentication header file that retrieves the value of the Organization attribute from the LDAP database and returns it to Steel-Belted Radius as the name of the profile to use.

Note: If you are using BindName authentication, you need to be able to identify which LDAP attribute contains the user's password. In the schema below, this attribute is called User-Secret.

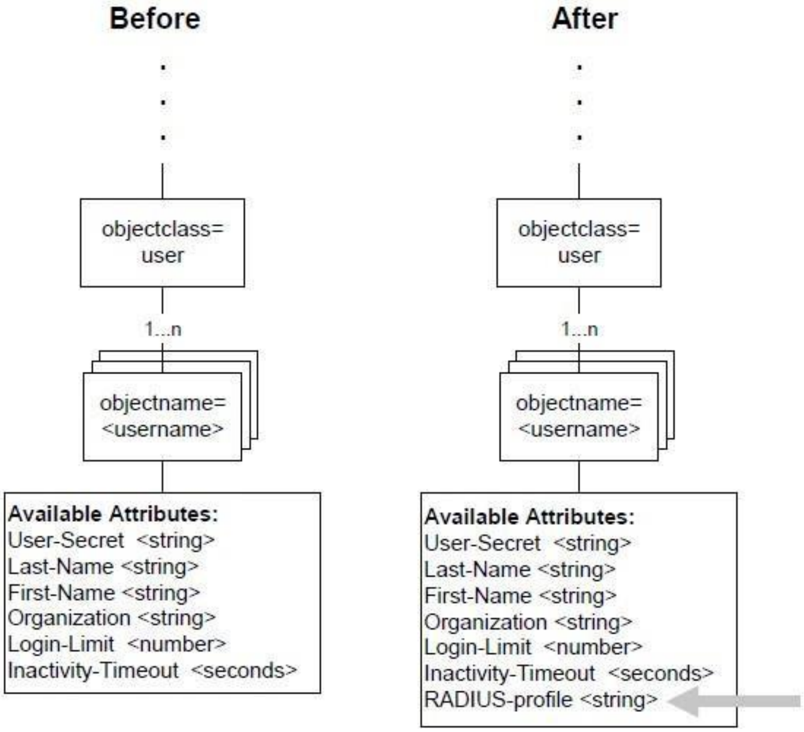
Figure 211: Capitalizing on an Existing Schema for LDAP Authentication



When the authentication strategy you have chosen requires data that is not currently in the schema, you might need to modify the schema.

The name of a Steel-Belted Radius profile is a typical example. Consider the example shown in Figure 212: Modifying a Schema to Enhance LDAP Authentication. If you want to assign connection privileges to users in some way other than by Organization, and no other LDAP attribute seems appropriate, you can add an LDAP attribute that names a profile. In Figure 212: Modifying a Schema to Enhance LDAP Authentication, this attribute is called RADIUS-Profile. This attribute contains a string value that can be set to the name of a profile defined in the Steel-Belted Radius database.

Figure 212: Modifying a Schema to Enhance LDAP Authentication



Note: LDAP concepts and the details of your own LDAP schema are entirely outside the scope of this chapter. The instructions in this chapter are provided to help you to make the LDAP plug-in work with an existing LDAP database or databases to provide a Steel-Belted Radius authentication method. The instructions assume that you already have a working knowledge of LDAP syntax and conventions. For details about LDAP, please refer to your usual LDAP information source.

LDAP Authentication and Password Format

Steel-Belted Radius supports authentication of users whose records reside in an LDAP table in which password values are stored in one of the following formats: clear text, UNIXcrypt, Secured Hash Algorithm (SHA1+Base64 hash), MD4 hash, or enc-md5 reversibly-encoded password.

Hashed Passwords

Encoded values include a prefix that indicates how the password has been processed. The prefix is in clear text between curly braces '{' '}' and is immediately followed by a hash value computed from the password. If no prefix is present in the value retrieved, the entire password is assumed to be in clear text format. In summary:

- PasswordText indicates clear text format (no encryption)
- {crypt}HashHash indicates UNIXcrypt format
- {SHA}HashHashHash indicates SHA1+Base64 hash
- {SSHA}HashHashHashSalt indicates salted SHA1+Base64 hash

- {md4}HashHash indicates MD4 hash of the Unicode form of password
- {enc-md5}EncryptedEncrypted indicates a reversibly encrypted password. (Note that, although Steel-Belted Radius reads passwords encoded in this format, you must purchase the Software Developer's Kit to convert clear-text passwords to this format.)

You can configure Steel-Belted Radius to expect that the values retrieved from a table have been run through UNIXcrypt by adding the following entry into the [Settings] section of the LDAP authentication header file:

```
PasswordFormat=3
```

Automatic Parsing

If PasswordFormat is set to 0, Steel-Belted Radius attempts to determine the password format automatically by parsing it. This is the recommended setting. Automatic parsing expects the password to be stored in one of the formatsdescribed in this chapter.

This technique is useful if clear text passwords are available to Steel-Belted Radius (that is, if PAP is used). If you set PasswordFormat to 0, the stored password can be returned to Steel-Belted Radius still encrypted, and the comparison with the password received from the RADIUS client can be done on the Steel-Belted Radius side.

ℹ Note: The setting for automatic password parsing in previous versions of Steel-Belted Radius (auto) has been deprecated.

LDAP Authentication Sequence

The sequence of an LDAP authentication transaction is controlled by the LDAP authentication header file as follows:

1. The Variable Table is initialized to default values as specified in the [Defaults] section. All variables that are not listed in the [Defaults] section are initialized to null values.
2. The values of RADIUS attributes in the Access-Request are copied to the Variable Table, as specified in the [Request] section.
3. If a Bind entry was specified in the [Settings] section, authentication via LDAP Bind is now performed. The Bind entry is used as a template to construct a bind string, using replacement values from the Variable Table. An LDAP Bind is then performed to authenticate the user.
4. An LDAP Search request is performed for each [Search/name] section specified. You may specify zero or more separate Search requests. For each Search request, LDAP Base and Filter strings are constructed from templates, using replacement values from the Variable Table. These Base and Filter strings are then transmitted to the LDAP server in a Search request. Each attribute/value pair returned by the LDAP Search is used to set the value of the corresponding entry in the Variable Table. Also, the DN returned by the search may be used to set a variable.
5. If a %Password entry appears in the [Response] section, authentication is now performed. The password entered by the user is validated against the value that appears in the %Password variable, and the user is rejected if the passwords don't match.
6. If a %Profile entry appears in the [Response] section, the value of the %Profile variable is used to look up a Profile entry in the Steel-Belted Radius database. The checklist and return list attributes in that Profile are used to validate the request and return an appropriate response.
7. If a %Alias entry appears in the [Response] section, the value of the %Alias variable is used to look up a Native User entry in the Steel-Belted Radius database. The current transaction is treated as if it

came from the "alias" user; that is, the checklist and return list attributes of the alias user are used to validate the request and return an appropriate response.

8. If neither a %Profile nor a %Alias entry appears in the [Response] section, then RADIUS attributes for the response packet are created from the Variable Table, based on attribute entries in the [Response] section.

LDAP Authentication Examples

This topic provides examples of LDAP authentication header file syntax. The examples illustrate how you might:

- Authenticate passwords (Bind or BindName).
- Specify checklist and return list attributes (list the attributes or name a profile entry in the Steel-Belted Radius database).

Bind Authentication with Default Profile

The following example is a simple LDAP authentication header file. Every user is authenticated using a Bind request to the LDAP database. The same Steel-Belted Radius attribute profile is applied to every Access-Request.

```
[Settings]
MaxConcurrent=1
Timeout=20
ConnectTimeout=25
QueryTimeout=10
WaitReconnect=2
MaxWaitReconnect=360
Bind=uid=<User-Name>,ou=Special Users,o=bigco.com
LogLevel = 2
UpperCaseName = 0
PasswordCase=original
SSL = 0
[Server]
s1=
[Server/s1]
Host=199.185.162.147
Port = 389
[Defaults]
TheUserProfile = Sample
[Request]
%User-Name = User-Name
[Response]
```

```
%Profile = TheUserProfile

[Search/DoLdapSearch]

Base = ou=Special Users, o=bigco.com

Scope = 2

Filter = uid=<dialup>

Attributes = AttrList

Timeout = 20

%DN = dn

[Attributes/AttrList]
```

If the [Response] section was empty, Steel-Belted Radius would pass the Bind results (accept or reject) directly to its client and no additional RADIUS attributes would be returned in the Access-Response.

BindName Authentication with Callback Number Returned

In the following example, requests are authenticated using Search. BindName and BindPassword values are supplied to permit a connection to the LDAP database. Return list attributes for authentication are listed in the [Response] section. In this example, the network access device needs a callback number to complete the connection. The value of the incoming DNIS attribute Calling-Station-ID is used to ensure that the callback number is the number from which the user's request originated.

Note: This example is incomplete; it omits the [Bootstrap] and [Settings] sections to save space.

```
[Server]

s1=

[Server/s1]

Host = 67.186.4.3

Port = 389

BindName=uid=admin,   ou=Administrators,   ou=TopologyManagement,

o=NetscapeRoot

BindPassword=ourlittlesecret

Search = DoLdapSearch

[Defaults]

SendThis = DidLDAPAuthSearch

[Request]

%UserName = dialup

Calling-Station-ID =thenumbertocall

[Search/DoLdapSearch]

Base = ou=Special Users, o=bigco.com
```

```
Scope = 2

Filter = uid=<dialup>

Attributes = AttrList

Timeout = 20

%DN = dn

[Attributes/AttrList]

dialuppassword

[Response]

%Password = dialuppassword

Reply-Message = SendThis

Ascend-Callback-No=thenumbertocall
```

LDAP Bind with Profile Based on Network Access Device

In the following example, requests are authenticated using Bind. Checklist and return list attributes for authentication are provided by referencing a profile entry in the Steel-Belted Radius database. The profile to be used depends on the specific network access device from which the user's request originates. Steel-Belted Radius retrieves the profile name by the LDAP database for an IP address that matches the address of the requesting RAS. If this search fails, a profile called limited is used. If a profile name is successfully retrieved from the LDAP database, but no profile by that name can be found in the Steel-Belted Radius database, authentication fails due to "lack of resources" and the user is rejected.

Note: This example is incomplete; it omits the [Bootstrap] section and many [Settings] entries to save space.

```
[Settings]

Bind=uid=<loginID>, ou=Special Users, o=bigco.com

Search = DoLdapSearch

[Server]

s1=

[Server/s1]

Host = 67.186.4.3

Port = 389

[Request]

%UserName = loginID

%NASAddress = deviceIP

[Defaults]

%Profile = limited

[Search/DoLdapSearch]
```

```
Base = ou=CommServers, o=bigco.com
Scope = 1
Filter = ipaddr=<deviceIP>
Attributes = AttrList
Timeout = 20
%DN = dn
[Attributes/AttrList]
profile
[Response]
%Profile = profile
```

Chapter 34

Displaying Statistics via Legacy SBR Administrator

The Statistics panel lets you display summary statistics for authentication, accounting, and proxy forwarding transactions. You can also use the Statistics panel to see how long Steel-Belted Radius has been running and to display a list of the users currently connected through a RAS or tunnel via legacy SBR administrator.

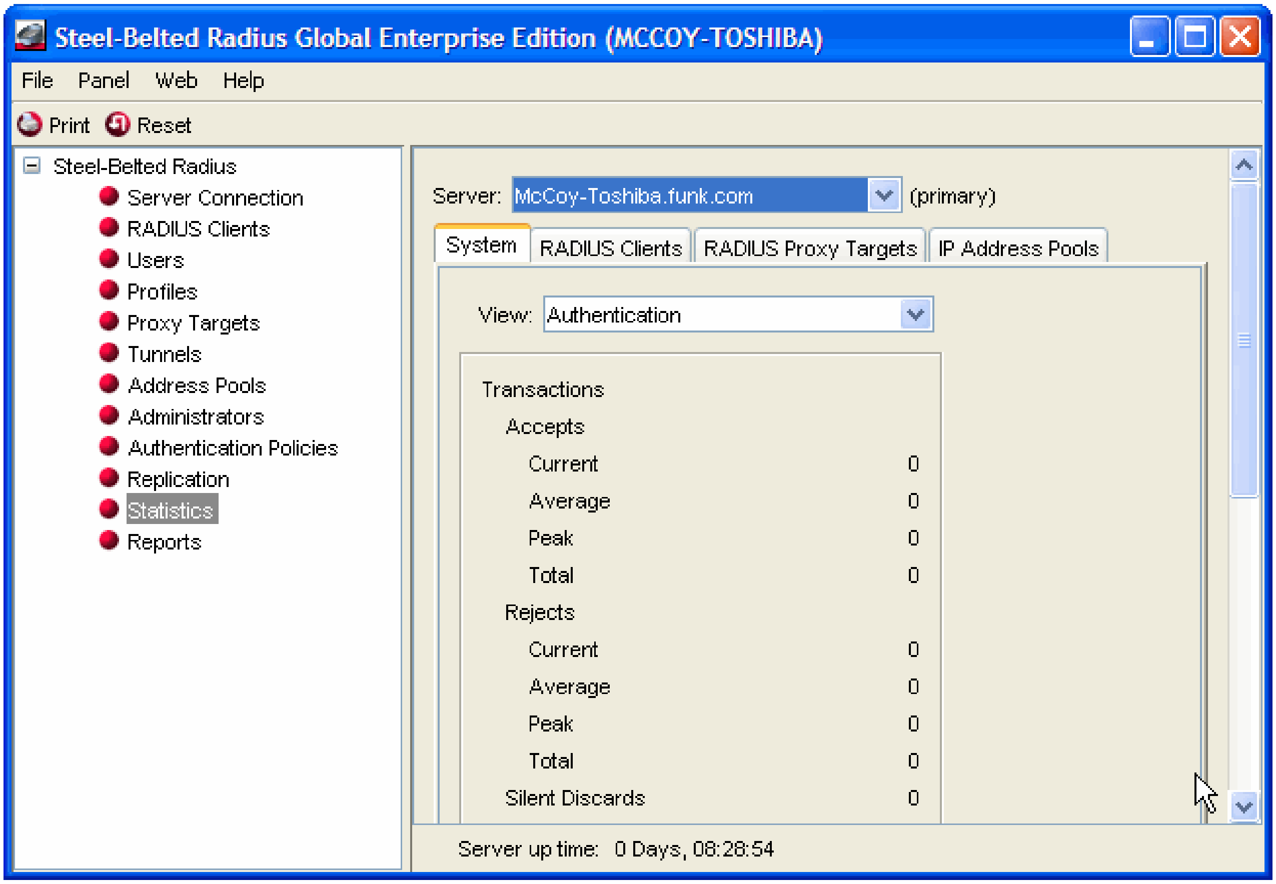
Note: Only the standard IETF RADIUS statistics are available from the Statistics panel.
To access Steel-Belted Radius extended statistics, you must use other utilities. See "Statistics Variables".

Displaying Authentication Statistics

Authentication statistics (Figure 213: Statistics Panel: Authentication Statistics) summarize the number of authentication acceptances and rejections, with summary totals for each type of rejection or retry.

To display authentication statistics, open the Statistics panel, click the System tab, pull down the View list, and chooseAuthentication.

Figure 213: Statistics Panel: Authentication Statistics



Server: McCoy-Toshiba.funk.com (primary)

View: Authentication

```
Transactions
  Accepts
    Current          0
    Average          0
    Peak             0
    Total            0
  Rejects
    Current          0
    Average          0
    Peak             0
    Total            0
  Silent Discards    0
```

Server up time: 0 Days, 08:28:54

Table 50 explains the authentication statistics fields and describes possible causes for authentication rejections.

Table 50: Authentication Statistics

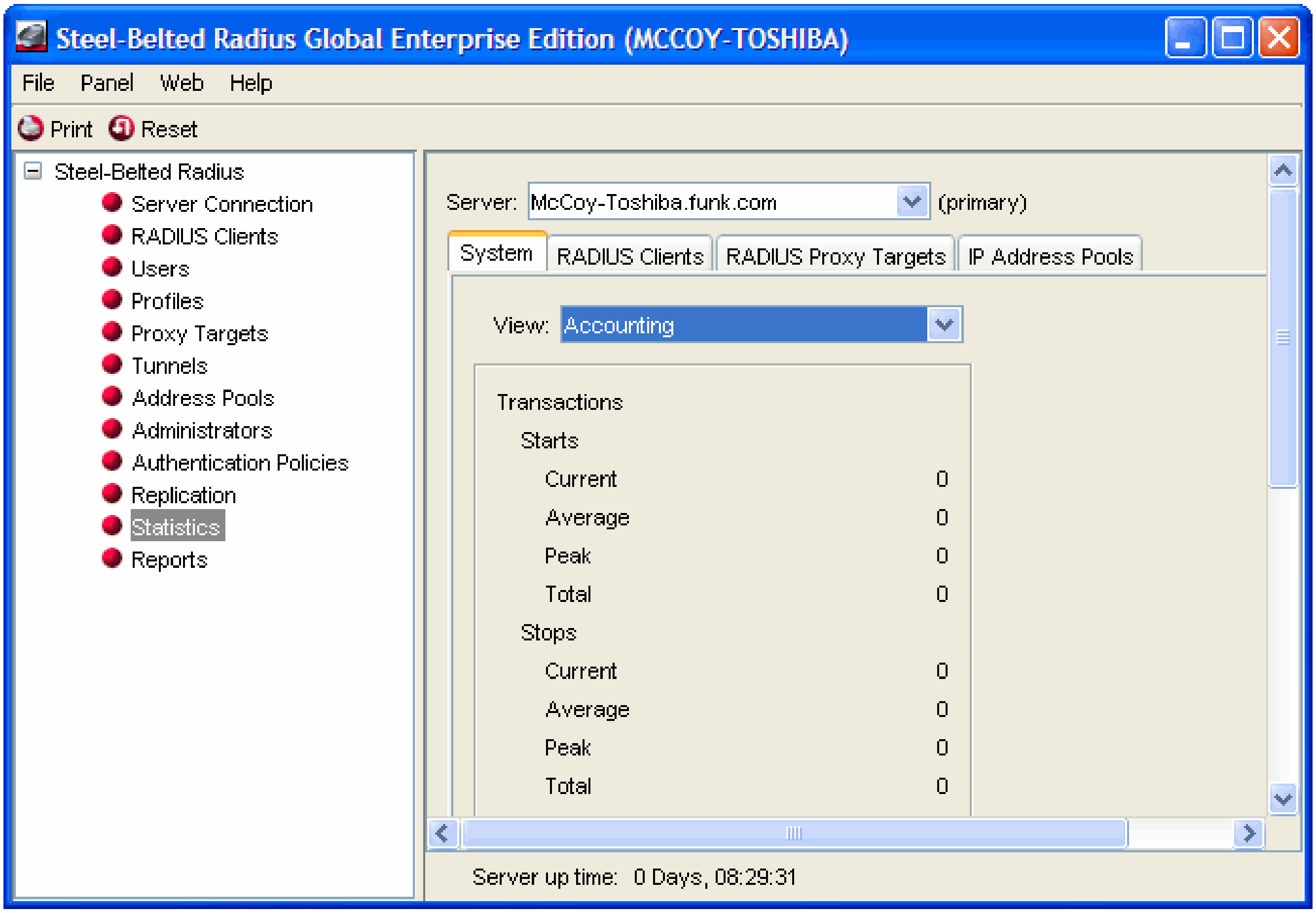
File Name	Function
Transactions	
Accepts	The current, average, and peak number of RADIUS transactions that resulted in an accept response.
Rejects	The current, average, and peak number of RADIUS transactions that resulted in a reject response. These are detailed in the Reject Details section.
Silent Discards	The number of requests in which the client could not be identified. This might occur if a RADIUS client entry cannot be found for a device with the name and/or IP address of a device requesting authentication services.
Total Transactions	The sum of the accept, reject, and silent discard totals.
Reject Details	
Dropped Packet	The number of RADIUS authentication packets dropped by Steel-Belted Radius because the server was flooded with more packets than it could handle.
Invalid Request	The number of invalid RADIUS requests made. A device is sending incorrectly formed packets to Steel-Belted Radius; either there is a configuration error or the device does not conform to the RADIUS standard.
Failed Authentication	The number of failed authentication requests, where the failure is due to invalid username or password. If all transactions are failing authentication, the problem might be that the shared secret entered into Steel- Belted Radius does not match the shared secret entered on the client device.
Failed on Checklist	The number of requests that were authenticated but failed to meet the checklist requirements.
Insufficient Resources	The number of rejects due to a server resource problem.
Proxy Failure	The number of rejects that had to be issued because Proxy forwarding to another RADIUS server failed.
Rejected by Proxy	The number of rejects due to receiving a reject response from a proxy RADIUS target server.
Retries Received	
Transactions Retried	The number of requests for which one or more duplicates was received.
Total Retry Packets	The number of duplicate packets received.
Challenges	The number of challenges received.

Displaying Accounting Statistics

Accounting statistics provide information such as the number of transaction starts and stops and the reasons for rejecting attempted transactions. The start and stop numbers rarely match, as many transactions can be in progress at any given time.

To display authentication statistics, open the Statistics panel, click the System tab, pull down the View list, and choose Accounting.

Figure 214: Statistics Panel: Accounting Statistics



Server: McCoy-Toshiba.funk.com (primary)

View: Accounting

Transactions
Starts
Current 0
Average 0
Peak 0
Total 0
Stops
Current 0
Average 0
Peak 0
Total 0

Server up time: 0 Days, 08:29:31

Table 51 describes the accounting statistics and describes possible causes for accounting errors.

Table 51: Accounting Statistics

Statistic	Meaning
Transactions	
Starts	The current, average, and peak number of transactions in which a dial-in connection was started following a successful authentication.
Stops	The current, average, and peak number of transactions in which a dial-in connection was terminated.
Ons	The number of Accounting-On messages received, indicating that a RADIUS client has restarted.
Offs	The number of Accounting-Off messages received, indicating that a RADIUS client has shut down.
Total	The sum of the start, stop, on and off totals.
Failure Details	

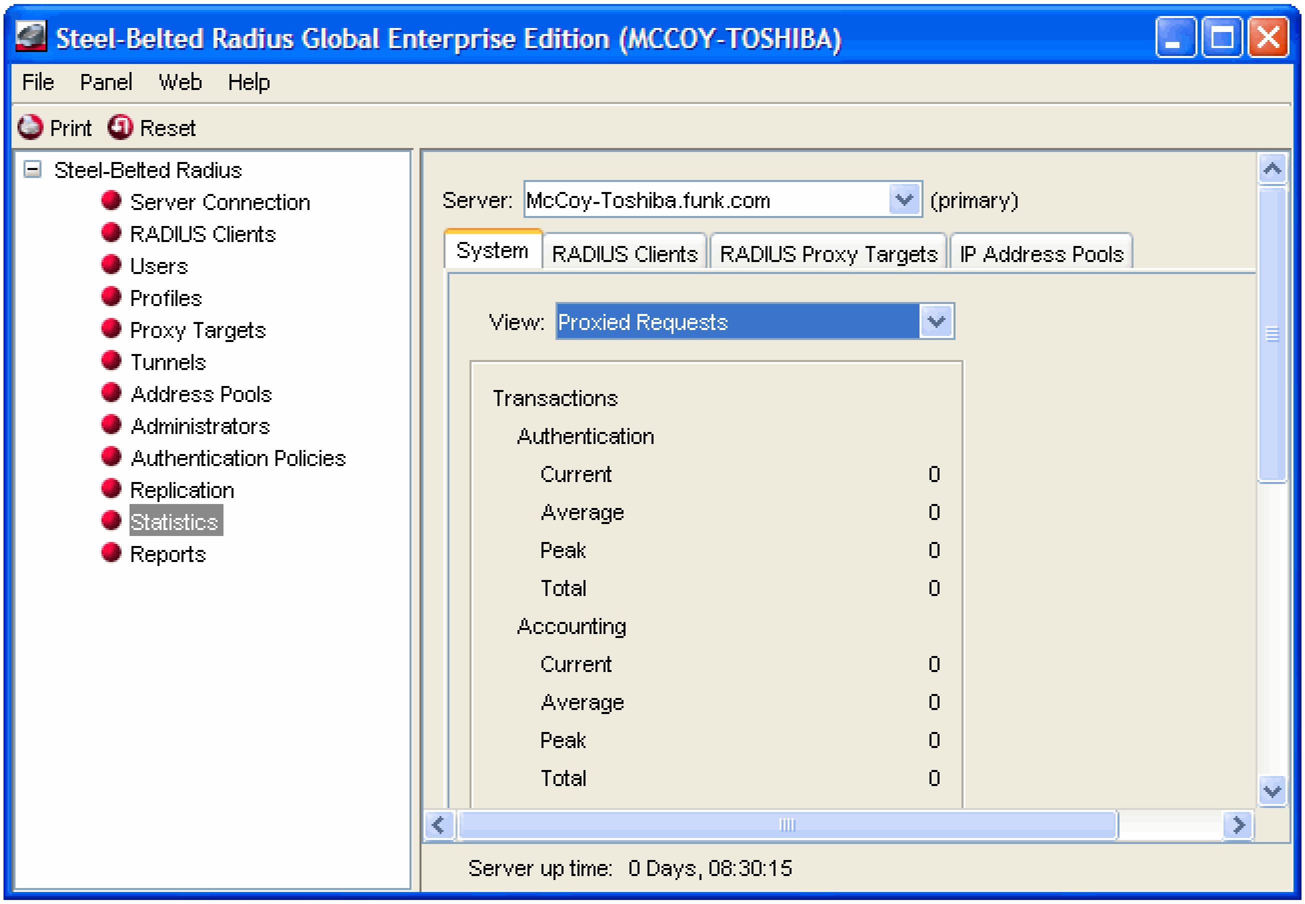
Statistic	Meaning
Dropped Packet	The number of RADIUS accounting packets dropped by Steel-Belted Radius because the server was flooded with more packets than it could handle.
Invalid Request	The number of invalid RADIUS requests made. A device is sending incorrectly formed packets to Steel-Belted Radius; either there is a configuration error or the device does not conform to the RADIUS standard.
Invalid Client	The number of requests in which the RADIUS client could not be identified. A device might be configured to use Steel-Belted Radius but no RADIUS client entry has been created with the name and/or IP address of the client; or the RADIUS client entry might be configured with an incorrect name or IP address; or some rogue device is attempting to compromise RADIUS security.
Invalid Shared Secret	The number of packets for which an incorrect digital signature was received. The shared secret does not match between Steel-Belted Radius and the client device; or some rogue device is attempting to compromise RADIUS security.
Insufficient Resources	The number of rejects due to a server resource problem.
Proxy Failure	The number of times that proxy RADIUS forwarding failed.
Retries Received	
Transactions Retried	The number of requests for which one or more duplicates was received.
Total Retry Packets	The number of duplicate packets received.
Interim Requests	The number of interim accounting packets received.

Displaying Proxied Request Statistics

Proxied request statistics provide information such as the number of proxy authentication or accounting requests and the reasons for any transaction failures that occur.

To display proxied request statistics, open the Statistics panel, click the **System** tab, pull down the **View** list, and choose **Proxied Requests**.

Figure 215: Statistics Panel: Proxied Request Statistics



Steel-Belted Radius Global Enterprise Edition (MCCOY-TOSHIBA)

Server: McCoy-Toshiba.funk.com (primary)

View: Proxied Requests

```
Transactions
  Authentication
    Current      0
    Average      0
    Peak         0
    Total        0
  Accounting
    Current      0
    Average      0
    Peak         0
    Total        0
```

Server up time: 0 Days, 08:30:15

Table 52 describes the proxy request statistics, with possible interpretations in italics.

Table 52: Proxy Statistics

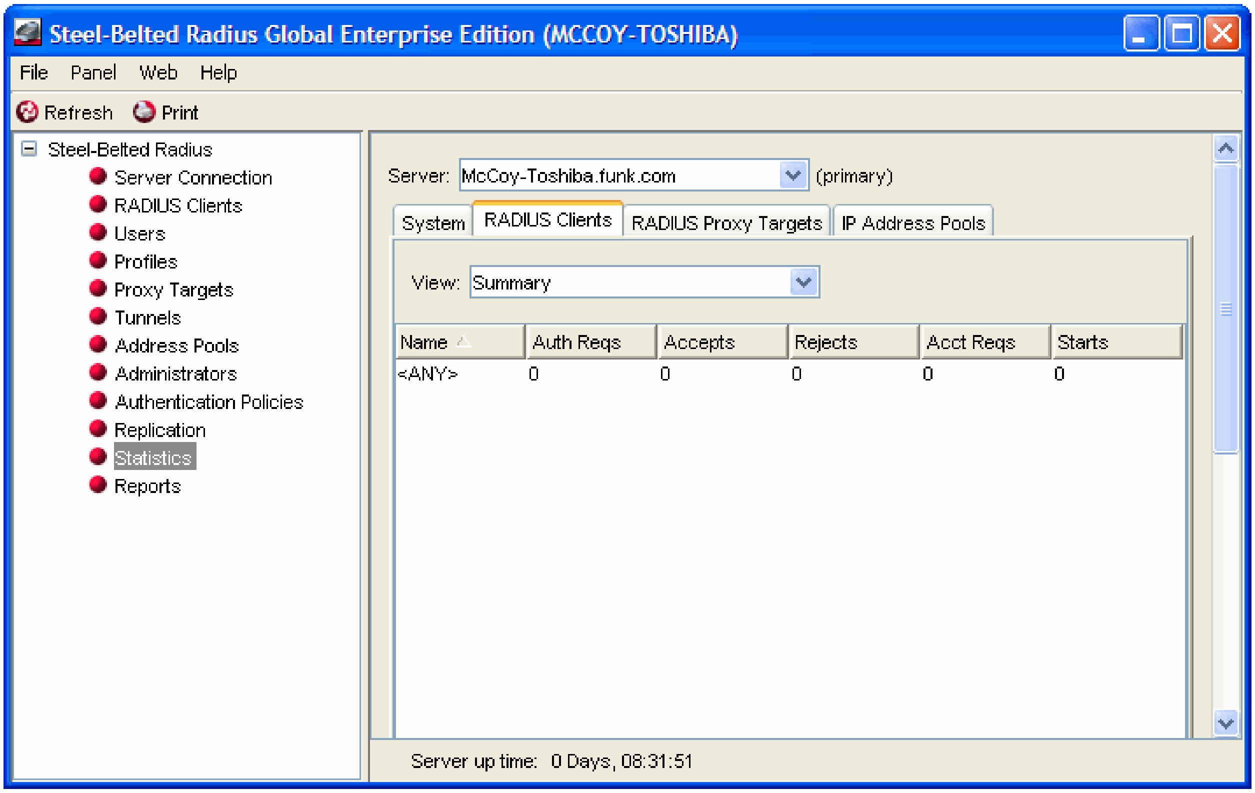
Proxy Statistic	Meaning
Transactions	
Authentication	The number of authentication transactions between the proxy and target RADIUS servers.
Accounting	The number of accounting transactions between the proxy and target RADIUS servers.
Total Transactions	The sum of the authentication and accounting transaction totals.
Failure Details	
Timed Out	The number of RADIUS transactions that timed out. This means that after all retry attempts were made, the transaction still timed out.
Invalid Response	The number of invalid RADIUS responses received. A target is sending incorrectly formed packets to Steel-Belted Radius; there is a configuration error, the target RADIUS server does not conform to the RADIUS standard, or Steel-Belted Radius did not receive a proxy state echo in the received packet.
Invalid Shared Secret	The number of packets for which an incorrect digital signature was received. The shared secret does not match between Steel-Belted Radius and the target; or some unauthorized rogue device is attempting to compromise RADIUS security.

Proxy Statistic	Meaning
Insufficient Resources	The number of rejects due to a server resource problem.
Retries Sent	
Transactions Retried	The number of requests for which one or more retried transmissions was performed.
Total Retry Packets	The number of duplicate packets received.

Displaying RADIUS Client Statistics

RADIUS client statistics provide information about the number of authentication and accounting requests by client.

Figure 216: Statistics Panel: RADIUS Clients Tab



Server: McCoy-Toshiba.funk.com (primary)

View: Summary

Name	Auth Reqs	Accepts	Rejects	Acct Reqs	Starts
<ANY>	0	0	0	0	0

Server up time: 0 Days, 08:31:51

To display statistics for RADIUS clients:

1. Open the Statistics panel and click the **RADIUS Clients** tab.
2. Use the **View** list to display the type of statistics you want to display.
    - **Summary—**Displays the number of authentication requests, Access-Accepts, and Accept-Reject messages and the total number of accounting requests, starts, and stops for each RADIUS client.
    - **Authentication Request Details—**Displays the number of duplicate messages, challenges, messages containing invalid authentication information, bad authentication requests, bad types, and dropped requests for each RADIUS client.
    - **Accounting Request Types—**Displays the number of accounting start messages, accounting stop messages, interim messages, Accounting-On messages, Accounting-Off messages, and acknowledgement messages sent for each RADIUS client.

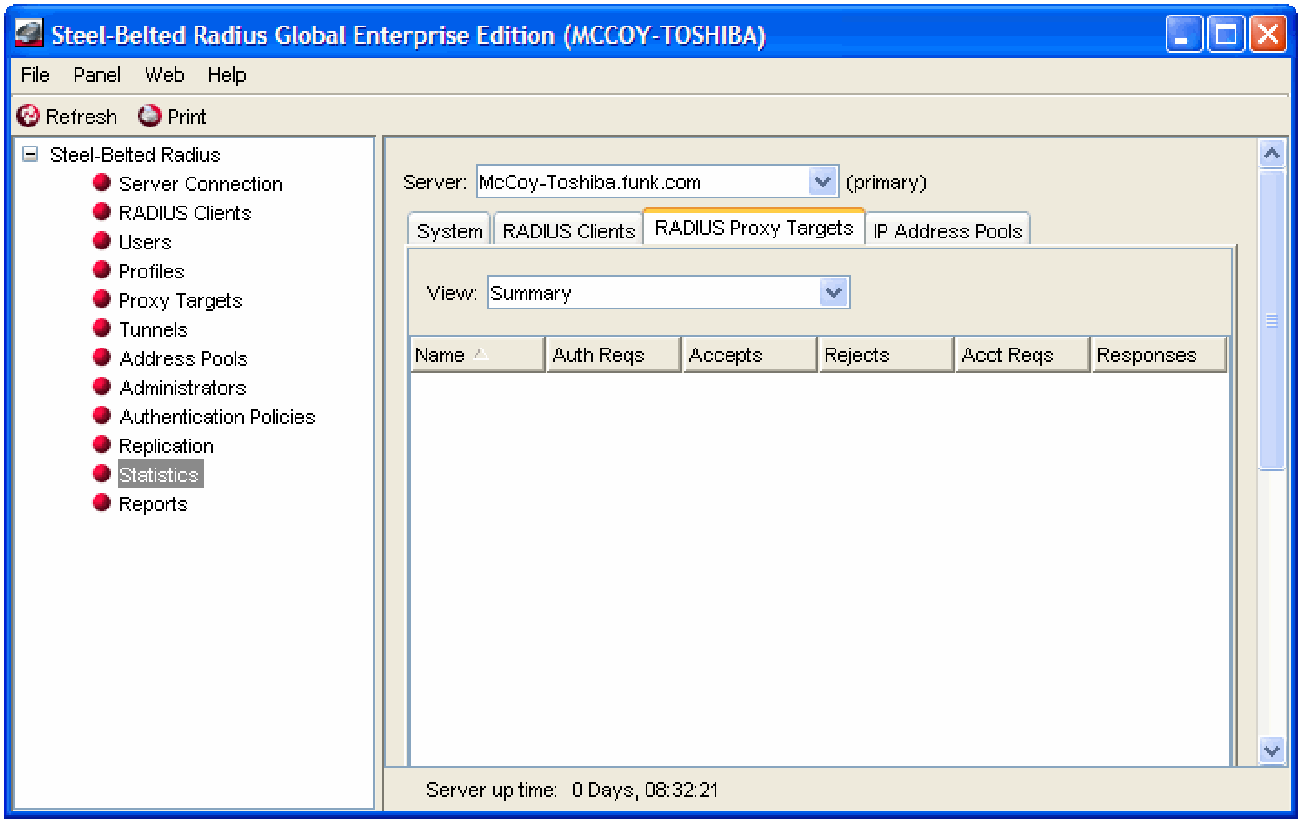
- **Accounting Request Diagnostics—**Displays the number of duplicate messages, messages with invalid secrets, malformed messages, messages with incorrect types, ignored messages, and dropped requests for each RADIUS client.

3. Optionally, sort the messages by clicking a column header.

Displaying RADIUS Proxy Targets Statistics

RADIUS proxy target statistics provide information about the number of authentication and accounting transactions associated with each proxy target.

Figure 217: Statistics Tab: RADIUS Proxy Targets Tab



Server: McCoy-Toshiba.funk.com (primary)

View: Summary

Server up time: 0 Days, 08:32:21

To display statistics for RADIUS proxy targets:

1. Open the Statistics panel and click the RADIUS Proxy Targets tab.

2. Use the View list to display the type of statistics you want to display.

   - **Summary—**Displays the number of authentication requests, accepts and reject messages, and the number of accounting requests and responses for each RADIUS proxy target.

   - **Authentication Request Details—**Displays the number of outstanding messages, retransmitted messages, and challenges, along with the most recent response time for the proxy target.

   - **Authentication Request Diagnostics—**Displays the number of timeouts, invalid secrets, incorrect requests, requests with invalid types, and dropped messages for each proxy target.

   - **Accounting Request Types—**Displays the number of outstanding messages and retransmitted messages, along with the most recent response time for the proxy target.

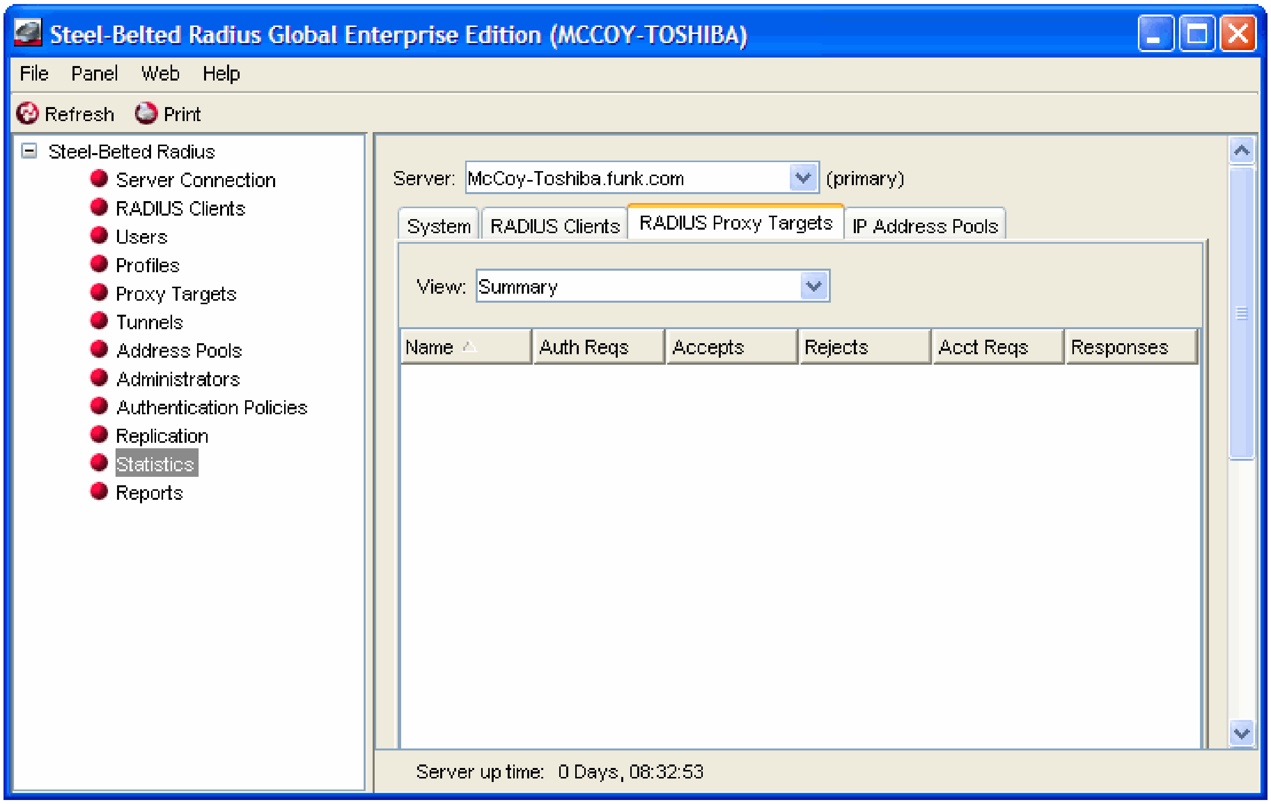
- **Accounting Request Diagnostics—**Displays the number of timeouts, invalid secrets, incorrect requests, requests with invalid types, and dropped messages for each proxy target.

3. Optionally, sort the messages by clicking a column header.

Displaying IP Address Pool Statistics

IP address pool statistics provide a summary of the number of addresses allocated from each IPv4 address pool and how many addresses remain available.

Figure 218: Statistics Panel: IP Address Pools Tab



Server: McCoy-Toshiba.funk.com (primary)

View: Summary

Server up time: 0 Days, 08:32:53

Chapter 35

Displaying Statistics via WebGUI

The Statistics page lets you display summary statistics for authentication, accounting, and proxy forwarding transactions. You can also use the Statistics page to see how long Steel-Belted Radius has been running and to display a list of the users currently connected through a RAS or tunnel via WebGUI.

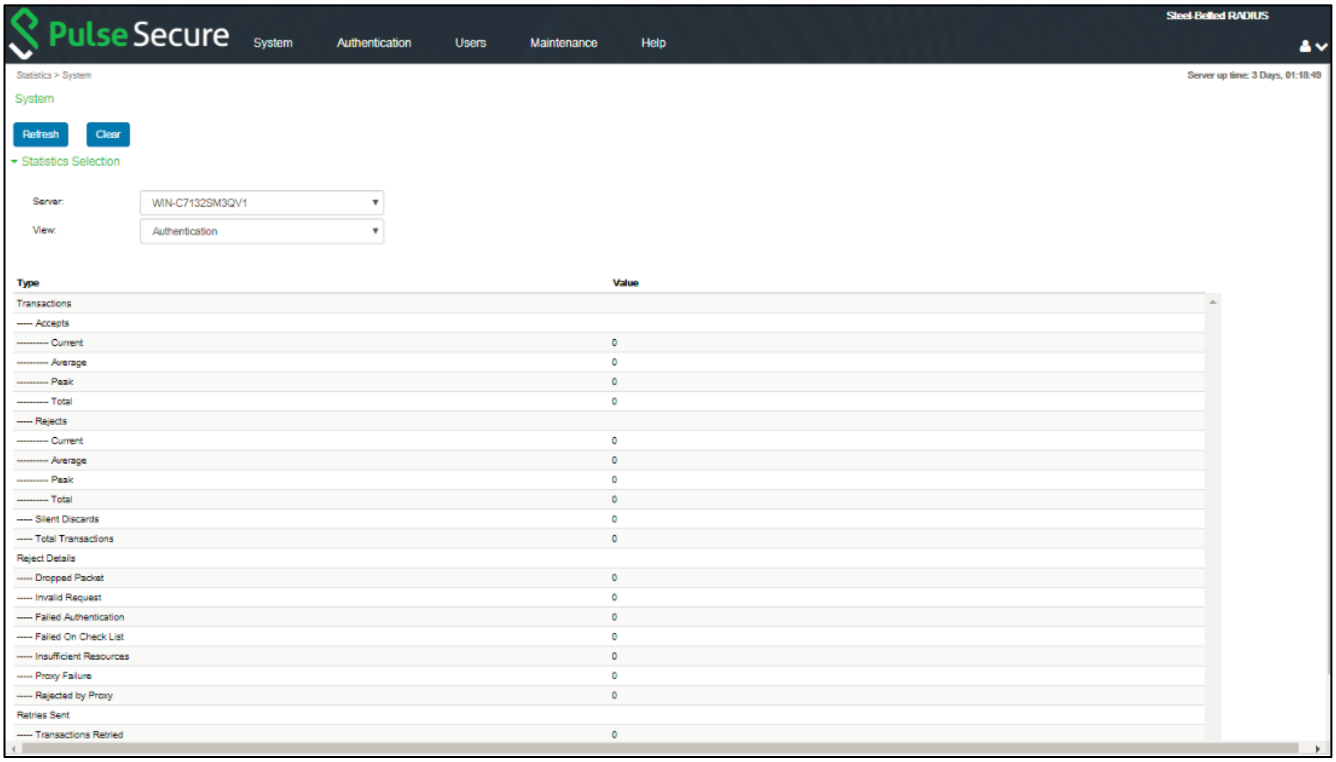
Note: Only the standard IETF RADIUS statistics are available from the Statistics page. To access Steel-Belted Radius extended statistics, you must use other utilities. See "Statistics Variables".

Displaying Authentication Statistics

Authentication statistics (Figure 219: Statistics Page: Authentication Statistics) summarize the number of authentication acceptances and rejections, with summary totals for each type of rejection or retry.

To display authentication statistics, open the Statistics page, click the System > Statistics > System, pull down the View list, and chooseAuthentication.

Figure 219: Statistics Page: Authentication Statistics



Server up time: 3 Days, 01:18:49

Server: WIN-C7132SM3QV1

View: Authentication

Type	Value
Transactions	
----- Accepts	
---------- Current	0
---------- Average	0
---------- Peak	0
---------- Total	0
----- Rejects	
---------- Current	0
---------- Average	0
---------- Peak	0
---------- Total	0
----- Silent Discards	0
----- Total Transactions	0
Reject Details	
----- Dropped Packet	0
----- Invalid Request	0
----- Failed Authentication	0
----- Failed On Check List	0
----- Insufficient Resources	0
----- Proxy Failure	0
----- Rejected by Proxy	0
Retries Sent	
----- Transactions Retried	0

Table 50 explains the authentication statistics fields and describes possible causes for authentication rejections.

Table 50: Authentication Statistics

File Name	Function
Transactions	
Accepts	The current, average, and peak number of RADIUS transactions that resulted in an accept response.

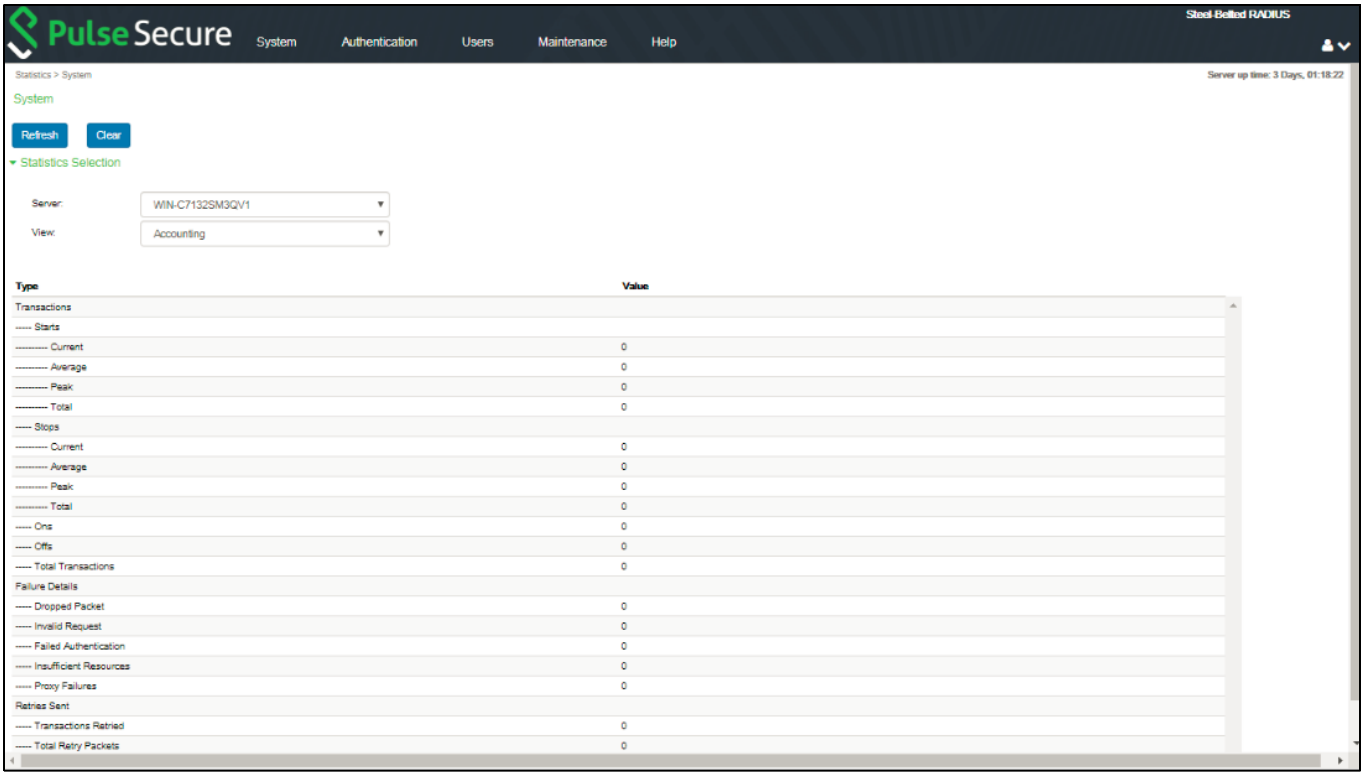
File Name	Function
Rejects	The current, average, and peak number of RADIUS transactions that resulted in a reject response. These are detailed in the Reject Details section.
Silent Discards	The number of requests in which the client could not be identified. This might occur if a RADIUS client entry cannot be found for a device with the name and/or IP address of a device requesting authentication services.
Total Transactions	The sum of the accept, reject, and silent discard totals.
Reject Details	
Dropped Packet	The number of RADIUS authentication packets dropped by Steel-Belted Radius because the server was flooded with more packets than it could handle.
Invalid Request	The number of invalid RADIUS requests made. A device is sending incorrectly formed packets to Steel-Belted Radius; either there is a configuration error or the device does not conform to the RADIUS standard.
Failed Authentication	The number of failed authentication requests, where the failure is due to invalid username or password. If all transactions are failing authentication, the problem might be that the shared secret entered into Steel- Belted Radius does not match the shared secret entered on the client device.
Failed on Checklist	The number of requests that were authenticated but failed to meet the checklist requirements.
Insufficient Resources	The number of rejects due to a server resource problem.
Proxy Failure	The number of rejects that had to be issued because Proxy forwarding to another RADIUS server failed.
Rejected by Proxy	The number of rejects due to receiving a reject response from a proxy RADIUS target server.
Retries Received	
Transactions Retried	The number of requests for which one or more duplicates was received.
Total Retry Packets	The number of duplicate packets received.
Challenges	The number of challenges received.

Displaying Accounting Statistics

Accounting statistics provide information such as the number of transaction starts and stops and the reasons for rejecting attempted transactions. The start and stop numbers rarely match, as many transactions can be in progress at any given time.

To display accounting statistics, choose System > Statistics > System, pull down the View list, and choose Accounting.

Figure 220: Statistics Page: Accounting Statistics



Server up time: 3 Days, 01:18:22

Server: WIN-C7132SM3QV1

View: Accounting

Type	Value
Transactions	
----- Starts	
---------- Current	0
---------- Average	0
---------- Peak	0
---------- Total	0
----- Stops	
---------- Current	0
---------- Average	0
---------- Peak	0
---------- Total	0
----- Ons	0
----- Offs	0
----- Total Transactions	0
Failure Details	
----- Dropped Packet	0
----- Invalid Request	0
----- Failed Authentication	0
----- Insufficient Resources	0
----- Proxy Failures	0
Retries Sent	
----- Transactions Retried	0
----- Total Retry Packets	0

Table 51 describes the accounting statistics and describes possible causes for accounting errors.

Table 51: Accounting Statistics

Statistic	Meaning
Transactions	
Starts	The current, average, and peak number of transactions in which a dial-in connection was started following a successful authentication.
Stops	The current, average, and peak number of transactions in which a dial-in connection was terminated.
Ons	The number of Accounting-On messages received, indicating that a RADIUS client has restarted.
Offs	The number of Accounting-Off messages received, indicating that a RADIUS client has shut down.
Total	The sum of the start, stop, on and off totals.
Failure Details	
Dropped Packet	The number of RADIUS accounting packets dropped by Steel-Belted Radius because the server was flooded with more packets than it could handle.

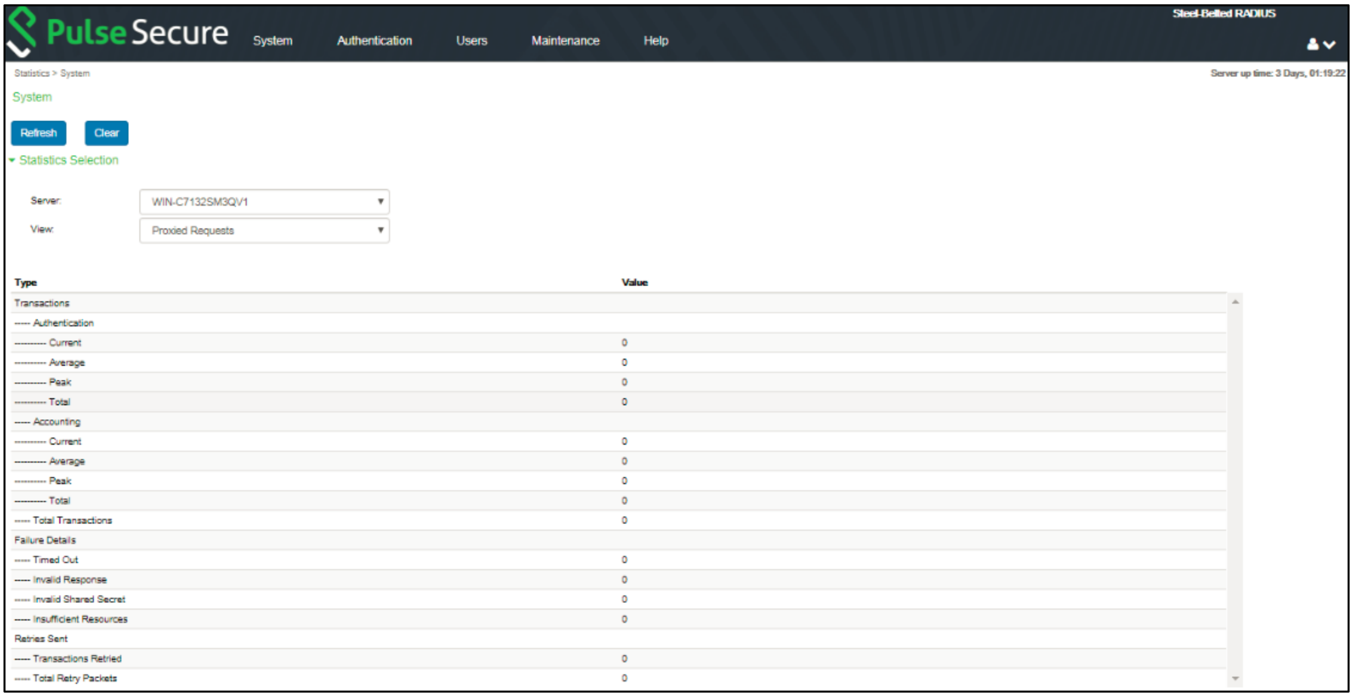
Statistic	Meaning
Invalid Request	The number of invalid RADIUS requests made. A device is sending incorrectly formed packets to Steel-Belted Radius; either there is a configuration error or the device does not conform to the RADIUS standard.
Invalid Client	The number of requests in which the RADIUS client could not be identified. A device might be configured to use Steel-Belted Radius but no RADIUS client entry has been created with the name and/or IP address of the client; or the RADIUS client entry might be configured with an incorrect name or IP address; or some rogue device is attempting to compromise RADIUS security.
Invalid Shared Secret	The number of packets for which an incorrect digital signature was received. The shared secret does not match between Steel-Belted Radius and the client device; or some rogue device is attempting to compromise RADIUS security.
Insufficient Resources	The number of rejects due to a server resource problem.
Proxy Failure	The number of times that proxy RADIUS forwarding failed.
Retries Received	
Transactions Retried	The number of requests for which one or more duplicates was received.
Total Retry Packets	The number of duplicate packets received.
Interim Requests	The number of interim accounting packets received.

Displaying Proxied Request Statistics

Proxied request statistics provide information such as the number of proxy authentication or accounting requests and the reasons for any transaction failures that occur.

To display proxied request statistics, choose **System > Statistics > System**, pull down the **View** list, and choose **Proxied Requests**.

Figure 221: Statistics Page: Proxied Request Statistics



Server up time: 3 Days, 01:19:22

Server: WIN-C7132SM3QV1

View: Proxied Requests

Type	Value
Transactions	
----- Authentication	
---------- Current	0
---------- Average	0
---------- Peak	0
---------- Total	0
----- Accounting	
---------- Current	0
---------- Average	0
---------- Peak	0
---------- Total	0
----- Total Transactions	0
Failure Details	
----- Timed Out	0
----- Invalid Response	0
----- Invalid Shared Secret	0
----- Insufficient Resources	0
Retries Sent	
----- Transactions Retried	0
----- Total Retry Packets	0

Table 52 describes the proxy request statistics, with possible interpretations in italics.

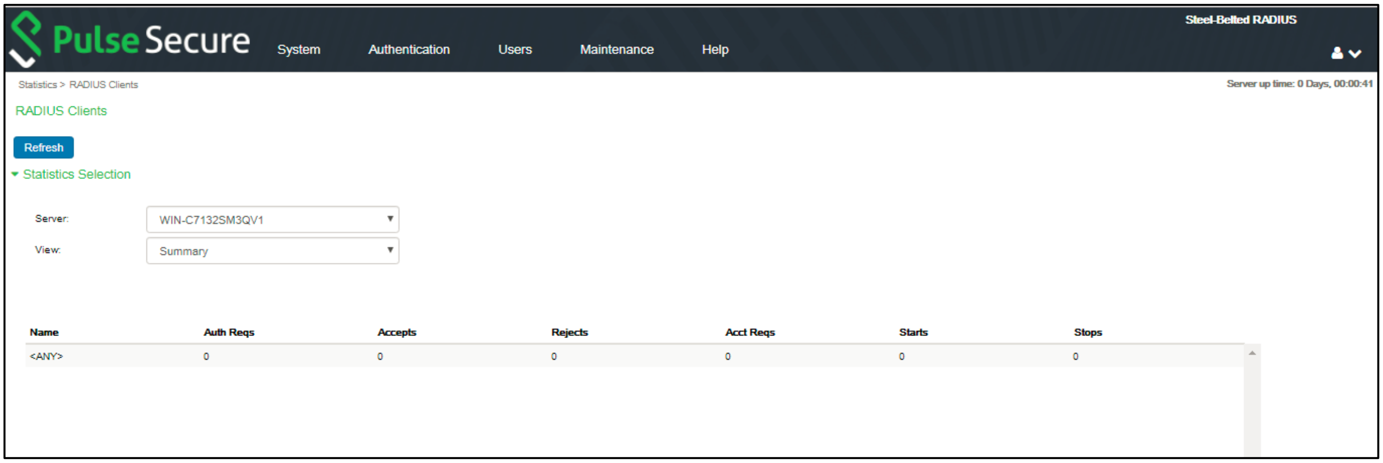
Table 52: Proxy Statistics

Proxy Statistic	Meaning
Transactions	
Authentication	The number of authentication transactions between the proxy and target RADIUS servers.
Accounting	The number of accounting transactions between the proxy and target RADIUS servers.
Total Transactions	The sum of the authentication and accounting transaction totals.
Failure Details	
Timed Out	The number of RADIUS transactions that timed out. This means that after all retry attempts were made, the transaction still timed out.
Invalid Response	The number of invalid RADIUS responses received. A target is sending incorrectly formed packets to Steel-Belted Radius; there is a configuration error, the target RADIUS server does not conform to the RADIUS standard, or Steel-Belted Radius did not receive a proxy state echo in the received packet.
Invalid Shared Secret	The number of packets for which an incorrect digital signature was received. The shared secret does not match between Steel-Belted Radius and the target; or some unauthorized rogue device is attempting to compromise RADIUS security.
Insufficient Resources	The number of rejects due to a server resource problem.
Retries Sent	
Transactions Retried	The number of requests for which one or more retried transmissions was performed.
Total Retry Packets	The number of duplicate packets received.

Displaying RADIUS Client Statistics

RADIUS client statistics provide information about the number of authentication and accounting requests by client.

Figure 222: Statistics Page: RADIUS Clients Tab



Server: WIN-C7132SM3QV1

View: Summary

Name	Auth Reqs	Accepts	Rejects	Acct Reqs	Starts	Stops
<ANY>	0	0	0	0	0	0

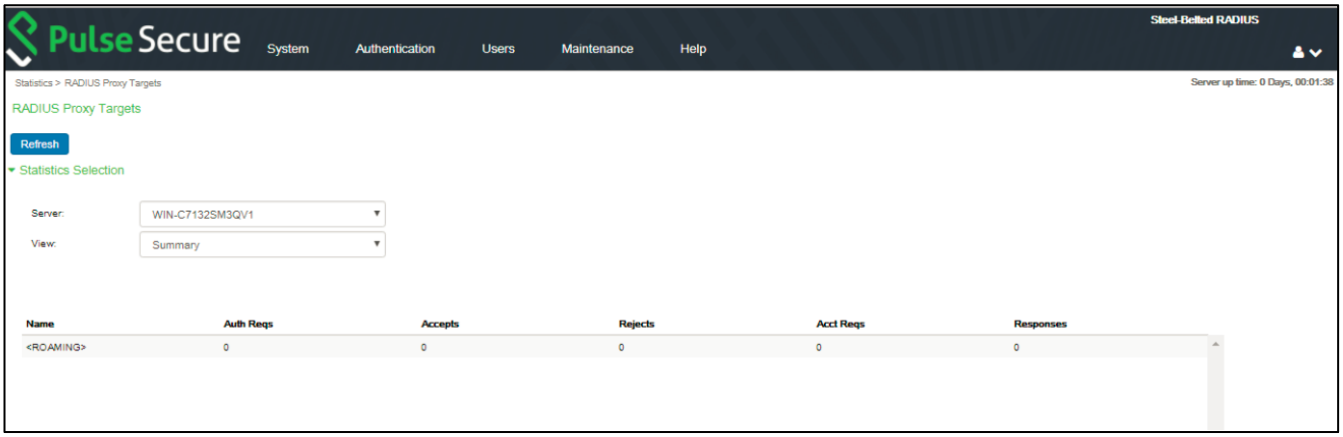
To display statistics for RADIUS clients:

1. Choose System > Statistics > Radius Clients to open the Statistics page.
2. Use the **View** list to display the type of statistics you want to display.
   - **Summary—**Displays the number of authentication requests, Access-Accepts, and Accept-Reject messages and the total number of accounting requests, starts, and stops for each RADIUS client.
   - **Authentication Request Details—**Displays the number of duplicate messages, challenges, messages containing invalid authentication information, bad authentication requests, bad types, and dropped requests for each RADIUS client.
   - **Accounting Request Types—**Displays the number of accounting start messages, accounting stop messages, interim messages, Accounting-On messages, Accounting-Off messages, and acknowledgement messages sent for each RADIUS client.
   - **Accounting Request Diagnostics—**Displays the number of duplicate messages, messages with invalid secrets, malformed messages, messages with incorrect types, ignored messages, and dropped requests for each RADIUS client.
3. Optionally, sort the messages by clicking a column header.

Displaying RADIUS Proxy Targets Statistics

RADIUS proxy target statistics provide information about the number of authentication and accounting transactions associated with each proxy target.

Figure 223: Statistics Tab: RADIUS Proxy Targets Tab



Server: WIN-C7132SM3QV1

View: Summary

Name	Auth Reqs	Accepts	Rejects	Acct Reqs	Responses
<ROAMING>	0	0	0	0	0

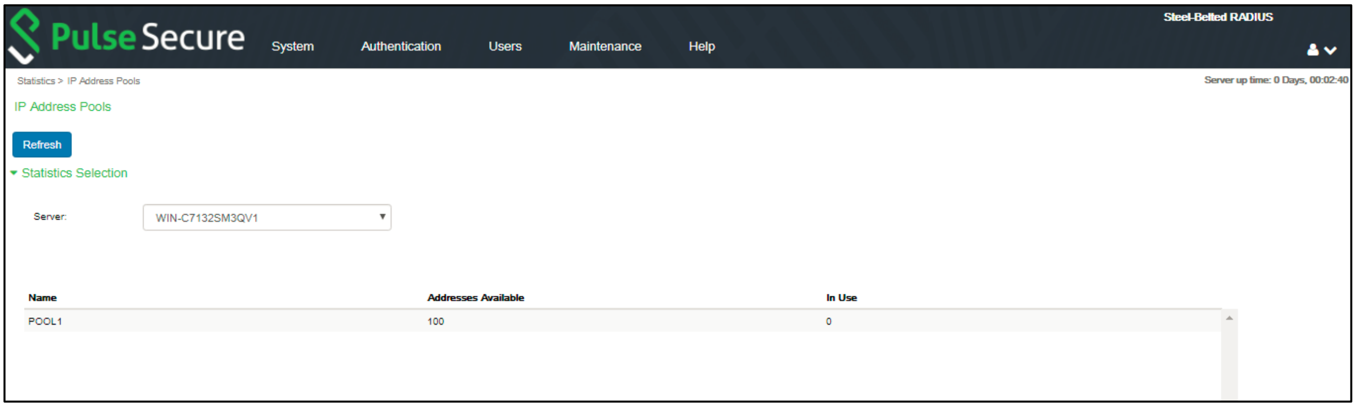
To display statistics for RADIUS proxy targets:

1. Choose System > Statistics > Radius Proxy Targets to open the Statistics pange.
2. Use the View list to display the type of statistics you want to display.
   - **Summary—**Displays the number of authentication requests, accepts and reject messages, and the number of accounting requests and responses for each RADIUS proxy target.
   - **Authentication Request Details—**Displays the number of outstanding messages, retransmitted messages, and challenges, along with the most recent response time for the proxy target.
   - **Authentication Request Diagnostics—**Displays the number of timeouts, invalid secrets, incorrect requests, requests with invalid types, and dropped messages for each proxy target.
   - **Accounting Request Types—**Displays the number of outstanding messages and retransmitted messages, along with the most recent response time for the proxy target.
   - **Accounting Request Diagnostics—**Displays the number of timeouts, invalid secrets, incorrect requests, requests with invalid types, and dropped messages for each proxy target.
3. Optionally, sort the messages by clicking a column header.

Displaying IP Address Pool Statistics

IP address pool statistics provide a summary of the number of addresses allocated from each IPv4 address pool and how many addresses remain available.

Figure 224: Statistics Page: IP Address Pools Tab



Server: WIN-C7132SM3QV1

Name	Addresses Available	In Use
POOL1	100	0

Chapter 36
Logging and Reporting via Legacy SBR Administrator

This chapter describes how to set up and use logging and reporting functions in Steel-Belted Radius via legacy SBR administrator.

Logging Files

The following files establish settings for logging and reporting. For more information about these files, refer to the *Steel-Belted Radius Reference Guide*.

Table 53: Logging and Reporting Files

File Name	Function
account.ini	Controls how RADIUS accounting attributes are logged.
authlog.ini	Controls how RADIUS authentication requests are logged by Steel-Belted Radius.
authReport.ini	Controls what authentication logs Steel-Belted Radius generates.
authReportAccept.ini	Controls options for the acceptance authentication log file.
authReportBadSharedSecret.ini	Controls options for the invalid shared secret authentication log file.
authReportReject.ini	Controls options for the rejection authentication log file.
authReportUnknownClient.ini	Controls options for the unknown client authentication log file.
events.ini	Controls dilutions and thresholds for Steel-Belted Radius events used to communicate failures, warnings, and other information.
radius.inic	Controls (among other things) the types of messages Steel-Belted Radius records in the server log file and the location of the log directory.

Displaying the Current Sessions List

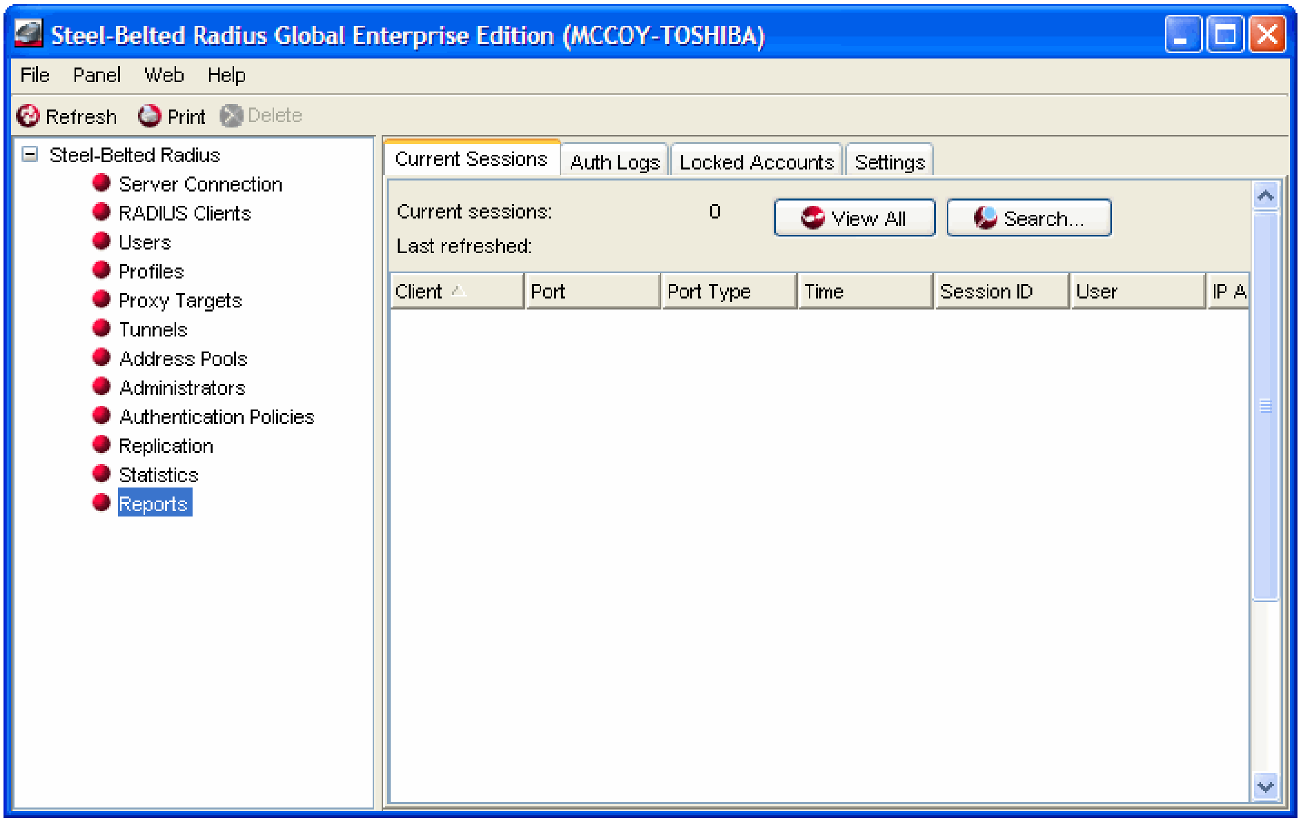
Steel-Belted Radius tracks the status of the user connections that it authenticates. To obtain a real-time snapshot of currently active connections, display the Current Sessions list. Because the Current Sessions list is based on RADIUS accounting data, the list is accurate only if all of your network access devices are configured to support RADIUS accounting.

Each server has its own Current Sessions display. Therefore, when you view this display, it typically reflects only the activity on the Steel-Belted Radius server to which you are connected. The Current Users display on a specific server reflects the activity across your entire RADIUS configuration only if (1) all clients in your configuration support RADIUS accounting, and (2) all clients are configured to send accounting messages to the server you are viewing.

Note: Steel-Belted Radius maintains the Current User list on disk. The information is preserved if you unload and reload the server.

To display the Current Sessions list, open the Reports panel and click the Current Sessions tab.

Figure 225: Reports Panel: Current Sessions List



Current sessions: 0

Last refreshed:

Table 54 describes the fields for each active session in the Current Sessions list.

Table 54: Sessions List Fields

Field	Meaning
Client	Client The identifier for the RADIUS client, which is typically the name or IP address of the device.
Port	The UDP port number on the RADIUS client that has been assigned to the connection. To determine slot number of the physical port on the RADIUS client, consult the device documentation.
Port Type	Describes how the port is used or configured.
Time	Identifies the date and time at which the connection was opened.
Session ID	Identifies the session key, which is a number generated by the RADIUS client.
User Name	Displays the name of the authenticated user. • If the user is local (native), the field shows only the username, in the form username. • If the user is non-local, the field shows the remote system name as well as the username, in the form \\systemname\username.

Field	Meaning
	- If the user is associated with a specific tunnel, the field shows the tunnel name as well as the username, in the form \\tunnelname\username.
IP Address	Identifies the IP address that was assigned to the user from an IP address pool. This field will be blank if a static IP address was assigned.

Note: For tunnel connections, if Steel-Belted Radius was used to authenticate both the user and the tunnel, then two entries are displayed in the Current Sessions panel: one entry for the authenticated user, and one for the authenticated tunnel.

Searching the Current Sessions List

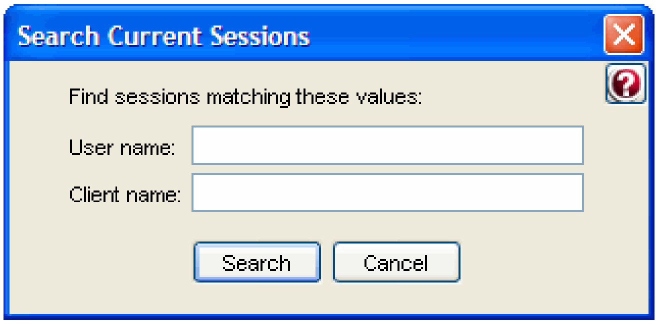
You can search the Current Sessions list to display only those sessions that match user name or client name criteria.

To search the Current Sessions list:

1. Open the Reports panel and click the Current Sessions tab.
2. Click the Search button.

   The Search Current Sessions dialog opens.

Figure 226: Search Current Sessions Dialog



Search Current Sessions

Find sessions matching these values:

User name:

Client name:

3. Enter the search criteria you want to use.
   - To search for a specific username, enter it in the User name field.
   - To search for a specific RADIUS client, enter a client name in the Client name field.
4. Click Search.

Deleting Entries from the Sessions List

Normally, the system maintains the information in the Sessions list based on accounting information received from the RADIUS client. However, a user who has logged off may still be identified as active in the Current Sessions list if communications between the RADIUS client and Steel-Belted Radius fail or if either the RADIUS client or Steel-Belted Radius is taken down for a period of time.

In most cases, Steel-Belted Radius can correct such anomalies itself. For example, if a new user dials in to the same port on the same RADIUS client, Steel-Belted Radius infers that the previous user must have disconnected and removes the entry.

You can manually correct the Sessions list by highlighting any entry and clicking Delete. This removes the user from the list and decrements the user's connection count (if it is being tracked) by one. Any pooled IP or IPX address assigned to the deleted user is returned to the appropriate pool.

Displaying the Authentication Log Files

The Auth Logs tab (Figure 235: Reports Page: Auth Logs Tab) on the Reports panel lets you enable and display the following authentication log files:

- Successful request authentication log file—The successful request authentication log file identifies the authentication requests that were approved by Steel-Belted Radius.
- Invalid shared secrets authentication log file—The invalid shared secrets authentication log file identifies the authentication requests that failed because a known RADIUS client supplied an incorrect shared secret. This condition is detectable only if the authentication request contains a Message-Authenticator attribute, which is required if credentials are of an EAP type but optional if credentials are PAP, CHAP, or MS-CHAP.
- Failed request authentication log file—The failed request authentication log file identifies the authentication requests that were rejected because the user supplied incorrect credentials.
- Unknown client request authentication log file—The unknown client request authentication log file identifies authentication requests received from unknown RADIUS clients.

File Permissions for Log Files (Linux)

When you run Steel-Belted Radius on a Linux server, you can specify the users who are authorized to read or edit important files, such as authentication and accounting log files. For example, you can specify that system administrators who install and configure Steel-Belted Radius have read/write access for system log files and that network operators who monitor Steel-Belted Radius have read-only (or no) access for system log files.

Security Groups and Permissions

Each file and directory on a Linux server has three security groups associated with it:

- The Owner identifies the person who created or owns the file.
- The Group security group identifies the set of users who are members of the group or groups to which the file Owner belongs. Group members can exercise special privileges with respect to that file. A user can belong to more than one group.
- The Other security group consists of the set of all users who do not belong to Owner or Group.

Each security group has three flags that control what privileges that group can exercise with respect to the file or directory:

- The Read flag (r) determines whether the file can be read. The Read flag has an octal value of 4.
- The Write flag (w) determines whether the security group can create, modify, or delete the file. The Write flag has an octal value of 2.
- The Execute flag (x) determines whether the security group can run a script or executable file. The Execute flag has an octal value of 1.

For example, a file owner might have rwx permission for a file, which indicates the file owner has read/write/execute access to the file. Similarly, Other might have r-- permission (where - indicates no permission), which means that the user can read but not edit or execute the file.

You can add the octal values for permission flags to generate a numeric representation of the file permissions for Owner, Group, and Other:

- 1 = execute only
- 2 = write only
- 3 = write and execute (1+2)
- 4 = read only
- 5 = read and execute (4+1)
- 6 = read and write (4+2)
- 7 = read and write and execute (4+2+1)

The security permissions exercised by Owner/Group/Other are typically expressed as string or a three-digit number. **Table 55** provides examples of different file permissions.

Table 55: File Permissions

Permission	Octal value	What It Means
-rwxrwxrwx	777	Read, write, and executable for Owner/Group/Other
-rw-rw-r--	664	Read and write for Owner/Group; read access for Other
-rw-rw----	660	Read and write for Owner/Group; no access for Other
-rwx------	700	Read, write, and executable for owner only
-rw-rw-rw	666	Read and write for owner, group, and all others

The UNIX chown command lets you change the owner or group (or both) associated with a file or directory. The UNIX chmod command lets you change the permissions of files and directories.

Using the User File Creation Mode Mask

The user file mode creation mode mask (often abbreviated as umask) determines the default file system mode for newly created files of the current process. Linux hosts typically have a hierarchy of umask values: a server-level umask value, which can be overridden by a user-, shell-, or application-level umask value. The result is an ambient umask value, which determines what file permissions are used when files are created by any given process.

The umask value is a three-digit octal number. The first digit sets the mask for Owner, the second for Group, and the third for Other. The umask value identifies the permissions that are withheld when a file is created: the umask value is subtracted from the full access mode value (777) to determine the access permissions for a new file. For example, if the umask value for a process is set to 022, the Write permission for Group and Other are withheld from the full access mode value (777), resulting in a file permission of 755 (rwxr-xr-x). Similarly, if the umask value of 177 is configured for a process (explicitly or by virtue of the ambient umask), files created by the process have a file permission of 600 (rw-------). **Table 56** summarizes the result of using different octal numbers in a umask value.

Table 56: Summary of umask Permissions

Octal Number	Access	Permission Resulting From umask Value
0	rwx	Read, Write, Execute
1	rw-	Read, Write
2	r-x	Read, Execute only
3	r--	Read only
4	-wx	Write, Execute only
5	-w-	Write only
6	--x	Execute only
7	---	No permissions

The umask value affects a file's access permissions only when the file is created. If you change the umask value, access permissions for existing files are not affected. Similarly, you can use the chown and chmod commands to change a file's access permissions after the file has been created.

Implementing Default File Permissions in Steel-Belted Radius

The RADIUSMASK parameter in the sbrd.conf file specifies the application-level umask value used to establish access permissions for all files created by Steel-Belted Radius. Refer to the Steel-Belted Radius Reference Guide for information on configuring the sbrd.conf file.

If you do not specify a value for the RADIUSMASK parameter, Steel-Belted Radius uses the ambient umask value established by the server-, user- or shell-level umask value to determine the access permissions for files it creates.

Some log files have explicit controls that let you override the umask value established by the RADIUSMASK parameter or the ambient umask value. See Implementing Override File Permissions in Steel-Belted Radius for more information on overriding the application-level default umask value.

As previously noted, the umask value affects a file's access permissions only when the file is created. If you change the RADIUSMASK setting, new files created by Steel-Belted Radius are assigned the access permission specified by the new setting. This includes files that roll over periodically; the existing file would retain the access file permission it received when it was created, and the new file would be assigned the access permission specified by the new RADIUSMASK value.

Note: The Execute file permission value for files created by Steel-Belted Radius is always set to None for Owner, Group, and Other. Thus, a umask value of 0 (no restrictions) is equivalent to a umask value of 1 (read/ write permission) for files created by Steel-Belted Radius.

Implementing Override File Permissions in Steel-Belted Radius

To override file permissions established by the Steel-Belted Radius RADIUSMASK or the ambient umask for specific log files, you must modify the LogFilePermissions parameter in the applicable initialization (.ini) file.

Table 57 identifies the configuration file you must modify to configure non-default file permissions for Steel-Belted Radius log files.

Table 57: Configuration Files for Setting Log File Permissions

Controlled Files	Configuration File
Server Diagnostics log (RADIUS log)	radius.ini
Authentication Reporting Library accepts log	authReportAccept.ini
Authentication Reporting Library bad shared secret log	authReportBadSharedSecret.ini
Authentication Reporting Library rejects log	authReportReject.ini
Authentication Reporting Library unknown client log	authReportUnknownClient.ini
Authentication Logging Library logs and header check-point logs	authlog.ini
Accounting Library logs and header check-point logs	account.ini
Server Statistics logs and header check-point logs	statlog.ini

The syntax for the LogFilePermissions parameter is:

```
LogfilePermissions = owner:group mode
```

- Specify the owner and group settings by entering character strings or decimal integers, as used for arguments to the UNIX chown(1) command. For example, ralphw:proj, ralphw:120, or 1007:120.
- Specify the mode setting as a character string or an octal integer. When permissions are specified as a character string, they follow the format that is used by the UNIX ls(1) command; for example, rw-rw-rw-. When permissions are specified as an octal integer, they follow the format used for arguments to the UNIX chmod(1) command; for example, 666.

Note: You can specify only read/write permissions for a Steel-Belted Radius file. You cannot specify execute permissions for Steel-Belted Radius files.

The value of each LogFilePermissions parameter is read when the Steel-Belted Radius server is started or restarted. The value of the LogFilePermissions parameter in the radius.ini file is also read when you issue a HUP command to the Steel-Belted Radius server.

- If you enter a valid value for a LogfilePermissions parameter, the ownership and permissions of the controlled log file are set as specified whenever the file is opened or created.
- If you do not enter a value for a LogfilePermissions parameter, the ownership and permissions of the controlled file are not changed. The controlled file is created using the ownership of the account that

is executing the server and the permissions that are derived from the default RADIUSMASK value or from the ambient umask setting. If the file already exists, new information is appended without changing the existing ownership and permissions of the controlled file.

- If you enter an invalid value for a LogfilePermissions setting, then the ownership of the controlled log file defaults to the effective user/group ID of the server process (normally root:root on Linux), and the permissions for the controlled file default to 0600 (-rw-------). This ensures that the affected log file can always be opened without any escalation of file access privileges. Messages similar to the following are logged whenever an explicit file access control is misconfigured:

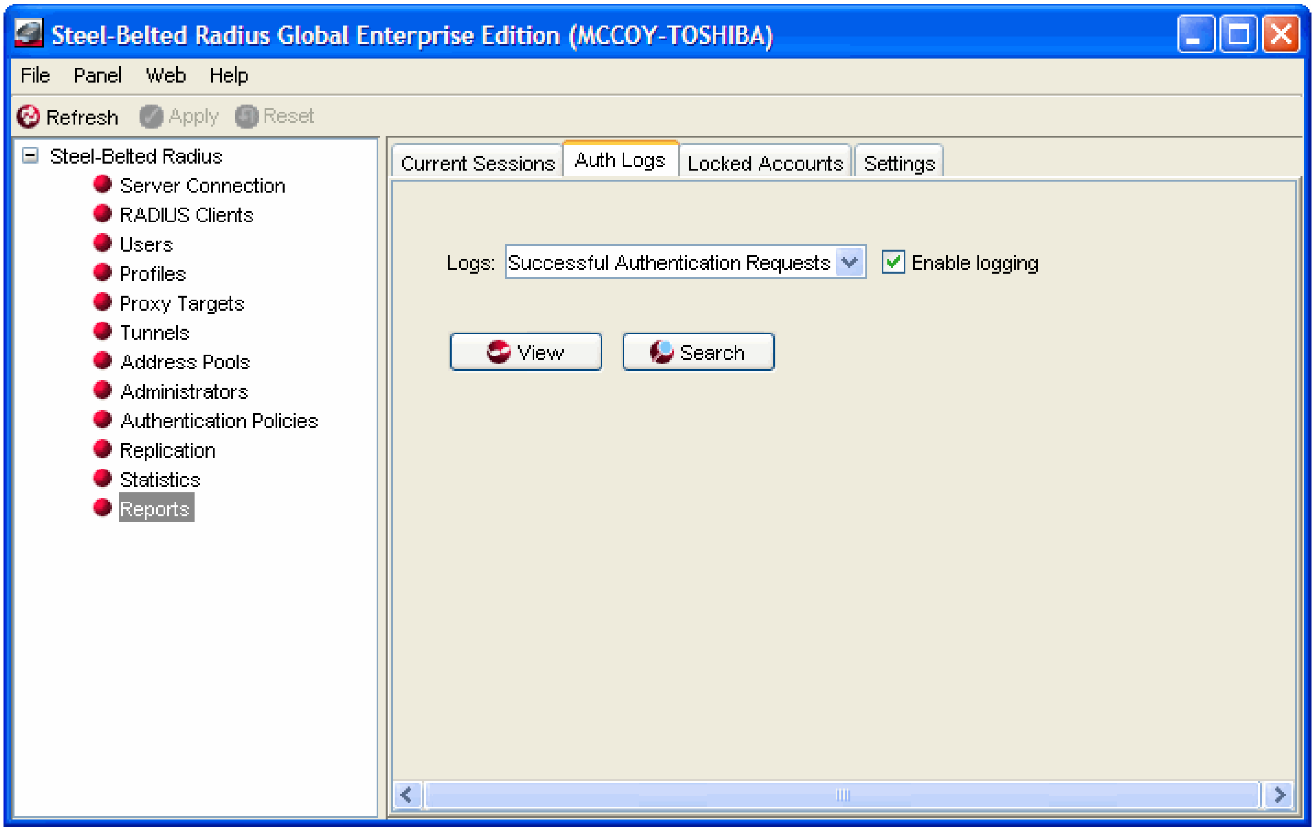
```
Invalid LogfilePermissions specified in radius.ini [Configuration]: -rwx------
Server log file permissions defaulted to 0:0 0600
```

Enabling and Disabling the Authentication Log Files

To enable an authentication log file:

1. Open the Reports panel and click the Auth Logs tab.

Figure 227: Reports Panel: Auth Logs Tab



Logs: Successful Authentication Requests

☑ Enable logging

2. Use the Logs list to select the authentication log file you want to enable or disable.

3. Click the Enable logging check box to enable the specified authentication log.

   Unclick the Enable logging check box to disable the specified authentication log.

Viewing the Authentication Log Files

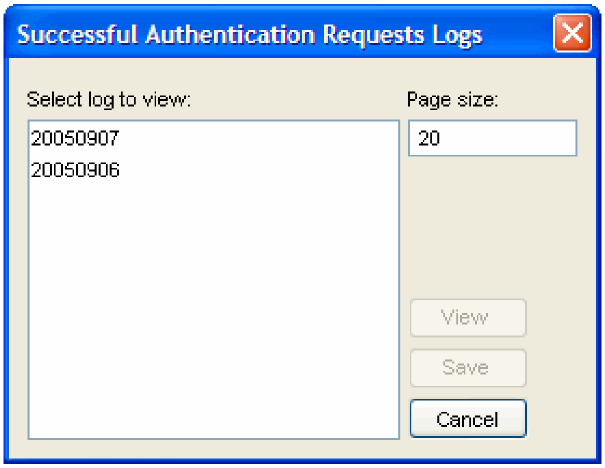
To display an authentication log file:

1. Open the Reports panel and click the **Auth Logs** tab.

2. Use the **Logs** list to select the log file you want to display.

3. Click the **View** button. The Log List dialog opens.

Figure 228: Log List Dialog



Successful Authentication Requests Logs

Select log to view:

20050907
20050906

Page size: 20

4. Select the log you want to display and click **View**.

   By default, SBR Administrator displays the authentication log file 20 lines at a time. To change the number of lines displayed, enter a different number in the **Page size** field before you click **View**.

5. When the authentication log file dialog opens, click the **Up** and **Down** arrows to page through the log file.

   To sort the authentication log file, click the appropriate column header.

   To print the authentication log file, click the **Print** button.

   To refresh the authentication log file display, click the **Refresh** button.

6. When you are finished, click **Close**.

Saving the Log Files

To save an authentication log file to a text file:

1. Open the Reports panel and click the **Auth Logs** tab.

2. Use the **Logs** list to select the authentication log file you want to save.

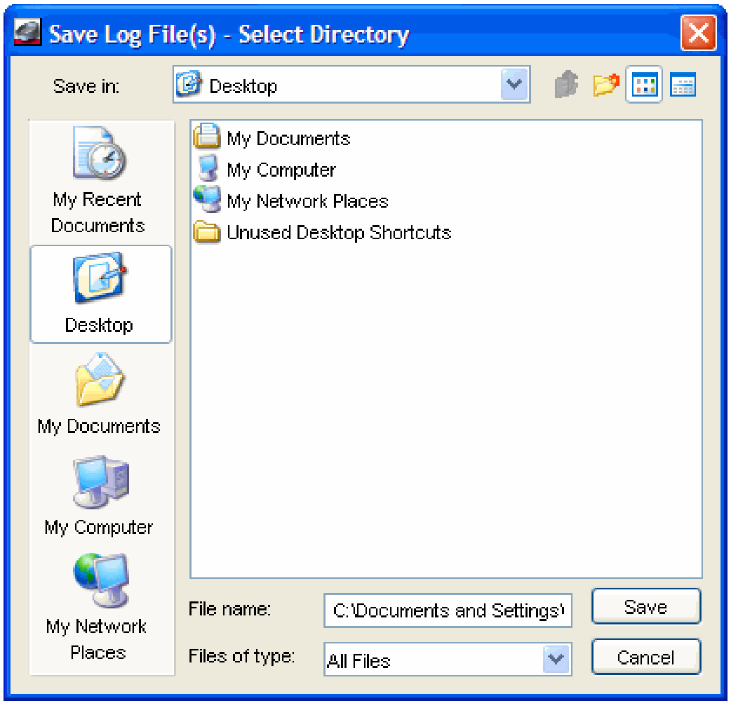
3. Click the **View** button.

   The Log List dialog (Figure 228: Log List Dialog) opens.

4. Select the log you want to save and click **Save.**

   The Save Log File As dialog opens.

Figure 229: Save Log File As Dialog



5. Specify the name and destination for the log file and click **Save.**

Searching the Log Files

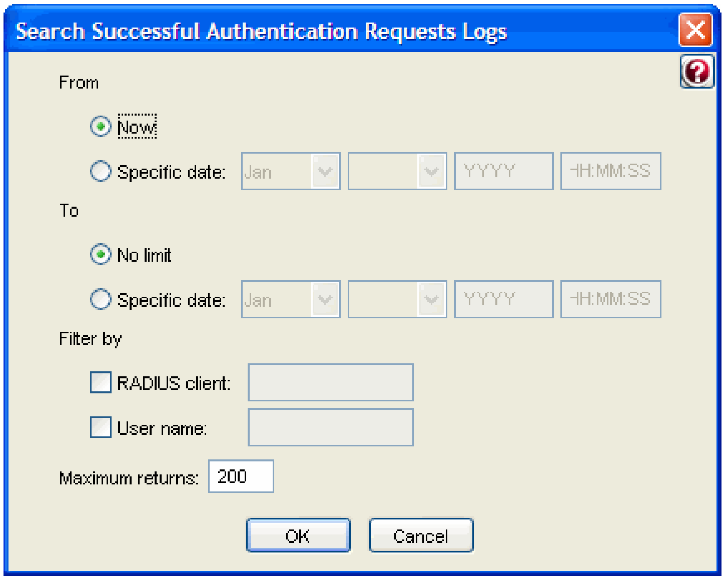
You can search the Steel-Belted Radius authentication log files to display messages within a specified time range, messages relating to a specific client, or messages relating to a specific user.

To search the authentication log files:

1. Open the Reports panel and click the **Auth Logs** tab.
2. Use the **Logs** list to select the type of authentication log file you want to search.
3. Click the **Search** button.

   The Search Logs dialog (**Figure 230: Search Logs Dialog**) opens.

Figure 230: Search Logs Dialog



Search Successful Authentication Requests Logs

From

● Now

○ Specific date: Jan YYYY HH:MM:SS

To

● No limit

○ Specific date: Jan YYYY HH:MM:SS

Filter by

□ RADIUS client:

□ User name:

Maximum returns: 200

4. If you want to search the authentication log file for messages within a specified time range:
   a. Specify the starting date/time in the range by clicking the **Now** radio button or by clicking the **From: Specific date** radio button.
   b. Specify the ending date/time in the range by clicking the **No limit** radio button or by clicking the To: Specified date radio button and entering a date and time.
5. If you want to filter message so that you see only those relating to a specified RADIUS client, click the **RADIUS client** check box and enter the name of the **RADIUS client** in the RADIUS client text field.
6. If you want to filter message so that you see only those relating to a specified user, click the **User name** check box and enter the name of the user in the User name text field.
7. If you want to limit the number of messages you want SBR Administrator to display, enter a number in the **Maximum returns** field.
8. Click **OK.**

Using the Locked Accounts List

Account lockout lets you disable an account after a configurable number of failed login attempts within a configurable period. For example, if a user enters an incorrect password three times within two minutes, Steel-Belted Radius can lock out the user's account temporarily. During the lockout period, the user cannot log in, even with the correct password.

When a user account is locked out, the user must wait until the lockout period expires, or unitl a network administrator can clear the lockout status for the account.

Note: Do not enable account lockout and account redirection at the same time. If account lockout and account redirection are both enabled, account lockout is used and account redirection settings are ignored. For information on account redirection, see "Account Redirection".

Note: Account lockout state is not maintained if Steel-Belted Radius is restarted.

Configuring Locked Account Settings

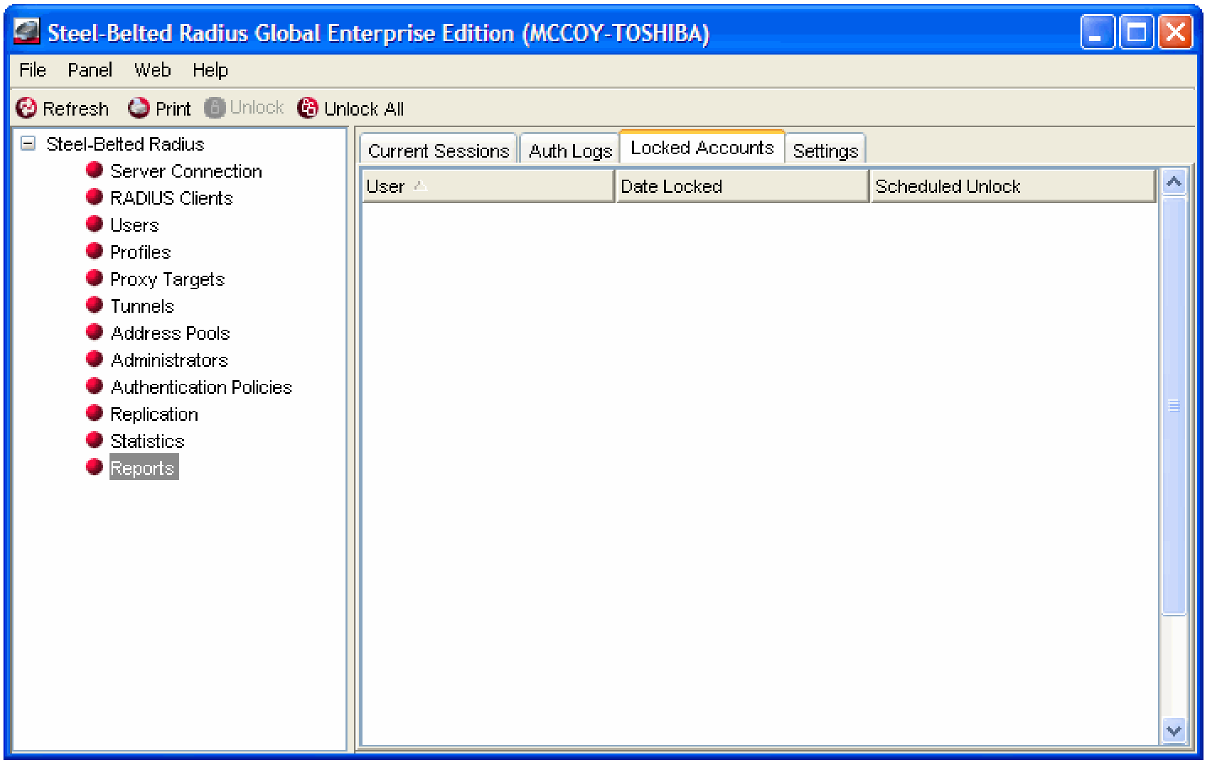
To configure account lockout, edit the lockout.ini file. For information on the lockout.ini file, refer to the Steel-Belted Radius Reference Guide.

Displaying the Locked Accounts List

The Locked Accounts list displays the list of user accounts that have been locked. To display the Locked Accounts list:

1. Open the Reports panel.
2. Click the Locked Accounts tab.

Figure 231: Reports Panel: Locked Accounts Tab



Unlocking a Locked Account

To unlock a locked account:

1. Open the Reports panel.
2. Click the Locked Accounts tab (Figure 231: Reports Panel: Locked Accounts Tab).
3. Select the account you want to unlock from the list.
4. Click the Unlock button in the SBR Administrator toolbar (or right-click the entry and choose Unlock from the context menu).

To unlock all currently locked accounts, click the Unlock All button in the SBR Administrator toolbar.

Note: You can use the LDAP configuration interface to clear a locked-out account by creating and executing an LDIF file with the following commands:
dn: user=user_name, radiusstatus=lockout, o=radius changetype: delete

where user_name is the name of the locked-out user.
For information on using the LDAP configuration interface, see “LDAP Configuration Interface”.

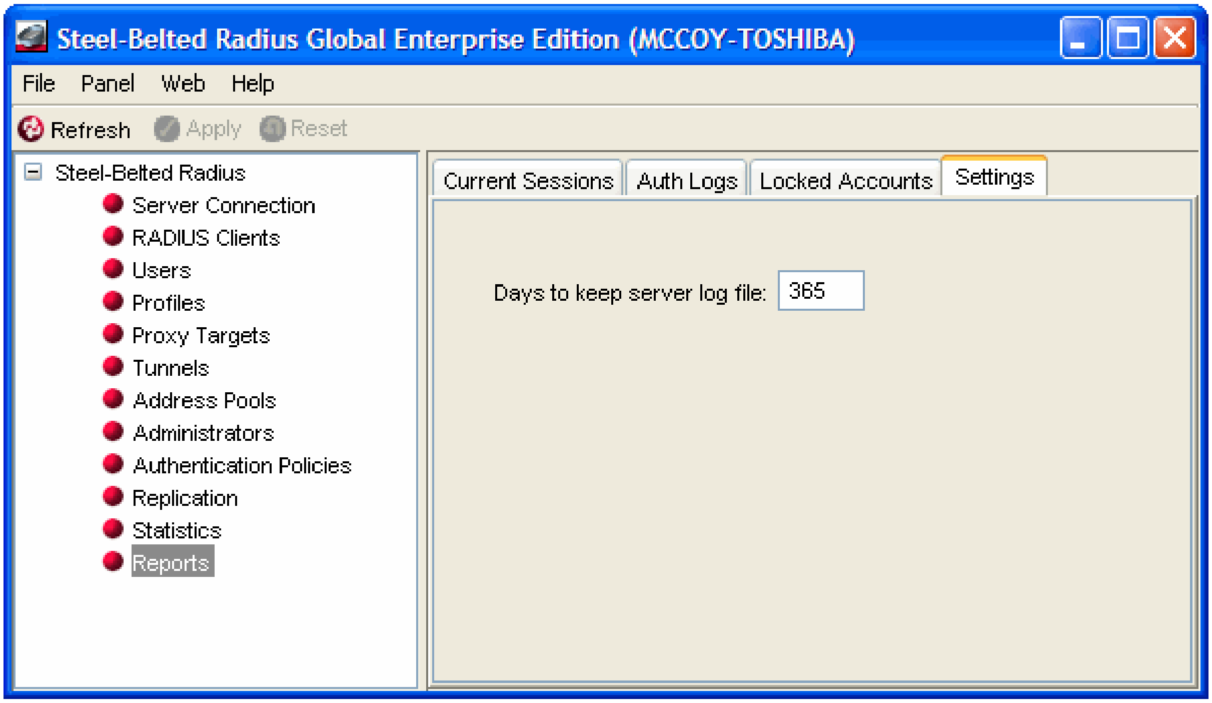
Configuring the Log Retention Period

Each day at midnight, the previous day’s log files are completed, and new log files are created for the new day’s transactions. To prevent the log files from filling up available disk space, you can configure Steel-Belted Radius to discard the log files after a specified number of days.

To configure the log retention period:

1. Open the Reports panel and click the **Settings** tab.
2. When the Settings tab opens, enter the number of days you want Steel-Belted Radius to retain log file in the **Days to keep server log file** field.

Figure 232: Reports Panel: Settings Tab



Days to keep server log file: 365

Using the Server Log File

The server log file records RADIUS events, such as server startup or shutdown or user authentication or rejection, as a series of messages in an ASCII text file. Each line of the server log file identifies the date and time of the RADIUS event, followed by event details. You can open the current log file while Steel-Belted Radius is running.

Server log files are kept for the number of days specified in the Settings tab in the Reports panel (described in “Configuring the Log Retention Period”) and then deleted to conserve disk space.

Optionally, you can specify a maximum size for a server log file by entering a non-zero value for the LogfileMaxMBytes setting in the [Configuration] section of the radius.ini file.

- If a maximum file size is set, the server log filename identifies the date and time it was opened (YYYYMMDD_HHMM.log). When the current server log file approaches the specified number of megabytes (1024 x 1024 bytes), the current log file is closed and a new one is opened. The closed file will be slightly smaller than the specified maximum file size.

- If the maximum file size is set to 0 (or if the LogfileMaxMBytes setting is absent), the server log file size is ignored and log file names are datestamped to identify when they were opened (YYYYMMDD.log).

Note: The size of the log file is checked once per minute, and the log file cannot roll over more than once a minute. The log file may exceed the specified maximum file size temporarily (for less than a minute) after it passes the LogfileMaxMBytes threshold between size checks.

By default, server log files are located in the RADIUS database directory. You can specify an alternate destination directory in the [Configuration] section of the radius.ini file.

Level of Logging Detail

You can control the level of detail recorded in server log files by use of the LogLevel, LogAccept, and LogReject settings.

The LogLevel setting determines the level of detail given in the server log file. The LogLevel can be the number 0, 1, or 2, where 0 is the least amount of information, 1 is intermediate, and 2 is the most verbose. The LogLevel setting is specified in the [Configuration] section of radius.ini and in the [Settings] sections of .aut files.

The LogAccept and LogReject flags allow you to turn on or off the logging of Access-Accept and Access-Reject messages in the server log file. These flags are set in the [Configuration] section of radius.ini: a value of 1 causes these messages to be logged, and a value of 0 causes the messages to be omitted. An Accept or Reject is logged only if LogAccept or LogReject, respectively, is enabled and the LogLevel is “verbose” enough for the message to be recorded.

The TraceLevel setting specifies whether packets should be logged when they are received and being processed, and what level of detail should be recorded in the log.

If you alter the LogLevel or TraceLevel settings, you can have them take effect without restarting the server by issuing the following command:

- Linux: Enter the kill -HUP pid command.

- Windows: Issue the radhup command.

Using the Authentication Log File

The authentication log file records each RADIUS authentication request received by Steel-Belted Radius. Authentication log files are Comma Separated Value (CSV) ASCII text files that can be imported into a spreadsheet or database program.

Authentication log files are located in the RADIUS database directory area by default, although you can specify an alternate destination directory in the [Configuration] section of authlog.ini. Authentication log files are named yyyymmdd.authlog, where yyyy is the 4-digit year, mm is the month, and dd is the day on which the log file was created.

Authentication log files are kept for the number of days specified in the Settings tab of the Reports panel, and are deleted after that.

The current log file can be opened while Steel-Belted Radius is running.

Authentication Log File Format

The first five fields in every authentication log entry are required by Steel-Belted Radius:

- **Date—**The date when the event occurred

- **Time—**The time when the event occurred
- **RAS-Client—**The name or IP address of the RADIUS client sending the authentication request
- **Full-Name—**The fully distinguished name of the user, based on the authentication performed by the RADIUS server
- **ACC/REJ—**The result of the authentication request (ACCEPT or REJECT)

The RADIUS attributes specified in the authlog.ini file appear next. Attributes in the authlog.ini file beginning with a semicolon (;), are commented out, and their values are not recorded in the authentication log file.

User-Name

NAS-IP-Address

NAS-Port

Service-Type

Framed-Protocol

Framed-IP-Address

Framed-IP-Netmask

Framed-Compression

Login-IP-Host

Callback-Number

State

Called-Station-Id=

Calling-Station-Id=

NAS-Identifier=

Proxy-State=

Login-LAT-Service

Login-LAT-Node

Login-LAT-Group

Event-Timestamp

NAS-Port-Type

Port-Limit

Login-LAT-Port

Note: If the User-Password attribute is included in the authlog.ini file, it is ignored during processing to prevent exposing users' cleartext passwords in the log file.

You can include vendor-specific attributes if the device sending the authentication packet supports them. For more information, see "Vendor-Specific Attributes".

You can edit the authlog.ini file to add, remove, or reorder the standard RADIUS or vendor-specific attributes that are logged. For information on authlog.ini, refer to the ***Steel-Belted Radius Reference Guide***.

First Line Headings

The first line of the authentication log file lists the names of all the attributes that have been enabled for logging, in the order in which they are logged. This first line serves as a complete set of column headings for the remaining entries in the file.

The content of the first line depends on the attributes specified in the authlog.ini file. The following example shows the heading line and an authentication log file entry consisting of the required attributes.

```
“Date”,”Time”,”RAS-Client”,”Full-Name”,”ACC/REJ”

“7/3/2003”,”12:11:55”,”RRAS”,”EdisonCarter”,”ACCEPT”,
```

Comma Placeholders

Log entries may not include every attribute listed in the first line of the authentication log file. When Steel-Belted Radius records the event in the authentication log file, it uses a comma “placeholder” to mark empty entries, so that all entries remain aligned with their headings.

For example, the following log entries indicate that Bob’s authentication request was rejected but Alice’s authentication request was accepted. The reported fields include Called-Station-Id, Calling-Station-Id, and Port-Limit. Note that the attributes listed in the log heading that were not returned for the authentication events are separated with commas.

```
“Date”,”Time”,”RAS-Client”,”Full-Name”,”Acc/Rej”,

“User-Name”,”NAS-IP-Address”,”NAS-Port”,”Service-Type”,

“Framed-Protocol”,”Framed-IP-Address”,     “Framed-IP-Netmask”,”Framed-Compression”,

“Login-IP-Host”,”Callback-Number”,”State”,   “Called-Station-Id”,”Calling-Station-Id”,

“NAS-Identifier”,”Proxy-State”,     “Event-Timestamp”,”NAS-Port-Type”,”Port-Limit”,

“Login-LAT-Port””07/14/2003”,”13:39:10”,”192.168.2.42”,

“BOB”,”REJECT”,,,,,,,,,,,,”Alice’s   Office”,”Bob’s

Office”,,,,,”5”,”07/14/2003”,”13:43:26”,”192.168.2.42”,

“ALICE”,”ACCEPT”,,,,,,,,,,,,”Bob’s Office”,”Alice’s Office”,,,,,”5”,
```

Using the Accounting Log File

RADIUS accounting events are recorded in the accounting log file. Accounting events include START messages, which indicate the beginning of a connection; STOP messages, which indicate the termination of a connection; and INTERIM messages, which indicate a connection is ongoing.

Accounting log files use comma-delimited, ASCII format, and are intended for import into a spreadsheet or database program. Accounting log files are located in the RADIUS database directory area by default, although you can specify an alternate destination directory in the [Configuration] section of the account.ini file. Accounting log files are named yyyymmdd.ACT, where yyyy is the 4-digit year, mm is the month, and dd is the day on which the log file was created.

Accounting log files are kept for the number of days specified in the Settings tab of the Reports panel, and are deleted after that to conserve disk space.

The current log file can be opened while Steel-Belted Radius is running.

Accounting Log File Format

The first six fields in every accounting log entry are provided by Steel-Belted Radius for your convenience in reading and sorting the file:

- Date—The date when the event occurred
- Time—The time when the event occurred
- RAS-Client—The name or IP address of the RADIUS client sending the accounting record
- Record-Type—START, STOP, INTERIM, ON, or OFF, the standard RADIUS accounting packet types
- Full-Name—The fully distinguished name of the user, based on the authentication performed by the RADIUS server
- Auth-Type—A number that indicates the class of authentication performed:

  0—Native

  10—SecurID User 11—

  SecurID Prefix 12—

  SecurID Suffix 13—

  TACACS+ User 16—

  TACACS+ Prefix 17—

  TACACS+ Suffix 100—Tunnel

  User 200—External

  Database (other)—Proxy

By default, the standard RADIUS attributes follow the Auth-Type identifier. See "Standard RADIUS Accounting Attributes".

You can include vendor-specific attributes if the device sending the accounting packet supports them. For more information, see "Vendor-Specific Attributes".

You can edit the account.ini initialization file to add, remove or reorder the standard RADIUS or vendor-specific attributes that are logged. For information on account.ini, refer to the ***Steel-Belted Radius Reference Guide***.

First Line Headings

The first line of the accounting log file is a file header that lists the attributes that have been enabled for logging in the order in which they are logged. The following example of a first line shows required headings in bold italic, standard RADIUS headings in bold, and vendor-specific headings in regular text:

"Date","Time", "RAS-Client", "Record-Type", "Full-Name", "Auth-Type", "User-Name",

"NAS-Port", "Acct-Status-Type", "Acct-Delay-Time", "Acct-Input-Octets",

"Acct-Output-Octets", "Acct-Session-Id", "Acct-Authentic","Acct-Session-Time",

"Acct-Input-Packets", "Acct-Output-Packets", "Acct-Termination-Cause",

"Acct-Multi-Session-Id", "Acct-Link-Count","Acc-Err-Message",

“Nautica-Acct-SessionId”,”Nautica-Acct-Direction”,

“Nautica-Acct-CauseProtocol”,”Nautica-Acct-CauseSource”,

“Telebit-Accounting-Info”,”Last-Number-Dialed-Out”,

“Last-Number-Dialed-In-DNIS”,”Last-Callers-Number-ANI”,

“Channel”,”Event-Id”,”Event-Date-Time”, “Call-Start-Date-Time”,”Call-End-Date-Time”,

“Default-DTE-Data-Rate”,”Initial-Rx-Link-Data-Rate”,

“Final-Rx-Link-Data-Rate”,”Initial-Tx-Link-Data-Rate”,

“Final-Tx-Link-Data-Rate”,”Sync-Async-Mode”,

“Originate-Answer-Mode”,”Modulation-Type”,

“Equalization-Type”,”Fallback-Enabled”,”Characters-Sent”,

“Characters-Received”,”Blocks-Sent”,”Blocks-Received”,

“Blocks-Resent”,”Retrains-Requested”,”Retrains-Granted”,

“Line-Reversals”,”Number-Of-Characters-Lost”,

“Number-of-Blers”,”Number-of-Link-Timeouts”,

“Number-of-Fallbacks”,”Number-of-Upshifts”,

“Number-of-Link-NAKs”,”Back-Channel-Data-Rate”,

“Simplified-MNP-Levels”,”Simplified-V42bis-Usage”, “PW_VPN_ID”

Comma Placeholders

Steel-Belted Radius writes accounting events to the accounting log file, If an event recorded in the accounting log file does not have data for every attribute, a comma “placeholder” marks the empty entry, so that all entries remain correctly aligned with their headings. For example, based on the “first line” of headings described above, the following is a valid accounting log entry, in which the value of the Acct-Status-Type attribute is 7:

“12/23/1997”,”12:11:55”,”RRAS”,”Accounting-On”,

,,,,7,,

,,,,,,,,,,,,,,,,,,

Standard RADIUS Accounting Attributes

Table 58 lists the standard RADIUS accounting attributes defined in RFC 2866, “RADIUS Accounting.”

Table 58: Standard RADIUS Accounting Attributes

User-Name	The name of the user as received by the client.
NAS-Port	The port number on the client device.

Acct-Status-Type	A number that indicates the beginning or ending of the user service: 1—Start 2—Stop 3— Interim-Acct 7— Accounting-On 8— Accounting-Off
Acct-Delay-Time	Indicates how many seconds the client has been trying to send this record, which can be subtracted from the time of arrival on the server to find the approximate time of the event generating this request.
Acct-Input-Octets	Number of octets (bytes) received by the port over the connection; present only in STOP records.
Acct-Output-Octets	Number of octets (bytes) sent by the port over the connection; present only in STOP records.
Acct-Session-Id	Identifier used to match START and STOP records in a log file.
Acct-Authentic	Indicates how the user was authenticated by RADIUS, the network access device (local), or another remote authentication protocol: 1—RADIUS 2—Local 3—Remote
Acct-Session-Time	Elapsed time of connection in seconds; present only in STOP records.
Acct-Input-Packets	Number of packets received by the port over the connection; present only in STOP records.
Acct-Output-Packets	Number of packets sent by the port over the connection; present only in STOP records.
Acct-Termination-Cause	Number that indicates how the session was terminated; present only in STOP records: 1—User Request 2—Lost Carrier 3— Lost Service 4—Idle Timeout 5—Session Timeout 6—Admin Reset 7—Admin Reboot 8—Port Error 9—NAS Error 10—NAS Request 11—NAS Reboot 12—Port Unneeded 13—Port Preempted

	14—Port Suspended 15—Service Unavailable 16—Callback 17—User Error 18—Host Request
Acct-Multi-Session-Id	Unique accounting identifier to make it easy to link together multiple related sessions in a log file.
Acct-Link-Count	The count of links that are known to have been in a given multi-link session at the time the accounting record is generated.

Chapter 37
Logging and Reporting via WebGUI

This chapter describes how to set up and use logging and reporting functions in Steel-Belted Radius via WebGUI.

Logging Files

The following files establish settings for logging and reporting. For more information about these files, refer to the *Steel-Belted Radius Reference Guide*.

Table 53: Logging and Reporting Files

File Name	Function
account.ini	Controls how RADIUS accounting attributes are logged.
authlog.ini	Controls how RADIUS authentication requests are logged by Steel-Belted Radius.
authReport.ini	Controls what authentication logs Steel-Belted Radius generates.
authReportAccept.ini	Controls options for the acceptance authentication log file.
authReportBadSharedSecret.ini	Controls options for the invalid shared secret authentication log file.
authReportReject.ini	Controls options for the rejection authentication log file.
authReportUnknownClient.ini	Controls options for the unknown client authentication log file.
events.ini	Controls dilutions and thresholds for Steel-Belted Radius events used to communicate failures, warnings, and other information.
radius.inic	Controls (among other things) the types of messages Steel-Belted Radius records in the server log file and the location of the log directory.

Displaying the Current Sessions List

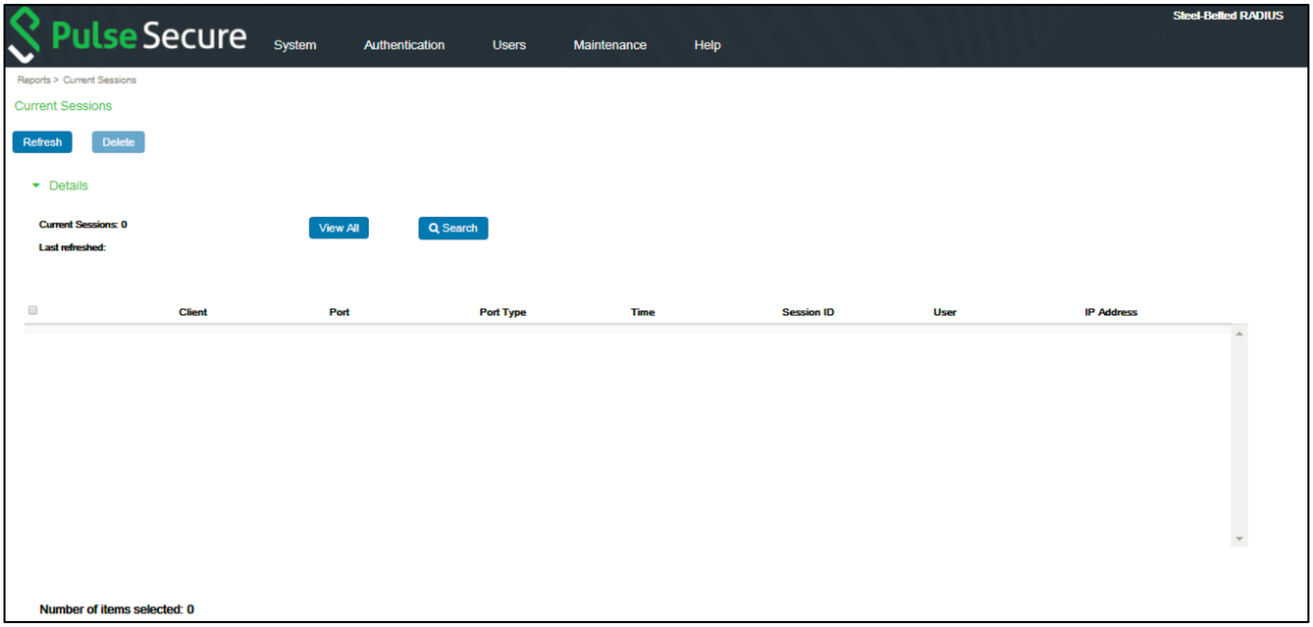
Steel-Belted Radius tracks the status of the user connections that it authenticates. To obtain a real-time snapshot of currently active connections, display the Current Sessions list. Because the Current Sessions list is based on RADIUS accounting data, the list is accurate only if all of your network access devices are configured to support RADIUS accounting.

Each server has its own Current Sessions display. Therefore, when you view this display, it typically reflects only the activity on the Steel-Belted Radius server to which you are connected. The Current Users display on a specific server reflects the activity across your entire RADIUS configuration only if (1) all clients in your configuration support RADIUS accounting, and (2) all clients are configured to send accounting messages to the server you are viewing.

Note: Steel-Belted Radius maintains the Current User list on disk. The information is preserved if you unload and reload the server.

To display the Current Sessions list, choose **System** > **Reports** > **Current Sessions**.

Figure 233: Reports Page: Current Sessions List



Current Sessions: 0

Last refreshed:

□	Client	Port	Port Type	Time	Session ID	User	IP Address

Number of items selected: 0

Table 54 describes the fields for each active session in the Current Sessions list.

Table 54: Sessions List Fields

Field	Meaning
Client	Client The identifier for the RADIUS client, which is typically the name or IP address of the device.
Port	The UDP port number on the RADIUS client that has been assigned to the connection. To determine slot number of the physical port on the RADIUS client, consult the device documentation.
Port Type	Describes how the port is used or configured.
Time	Identifies the date and time at which the connection was opened.
Session ID	Identifies the session key, which is a number generated by the RADIUS client.
User Name	Displays the name of the authenticated user. • If the user is local (native), the field shows only the username, in the form username. • If the user is non-local, the field shows the remote system name as well as the username, in the form \\systemname\username. • If the user is associated with a specific tunnel, the field shows the tunnel name as well as the username, in the form \\tunnelname\username.
IP Address	Identifies the IP address that was assigned to the user from an IP address pool. This field will be blank if a static IP address was assigned.

Note: For tunnel connections, if Steel-Belted Radius was used to authenticate both the user and the tunnel, then two entries are displayed in the Current Sessions page: one entry for the authenticated user, and one for the authenticated tunnel.

Searching the Current Sessions List

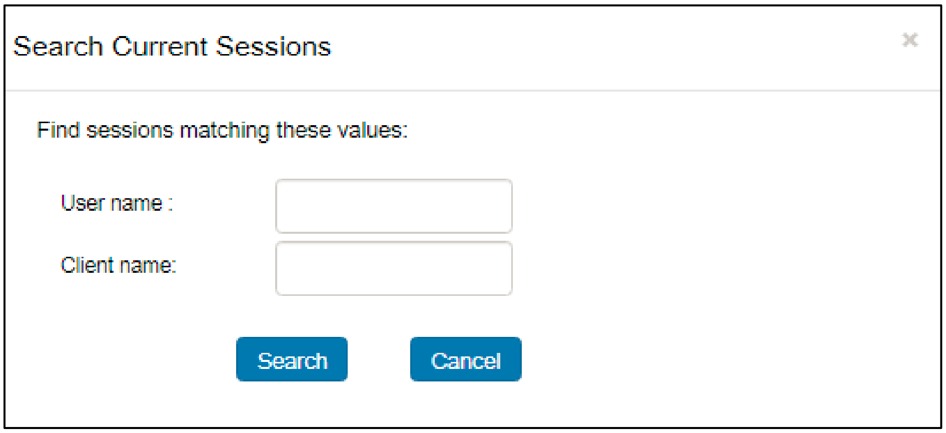
You can search the Current Sessions list to display only those sessions that match user name or client name criteria.

To search the Current Sessions list:

1. Choose System > Reports > Current Sessions.
2. Click the Search button.

   The Search Current Sessions page opens.

Figure 234: Search Current Sessions Page



Search Current Sessions

Find sessions matching these values:

User name :

Client name:

3. Enter the search criteria you want to use.
   - To search for a specific username, enter it in the User name field.
   - To search for a specific RADIUS client, enter a client name in the Client name field.
4. Click Search.

Deleting Entries from the Sessions List

Normally, the system maintains the information in the Sessions list based on accounting information received from the RADIUS client. However, a user who has logged off may still be identified as active in the Current Sessions list if communications between the RADIUS client and Steel-Belted Radius fail or if either the RADIUS client or Steel-Belted Radius is taken down for a period of time.

In most cases, Steel-Belted Radius can correct such anomalies itself. For example, if a new user dials in to the same port on the same RADIUS client, Steel-Belted Radius infers that the previous user must have disconnected and removes the entry.

You can manually correct the Sessions list by highlighting any entry and clicking Delete. This removes the user from the list and decrements the user’s connection count (if it is being tracked) by one. Any pooled IP or IPX address assigned to the deleted user is returned to the appropriate pool.

Displaying the Authentication Log Files

The Auth Logs tab (Figure 235: Reports Page: Auth Logs Tab) on the Reports page lets you enable and display the following authentication log files:

- Successful request authentication log file—The successful request authentication log file identifies the authentication requests that were approved by Steel-Belted Radius.
- Invalid shared secrets authentication log file—The invalid shared secrets authentication log file identifies the authentication requests that failed because a known RADIUS client supplied an incorrect shared secret. This condition is detectable only if the authentication request contains a Message-Authenticator attribute, which is required if credentials are of an EAP type but optional if credentials are PAP, CHAP, or MS-CHAP.
- Failed request authentication log file—The failed request authentication log file identifies the authentication requests that were rejected because the user supplied incorrect credentials.
- Unknown client request authentication log file—The unknown client request authentication log file identifies authentication requests received from unknown RADIUS clients.

File Permissions for Log Files (Linux)

When you run Steel-Belted Radius on a Linux server, you can specify the users who are authorized to read or edit important files, such as authentication and accounting log files. For example, you can specify that system administrators who install and configure Steel-Belted Radius have read/write access for system log files and that network operators who monitor Steel-Belted Radius have read-only (or no) access for system log files.

Security Groups and Permissions

Each file and directory on a Linux server has three security groups associated with it:

- The Owner identifies the person who created or owns the file.
- The Group security group identifies the set of users who are members of the group or groups to which the file Owner belongs. Group members can exercise special privileges with respect to that file. A user can belong to more than one group.
- The Other security group consists of the set of all users who do not belong to Owner or Group.

Each security group has three flags that control what privileges that group can exercise with respect to the file or directory:

- The Read flag (r) determines whether the file can be read. The Read flag has an octal value of 4.
- The Write flag (w) determines whether the security group can create, modify, or delete the file. The Write flag has an octal value of 2.
- The Execute flag (x) determines whether the security group can run a script or executable file. The Execute flag has an octal value of 1.

For example, a file owner might have rwx permission for a file, which indicates the file owner has read/write/execute access to the file. Similarly, Other might have r-- permission (where - indicates no permission), which means that the user can read but not edit or execute the file.

You can add the octal values for permission flags to generate a numeric representation of the file permissions for Owner, Group, and Other:

- 1 = execute only
- 2 = write only

- 3 = write and execute (1+2)
- 4 = read only
- 5 = read and execute (4+1)
- 6 = read and write (4+2)
- 7 = read and write and execute (4+2+1)

The security permissions exercised by Owner/Group/Other are typically expressed as string or a three-digit number. **Table 55** provides examples of different file permissions.

Table 55: File Permissions

Permission	Octal value	What It Means
-rwxrwxrwx	777	Read, write, and executable for Owner/Group/Other
-rw-rw-r--	664	Read and write for Owner/Group; read access for Other
-rw-rw----	660	Read and write for Owner/Group; no access for Other
-rwx------	700	Read, write, and executable for owner only
-rw-rw-rw	666	Read and write for owner, group, and all others

The UNIX chown command lets you change the owner or group (or both) associated with a file or directory. The UNIX chmod command lets you change the permissions of files and directories.

Using the User File Creation Mode Mask

The user file mode creation mode mask (often abbreviated as umask) determines the default file system mode for newly created files of the current process. Linux hosts typically have a hierarchy of umask values: a server-level umask value, which can be overridden by a user-, shell-, or application-level umask value. The result is an ambient umask value, which determines what file permissions are used when files are created by any given process.

The umask value is a three-digit octal number. The first digit sets the mask for Owner, the second for Group, and the third for Other. The umask value identifies the permissions that are withheld when a file is created: the umask value is subtracted from the full access mode value (777) to determine the access permissions for a new file. For example, if the umask value for a process is set to 022, the Write permission for Group and Other are withheld from the full access mode value (777), resulting in a file permission of 755 (rwxr-xr-x). Similarly, if the umask value of 177 is configured for a process (explicitly or by virtue of the ambient umask), files created by the process have a file permission of 600 (rw-------). **Table 56** summarizes the result of using different octal numbers in a umask value.

Table 56: Summary of umask Permissions

Octal Number	Access	Permission Resulting From umask Value
0	rwx	Read, Write, Execute
1	rw-	Read, Write
2	r-x	Read, Execute only
3	r--	Read only
4	-wx	Write, Execute only
5	-w-	Write only
6	--x	Execute only
7	---	No permissions

The umask value affects a file's access permissions only when the file is created. If you change the umask value, access permissions for existing files are not affected. Similarly, you can use the chown and chmod commands to change a file's access permissions after the file has been created.

Implementing Default File Permissions in Steel-Belted Radius

The RADIUSMASK parameter in the sbrd.conf file specifies the application-level umask value used to establish access permissions for all files created by Steel-Belted Radius. Refer to the Steel-Belted Radius Reference Guide for information on configuring the sbrd.conf file.

If you do not specify a value for the RADIUSMASK parameter, Steel-Belted Radius uses the ambient umask value established by the server-, user- or shell-level umask value to determine the access permissions for files it creates.

Some log files have explicit controls that let you override the umask value established by the RADIUSMASK parameter or the ambient umask value. See Implementing Override File Permissions in Steel-Belted Radius for more information on overriding the application-level default umask value.

As previously noted, the umask value affects a file's access permissions only when the file is created. If you change the RADIUSMASK setting, new files created by Steel-Belted Radius are assigned the access permission specified by the new setting. This includes files that roll over periodically; the existing file would retain the access file permission it received when it was created, and the new file would be assigned the access permission specified by the new RADIUSMASK value.

Note: The Execute file permission value for files created by Steel-Belted Radius is always set to None for Owner, Group, and Other. Thus, a umask value of 0 (no restrictions) is equivalent to a umask value of 1 (read/write permission) for files created by Steel-Belted Radius.

Implementing Override File Permissions in Steel-Belted Radius

To override file permissions established by the Steel-Belted Radius RADIUSMASK or the ambient umask for specific log files, you must modify the LogFilePermissions parameter in the applicable initialization (.ini) file.

Table 57 identifies the configuration file you must modify to configure non-default file permissions for Steel-Belted Radius log files.

Table 57: Configuration Files for Setting Log File Permissions

Controlled Files	Configuration File
Server Diagnostics log (RADIUS log)	radius.ini
Authentication Reporting Library accepts log	authReportAccept.ini
Authentication Reporting Library bad shared secret log	authReportBadSharedSecret.ini
Authentication Reporting Library rejects log	authReportReject.ini
Authentication Reporting Library unknown client log	authReportUnknownClient.ini
Authentication Logging Library logs and header check-point logs	authlog.ini
Accounting Library logs and header check-point logs	account.ini
Server Statistics logs and header check-point logs	statlog.ini

The syntax for the LogFilePermissions parameter is:

```
LogfilePermissions = owner:group mode
```

- Specify the owner and group settings by entering character strings or decimal integers, as used for arguments to the UNIX chown(1) command. For example, ralphw:proj, ralphw:120, or 1007:120.
- Specify the mode setting as a character string or an octal integer. When permissions are specified as a character string, they follow the format that is used by the UNIX ls(1) command; for example, rw-rw-rw-. When permissions are specified as an octal integer, they follow the format used for arguments to the UNIX chmod(1) command; for example, 666.

Note: You can specify only read/write permissions for a Steel-Belted Radius file. You cannot specify execute permissions for Steel-Belted Radius files.

The value of each LogFilePermissions parameter is read when the Steel-Belted Radius server is started or restarted. The value of the LogFilePermissions parameter in the radius.ini file is also read when you issue a HUP command to the Steel-Belted Radius server.

- If you enter a valid value for a LogfilePermissions parameter, the ownership and permissions of the controlled log file are set as specified whenever the file is opened or created.
- If you do not enter a value for a LogfilePermissions parameter, the ownership and permissions of the controlled file are not changed. The controlled file is created using the ownership of the account that is executing the server and the permissions that are derived from the default RADIUSMASK value or

from the ambient umask setting. If the file already exists, new information is appended without changing the existing ownership and permissions of the controlled file.

- If you enter an invalid value for a LogfilePermissions setting, then the ownership of the controlled log file defaults to the effective user/group ID of the server process (normally root:root on Linux), and the permissions for the controlled file default to 0600 (-rw-------). This ensures that the affected log file can always be opened without any escalation of file access privileges. Messages similar to the following are logged whenever an explicit file access control is misconfigured:

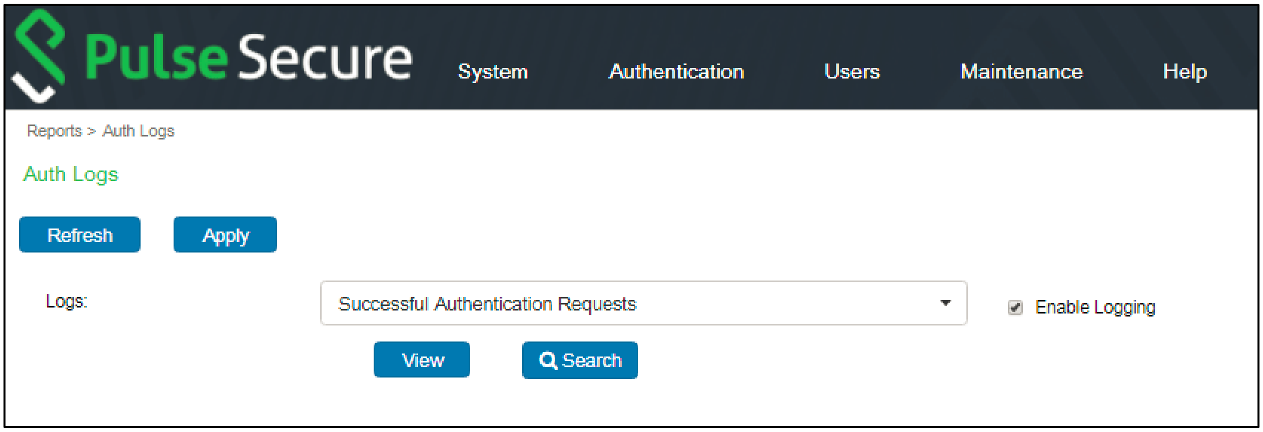
```
Invalid LogfilePermissions specified in radius.ini [Configuration]: -rwx------
Server log file permissions defaulted to 0:0 0600
```

Enabling and Disabling the Authentication Log Files

To enable an authentication log file:

1. Choose System > Reports > Auth Logs.

Figure 235: Reports Page: Auth Logs Tab



Logs: Successful Authentication Requests

☑ Enable Logging

2. Use the Logs list to select the authentication log file you want to enable or disable.
3. Click the Enable logging check box to enable the specified authentication log.

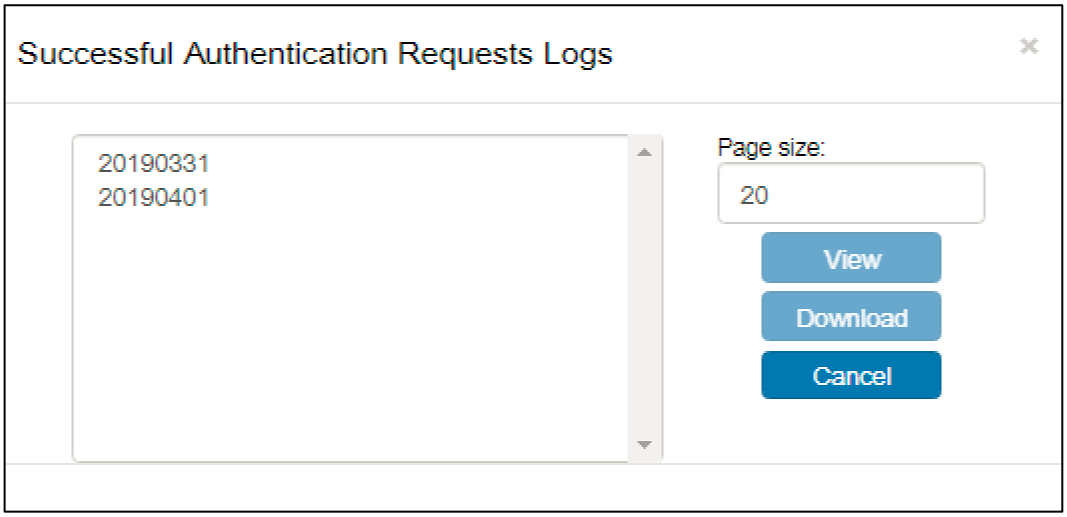
   Unclick the Enable logging check box to disable the specified authentication log.

Viewing the Authentication Log Files

To display an authentication log file:

1. Choose Systems > Reports > Auth Logs.
2. Use the **Logs** list to select the log file you want to display.
3. Click the **View** button. The Log List page opens.

Figure 236: Log List Page



Successful Authentication Requests Logs

20190331
20190401

Page size: 20

4. Select the log you want to display and click **View**.

   By default, SBR Administrator displays the authentication log file 20 lines at a time. To change the number of lines displayed, enter a different number in the **Page size** field before you click **View**.

5. When the authentication log file page opens, click the **Up** and **Down** arrows to page through the log file.

   To sort the authentication log file, click the appropriate column header.

   To refresh the authentication log file display, click the **Refresh** button.

6. When you are finished, click **Close**.

Downloading the Log Files

To download an authentication log file to a text file:

1. Choose System > Reports > Auth Logs.
2. Use the **Logs** list to select the authentication log file you want to save.
3. Click the **Download** button. The log file will be automatically downloaded.

Searching the Log Files

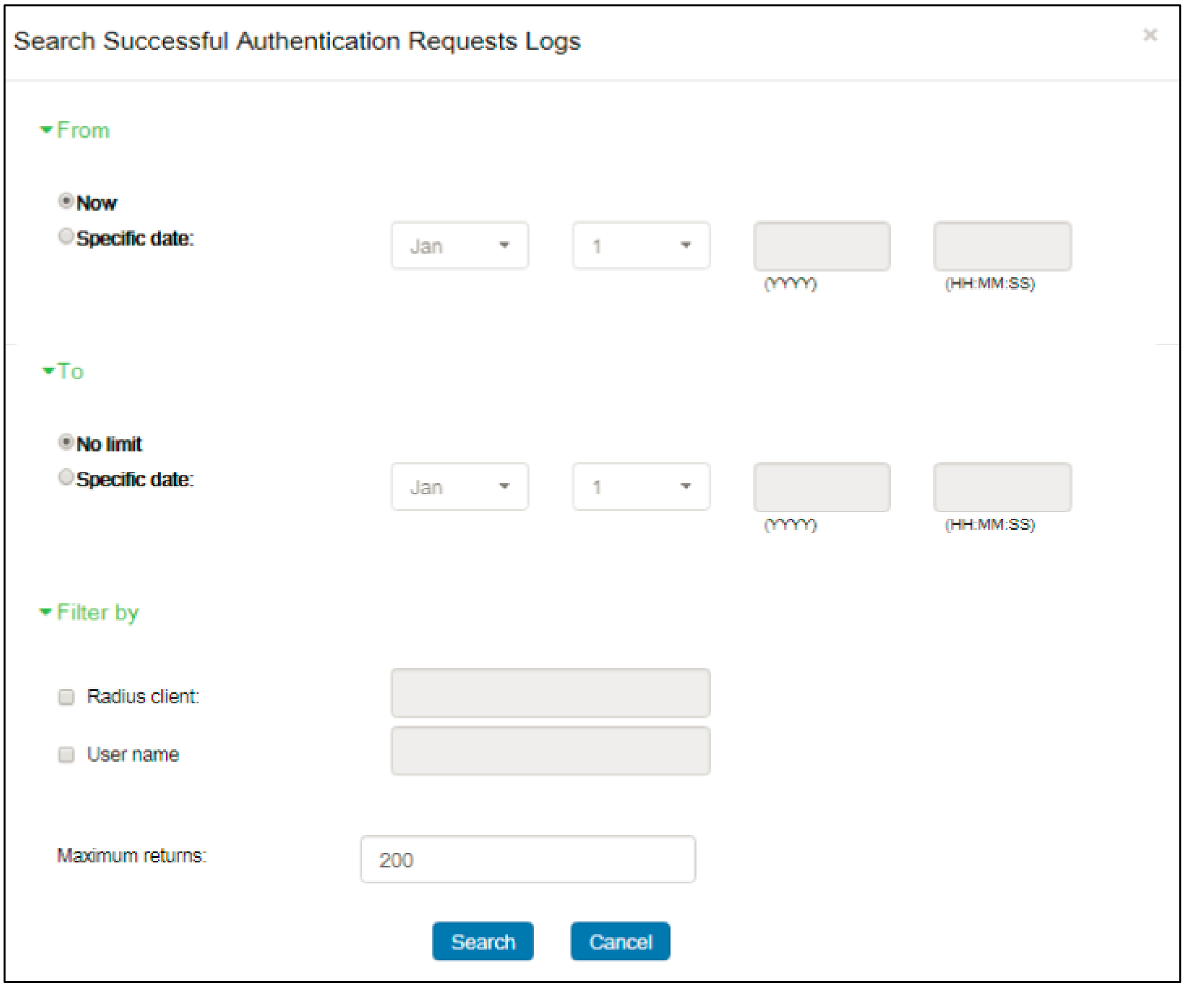
You can search the Steel-Belted Radius authentication log files to display messages within a specified time range, messages relating to a specific client, or messages relating to a specific user.

To search the authentication log files:

1. Choose System > Reports > Auth Logs.
2. Use the **Logs** list to select the type of authentication log file you want to search.
3. Click the **Search** button.

   The Search Logs page (**Figure 237: Search Logs Page**) opens.

Figure 237: Search Logs Page



Search Successful Authentication Requests Logs

From

● Now
○ Specific date: Jan 1 (YYYY) (HH:MM:SS)

To

● No limit
○ Specific date: Jan 1 (YYYY) (HH:MM:SS)

Filter by

□ Radius client:
□ User name

Maximum returns: 200

4. If you want to search the authentication log file for messages within a specified time range:

   a. Specify the starting date/time in the range by clicking the **Now** radio button or by clicking the **From: Specific date** radio button.

   b. Specify the ending date/time in the range by clicking the **No limit** radio button or by clicking the To: Specified date radio button and entering a date and time.

5. If you want to filter message so that you see only those relating to a specified RADIUS client, click the **RADIUS client** check box and enter the name of the **RADIUS client** in the RADIUS client text field.

6. If you want to filter message so that you see only those relating to a specified user, click the **User name** check box and enter the name of the user in the User name text field.

7. If you want to limit the number of messages you want SBR Administrator to display, enter a number in the **Maximum returns** field.

8. Click **OK.**

Using the Locked Accounts List

Account lockout lets you disable an account after a configurable number of failed login attempts within a configurable period. For example, if a user enters an incorrect password three times within two minutes, Steel-Belted Radius can lock out the user's account temporarily. During the lockout period, the user cannot log in, even with the correct password.

When a user account is locked out, the user must wait until the lockout period expires, or unitl a network administrator can clear the lockout status for the account.

Note: Do not enable account lockout and account redirection at the same time. If account lockout and account redirection are both enabled, account lockout is used and account redirection settings are ignored. For information on account redirection, see "Account Redirection".

Note: Account lockout state is not maintained if Steel-Belted Radius is restarted.

Configuring Locked Account Settings

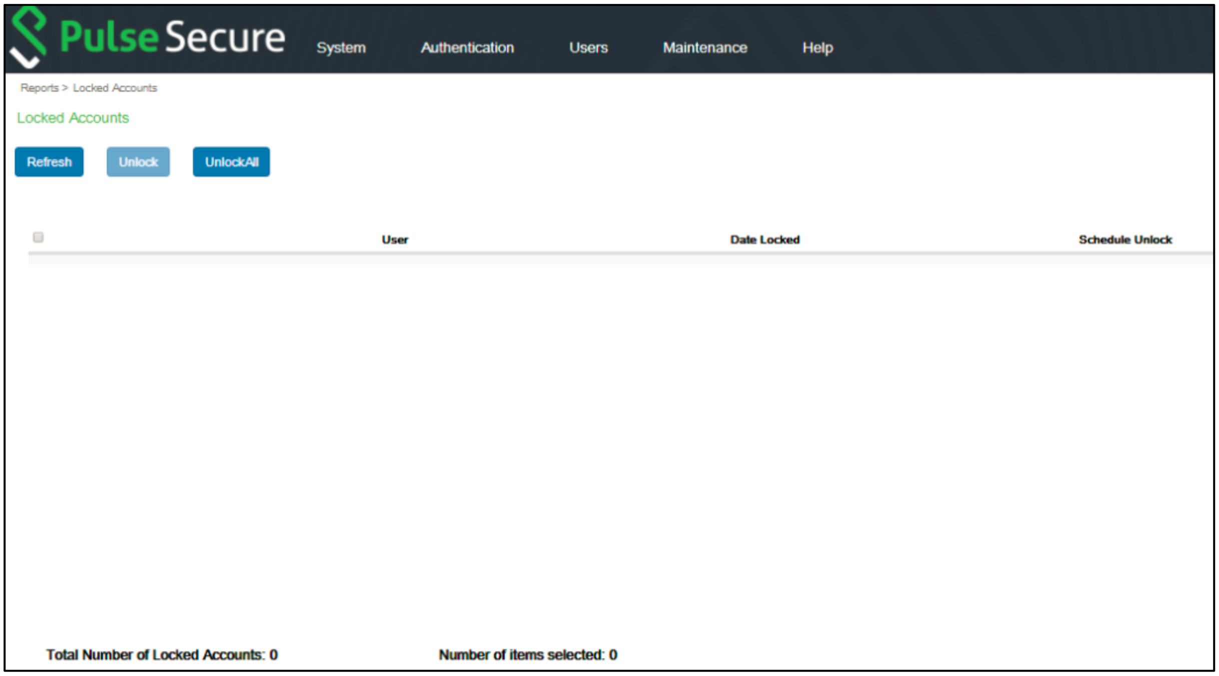
To configure account lockout, edit the lockout.ini file. For information on the lockout.ini file, refer to the Steel-Belted Radius Reference Guide.

Displaying the Locked Accounts List

The Locked Accounts list displays the list of user accounts that have been locked. To display the Locked Accounts list:

1. Choose System > Reports > Locked Accounts.

Figure 238: Reports Page: Locked Accounts Tab



Total Number of Locked Accounts: 0

Number of items selected: 0

Unlocking a Locked Account

To unlock a locked account:

1. Choose System > Reports > Locked Accounts.
2. Click the Locked Accounts tab (**Figure 238: Reports Page: Locked Accounts Tab**).
3. Select the account you want to unlock from the list.
4. Click the Unlock button.

To unlock all currently locked accounts, click the Unlock All button.

Note: You can use the LDAP configuration interface to clear a locked-out account by creating and executing an LDIF file with the following commands:

dn: user=user_name, radiusstatus=lockout, o=radius changetype: delete
where user_name is the name of the locked-out user.
For information on using the LDAP configuration interface, see "LDAP Configuration Interface".

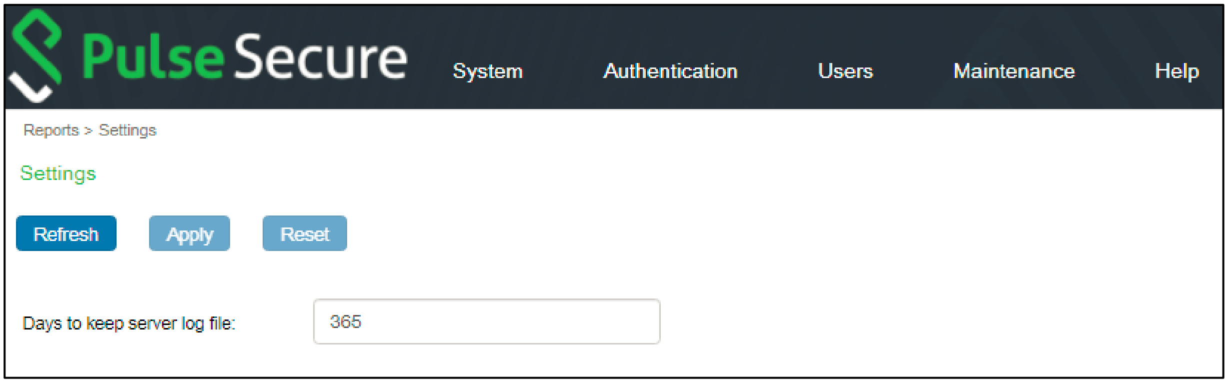
Configuring the Log Retention Period

Each day at midnight, the previous day's log files are completed, and new log files are created for the new day's transactions. To prevent the log files from filling up available disk space, you can configure Steel-Belted Radius to discard the log files after a specified number of days.

To configure the log retention period:

1. Choose System > Reports > Settings.
2. When the Settings tab opens, enter the number of days you want Steel-Belted Radius to retain log file in the **Days to keep server log file** field.

Figure 239: Reports Page: Settings Tab



Days to keep server log file: 365

Using the Server Log File

The server log file records RADIUS events, such as server startup or shutdown or user authentication or rejection, as a series of messages in an ASCII text file. Each line of the server log file identifies the date and time of the RADIUS event, followed by event details. You can open the current log file while Steel-Belted Radius is running.

Server log files are kept for the number of days specified in the Settings tab in the Reports page (described in "Configuring the Log Retention Period") and then deleted to conserve disk space.

Optionally, you can specify a maximum size for a server log file by entering a non-zero value for the LogfileMaxMBytes setting in the [Configuration] section of the radius.ini file.

- If a maximum file size is set, the server log filename identifies the date and time it was opened (YYYYMMDD_HHMM.log). When the current server log file approaches the specified number of megabytes (1024 x 1024 bytes), the current log file is closed and a new one is opened. The closed file will be slightly smaller than the specified maximum file size.
- If the maximum file size is set to 0 (or if the LogfileMaxMBytes setting is absent), the server log file

size is ignored and log file names are datestamped to identify when they were opened (YYYYMMDD.log).

ⓘ Note: The size of the log file is checked once per minute, and the log file cannot roll over more than once a minute. The log file may exceed the specified maximum file size temporarily (for less than a minute) after it passes the LogfileMaxMBytes threshold between size checks.

By default, server log files are located in the RADIUS database directory. You can specify an alternate destination directory in the [Configuration] section of the radius.ini file.

Level of Logging Detail

You can control the level of detail recorded in server log files by use of the LogLevel, LogAccept, and LogReject settings.

The LogLevel setting determines the level of detail given in the server log file. The LogLevel can be the number 0, 1, or 2, where 0 is the least amount of information, 1 is intermediate, and 2 is the most verbose. The LogLevel setting is specified in the [Configuration] section of radius.ini and in the [Settings] sections of .aut files.

The LogAccept and LogReject flags allow you to turn on or off the logging of Access-Accept and Access-Reject messages in the server log file. These flags are set in the [Configuration] section of radius.ini: a value of 1 causes these messages to be logged, and a value of 0 causes the messages to be omitted. An Accept or Reject is logged only if LogAccept or LogReject, respectively, is enabled and the LogLevel is "verbose" enough for the message to be recorded.

The TraceLevel setting specifies whether packets should be logged when they are received and being processed, and what level of detail should be recorded in the log.

If you alter the LogLevel or TraceLevel settings, you can have them take effect without restarting the server by issuing the following command:

- Linux: Enter the kill -HUP pid command.
- Windows: Issue the radhup command.

Using the Authentication Log File

The authentication log file records each RADIUS authentication request received by Steel-Belted Radius. Authentication log files are Comma Separated Value (CSV) ASCII text files that can be imported into a spreadsheet or database program.

Authentication log files are located in the RADIUS database directory area by default, although you can specify an alternate destination directory in the [Configuration] section of authlog.ini. Authentication log files are named yyyymmdd.authlog, where yyyy is the 4-digit year, mm is the month, and dd is the day on which the log file was created.

Authentication log files are kept for the number of days specified in the Settings tab of the Reports page, and are deleted after that.

The current log file can be opened while Steel-Belted Radius is running.

Authentication Log File Format

The first five fields in every authentication log entry are required by Steel-Belted Radius:

- **Date—**The date when the event occurred
- **Time—**The time when the event occurred

- **RAS-Client—**The name or IP address of the RADIUS client sending the authentication request
- **Full-Name—**The fully distinguished name of the user, based on the authentication performed by the RADIUS server
- **ACC/REJ—**The result of the authentication request (ACCEPT or REJECT)

The RADIUS attributes specified in the authlog.ini file appear next. Attributes in the authlog.ini file beginning with a semicolon (;), are commented out, and their values are not recorded in the authentication log file.

User-Name

NAS-IP-Address

NAS-Port

Service-Type

Framed-Protocol

Framed-IP-Address

Framed-IP-Netmask

Framed-Compression

Login-IP-Host

Callback-Number

State

Called-Station-Id=

Calling-Station-Id=

NAS-Identifier=

Proxy-State=

Login-LAT-Service

Login-LAT-Node

Login-LAT-Group

Event-Timestamp

NAS-Port-Type

Port-Limit

Login-LAT-Port

Note: If the User-Password attribute is included in the authlog.ini file, it is ignored during processing to prevent exposing users' cleartext passwords in the log file.

You can include vendor-specific attributes if the device sending the authentication packet supports them. For more information, see "Vendor-Specific Attributes".

You can edit the authlog.ini file to add, remove, or reorder the standard RADIUS or vendor-specific attributes that are logged. For information on authlog.ini, refer to the *Steel-Belted Radius Reference Guide*.

First Line Headings

The first line of the authentication log file lists the names of all the attributes that have been enabled for logging, in the order in which they are logged. This first line serves as a complete set of column headings for the remaining entries in the file.

The content of the first line depends on the attributes specified in the authlog.ini file. The following example shows the heading line and an authentication log file entry consisting of the required attributes.

```
        “Date”,”Time”,”RAS-Client”,”Full-Name”,”ACC/REJ”

        “7/3/2003”,”12:11:55”,”RRAS”,”EdisonCarter”,”ACCEPT”,
```

Comma Placeholders

Log entries may not include every attribute listed in the first line of the authentication log file. When Steel-Belted Radius records the event in the authentication log file, it uses a comma “placeholder” to mark empty entries, so that all entries remain aligned with their headings.

For example, the following log entries indicate that Bob’s authentication request was rejected but Alice’s authentication request was accepted. The reported fields include Called-Station-Id, Calling-Station-Id, and Port-Limit. Note that the attributes listed in the log heading that were not returned for the authentication events are separated with commas.

```
“Date”,”Time”,”RAS-Client”,”Full-Name”,”Acc/Rej”,

“User-Name”,”NAS-IP-Address”,”NAS-Port”,”Service-Type”,

“Framed-Protocol”,”Framed-IP-Address”,      “Framed-IP-Netmask”,”Framed-Compression”,

“Login-IP-Host”,”Callback-Number”,”State”,   “Called-Station-Id”,”Calling-Station-Id”,

“NAS-Identifier”,”Proxy-State”,     “Event-Timestamp”,”NAS-Port-Type”,”Port-Limit”,

“Login-LAT-Port””07/14/2003”,”13:39:10”,”192.168.2.42”,

“BOB”,”REJECT”,,,,,,,,,,,,,”Alice’s    Office”,”Bob’s

Office”,,,,,”5”,”07/14/2003”,”13:43:26”,”192.168.2.42”,

“ALICE”,”ACCEPT”,,,,,,,,,,,,,”Bob’s Office”,”Alice’s Office”,,,,,”5”,
```

Using the Accounting Log File

RADIUS accounting events are recorded in the accounting log file. Accounting events include START messages, which indicate the beginning of a connection; STOP messages, which indicate the termination of a connection; and INTERIM messages, which indicate a connection is ongoing.

Accounting log files use comma-delimited, ASCII format, and are intended for import into a spreadsheet or database program. Accounting log files are located in the RADIUS database directory area by default, although you can specify an alternate destination directory in the [Configuration] section of the account.ini file. Accounting log files are named yyyymmdd.ACT, where yyyy is the 4-digit year, mm is the month, and dd is the day on which the log file was created.

Accounting log files are kept for the number of days specified in the Settings tab of the Reports page, and are deleted after that to conserve disk space.

The current log file can be opened while Steel-Belted Radius is running.

Accounting Log File Format

The first six fields in every accounting log entry are provided by Steel-Belted Radius for your convenience in reading and sorting the file:

- Date—The date when the event occurred
- Time—The time when the event occurred
- RAS-Client—The name or IP address of the RADIUS client sending the accounting record
- Record-Type—START, STOP, INTERIM, ON, or OFF, the standard RADIUS accounting packet types
- Full-Name—The fully distinguished name of the user, based on the authentication performed by the RADIUS server
- Auth-Type—A number that indicates the class of authentication performed:

  0—Native

  10—SecurID User 11—

  SecurID Prefix 12—

  SecurID Suffix 13—

  TACACS+ User 16—

  TACACS+ Prefix 17—

  TACACS+ Suffix 100—Tunnel

  User 200—External

  Database (other)—Proxy

By default, the standard RADIUS attributes follow the Auth-Type identifier. See "Standard RADIUS Accounting Attributes".

You can include vendor-specific attributes if the device sending the accounting packet supports them. For more information, see "Vendor-Specific Attributes".

You can edit the account.ini initialization file to add, remove or reorder the standard RADIUS or vendor-specific attributes that are logged. For information on account.ini, refer to the ***Steel-Belted Radius Reference Guide***.

First Line Headings

The first line of the accounting log file is a file header that lists the attributes that have been enabled for logging in the order in which they are logged. The following example of a first line shows required headings in bold italic, standard RADIUS headings in bold, and vendor-specific headings in regular text:

"Date","Time", "RAS-Client", "Record-Type", "Full-Name", "Auth-Type", "User-Name",

"NAS-Port", "Acct-Status-Type", "Acct-Delay-Time", "Acct-Input-Octets",

"Acct-Output-Octets", "Acct-Session-Id",   "Acct-Authentic","Acct-Session-Time",

"Acct-Input-Packets",   "Acct-Output-Packets",   "Acct-Termination-Cause",

"Acct-Multi-Session-Id",   "Acct-Link-Count","Acc-Err-Message",

"Nautica-Acct-SessionId","Nautica-Acct-Direction", “Nautica-Acct-CauseProtocol”,”Nautica-Acct-CauseSource”,

“Telebit-Accounting-Info”,”Last-Number-Dialed-Out”,

“Last-Number-Dialed-In-DNIS”,”Last-Callers-Number-ANI”,

“Channel”,”Event-Id”,”Event-Date-Time”, “Call-Start-Date-Time”,”Call-End-Date-Time”,

“Default-DTE-Data-Rate”,”Initial-Rx-Link-Data-Rate”,

“Final-Rx-Link-Data-Rate”,”Initial-Tx-Link-Data-Rate”,

“Final-Tx-Link-Data-Rate”,”Sync-Async-Mode”,

“Originate-Answer-Mode”,”Modulation-Type”,

“Equalization-Type”,”Fallback-Enabled”,”Characters-Sent”,

“Characters-Received”,”Blocks-Sent”,”Blocks-Received”,

“Blocks-Resent”,”Retrains-Requested”,”Retrains-Granted”,

“Line-Reversals”,”Number-Of-Characters-Lost”,

“Number-of-Blers”,”Number-of-Link-Timeouts”,

“Number-of-Fallbacks”,”Number-of-Upshifts”,

“Number-of-Link-NAKs”,”Back-Channel-Data-Rate”,

“Simplified-MNP-Levels”,”Simplified-V42bis-Usage”, “PW_VPN_ID”

Comma Placeholders

Steel-Belted Radius writes accounting events to the accounting log file, If an event recorded in the accounting log file does not have data for every attribute, a comma “placeholder” marks the empty entry, so that all entries remain correctly aligned with their headings. For example, based on the “first line” of headings described above, the following is a valid accounting log entry, in which the value of the Acct-Status-Type attribute is 7:

“12/23/1997”,”12:11:55”,”RRAS”,”Accounting-On”,

,,,,7,,

,,,,,,,,,,,,,,,,,,

Standard RADIUS Accounting Attributes

Table 58 lists the standard RADIUS accounting attributes defined in RFC 2866, “RADIUS Accounting.”

Table 58: Standard RADIUS Accounting Attributes

User-Name	The name of the user as received by the client.
NAS-Port	The port number on the client device.

Acct-Status-Type	A number that indicates the beginning or ending of the user service: 1—Start 2—Stop 3— Interim-Acct 7— Accounting-On 8— Accounting-Off
Acct-Delay-Time	Indicates how many seconds the client has been trying to send this record, which can be subtracted from the time of arrival on the server to find the approximate time of the event generating this request.
Acct-Input-Octets	Number of octets (bytes) received by the port over the connection; present only in STOP records.
Acct-Output-Octets	Number of octets (bytes) sent by the port over the connection; present only in STOP records.
Acct-Session-Id	Identifier used to match START and STOP records in a log file.
Acct-Authentic	Indicates how the user was authenticated by RADIUS, the network access device (local), or another remote authentication protocol: 1—RADIUS 2—Local 3—Remote
Acct-Session-Time	Elapsed time of connection in seconds; present only in STOP records.
Acct-Input-Packets	Number of packets received by the port over the connection; present only in STOP records.
Acct-Output-Packets	Number of packets sent by the port over the connection; present only in STOP records.
Acct-Termination-Cause	Number that indicates how the session was terminated; present only in STOP records: 1—User Request 2—Lost Carrier 3— Lost Service 4—Idle Timeout 5—Session Timeout 6—Admin Reset 7—Admin Reboot 8—Port Error 9—NAS Error 10—NAS Request 11—NAS Reboot 12—Port Unneeded 13—Port Preempted

	14—Port Suspended 15—Service Unavailable 16—Callback 17—User Error 18—Host Request
Acct-Multi-Session-Id	Unique accounting identifier to make it easy to link together multiple related sessions in a log file.
Acct-Link-Count	The count of links that are known to have been in a given multi-link session at the time the accounting record is generated.

Appendix A
Glossary

802.1X	The IEEE 802.1X standard defines a mechanism that allows a supplicant (client) to connect to a wireless access point or wired switch (authenticator) so that the supplicant can provide authentication credentials that can be verified by an authentication server.
AAA	Authentication, authorization, and accounting.
accounting	The process of recording and aggregating resource use statistics and log files for a user, connection session, or function for billing, system diagnosis, and usage planning.
agent	SNMP module on a managed device that responds to requests from a management station and sends traps to one or more recipients (trap sinks) to inform administrators of potential problems.
AP	Access Point. A device that serves as a communication hub to connect 802.1X wireless clients to a wired network.
attribute	RADIUS attributes carry the specific authentication, authorization, and accounting.
authentication	The process of verifying the identity of a person or file system and whether the person is allowed on a protected network.
authentication server	A back-end database server that verifies, from the credentials provided by an access client, whether the access client is authorized to use network resources.
authorization	The process of controlling the access settings, such as privileges and time limits that the user can exercise on a protected network.
AVP	Attribute-value pair. An attribute and its corresponding value.; for example, **User-Name = admin**.
blacklist	A profile of checklist attributes that cause Steel-Belted Radius to reject an authentication request. For example, a blacklist profile might specify calling station phone numbers or IP addresses that are blocked by Steel-Belted Radius.
CA	Certificate authority. A trusted entity that registers the digital identity of a site or individual and issues a digital certificate that guarantees the binding between the identity and the data items in a certificate.
CCM	Centralized configuration management. The process by which information is shared between a primary RADIUS server and one or more replica RADIUS servers in a multi-server environment.
certificate	A digital file signed by a CA that guarantees the binding between an identity and the contents of the certificate.
CHAP	Challenge Handshake Authentication Protocol. An authentication protocol where a server sends a challenge to a requestor after a link has been established. The requestor responds with a value obtained by executing a hash function. The server verifies the response by calculating its own hash value: if the two hash values match, the authentication is acknowledged.
checklist	A list of attributes that must accompany a request for connection before the connection request can be authenticated.
CIDR	Classless Inter-Domain Routing. In CIDR notation, an IP address is represented as A.B.C.D/n, where /n identifies the IP prefix or network prefix). The IP prefix identifies the number of significant bits used to identify a network. For example, 192.168.1.22/18 means “use the first 18 bits to represent the network and the remaining 14 bits to identify hosts.” Common prefixes are /8 (Class A network), /16 (Class B network), /24 (Class C network), and /32.
community	An SNMP community is a group of devices and management stations running SNMP. An SNMP device or agent may belong to more than one SNMP community.

community string

Character string included in SNMP messages to identify valid sources for SNMP requests and to limit access to authorized devices.

- The read community string allows an SNMP management station to issue Get and GetNext messages.
- The write community string allows an SNMP management station to issue Set messages.

credentials

Data that is verified when presented to an authenticator, such as a password or a digital certificate.

CRL

Certificate Revocation List. A data structure that identifies the digital certificates that have been invalidated by the certificates' issuing CA prior to their expiration date.

daemon

A program on a Linux host that runs continuously to handle service requests.

dictionary

Text file that maps the attribute/value pairs supported by third-party RADIUS vendors.

DHCP

Dynamic Host Configuration Protocol. Protocol by which a server automatically assigns (leases) a network address and other configuration settings to a client temporarily or permanently.

DNIS

Dialed number identification service. A telephone service that identifies what number was dialed by a caller.

DNS

Domain Name Service. Internet protocol for mapping host names, domain names, and aliases to IP addresses.

EAP

Extensible Authentication Protocol. An industry-standard authentication protocol for network access that acts as a transport for multiple authentication methods or types. Defined by RFC 2284.

EAP-15

A simple one-time password EAP method.

EAP-32

See POTP.

EAP-TLS

Authentication method that uses EAP (Extensible Authentication Protocol) and TLS (Transport Layer Security).

EAP-TTLS

Authentication method that uses EAP (Extensible Authentication Protocol) and TTLS (Tunneled Transport Layer Security).

GTC

Generic Token Card.

IEEE

Institute of Electrical and Electronics Engineers.

IETF

Internet Engineering Task Force. Technical subdivision of the Internet Architecture Board that coordinates the development of Internet standards.

IPv4

Implementation of the TCP/IP suite that uses a 32-bit addressing structure.

IPv6

Implementation of the TCP/IP suite that uses a 128-bit addressing structure.

Java

Programming language designed for use in distributed environments such as the Internet.

JDBC

Java Database Connectivity. Application programming interface for accessing a database from programs written in Java.

LCI

LDAP configuration interface.

LDAP

Lightweight Directory Access Protocol. An IETF standard protocol for updating and searching directories over TCP/IP networks.

LDIF

LDAP Data Interchange Format. The format used to represent directory server entries in text form.

LEAP

Lightweight Extensible Authentication Protocol.

MAC

(1) Message Authentication Code. A MAC function takes a variable-length input and a key to produce a fixed-

	length output to carry authentication and integrity protection of data. (2) Media Access Control. The unique hardware address associated with a computer network interface.
managed device	A device that runs an SNMP agent.
management station	Host that monitors and controls managed devices running SNMP agents.
MIB	Management Information Base. A database of objects, such as alarm status or statistics counters, that can be monitored or overwritten by an SNMP management station.
MPPE	Microsoft Point-to-Point Encryption. A means of representing point-to-point packets in an RC4 encrypted format. Defined in RFC 3078.
MS-CHAP	Microsoft CHAP. Proprietary version of CHAP.
NAS	Network Access Server. Network device that accepts connection requests from remote users, authenticates users via RADIUS, and routes user onto the network. Identical in meaning to RAS.
NAT	Network Address Translation. Technique that allows an intranet to use IP addresses that are different from what the Internet recognizes
native user	A user authenticated by Steel-Belted Radius using its internal authentication database.
ODBC	Open Database Connectivity. Standard (open) application programming interface for accessing a database.
OTP token	One-time password token. Hardware or software module that generates one-time passwords that can be used to authenticate a user.
PAC	Protected Access Credential. A high-entropy secret that is known to both the RADIUS client and the RADIUS server to secure the TLS handshake.
PAP	Password Authentication Protocol. An authentication protocol where a requestor sends an identifier and password to a server after a link has been established. If the identifier and password match an entry in the server's database, the authentication is acknowledged.
PEAP	Protected Extensible Authentication Protocol. A two-phase authentication protocol where (1) an authentication server is authenticated to a supplicant using a digital certificate and a secure channel is established; and (2) the supplicant is authenticated to the authentication server via the secure channel.
POTP	Protected One-Time Password. EAP method that uses one-time password tokens for unilateral or mutual authentication.
PPP	Point-to-Point Protocol. Network protocol defined in RFC 1661 that provides a standard method for transporting multi-protocol datagrams over point-to-point links.
proxy RADIUS	Process of authenticating users whose profiles are on other RADIUS servers by forwarding access-request packets received from a RADIUS client to a remote RADIUS server (the proxy target), and then forwarding the response from the remote server back to the RADIUS client.
proxy target	The remote RADIUS server that actually performs authentication in a proxy RADIUS sequence.
RADIUS	Remote Authentication Dial In User Service. A client/server security administration standard that functions as an information clearinghouse, storing authentication information about users and administering multiple security systems across complex networks.
RAS	Remote Access Server. Network device that accepts connection requests from remote users, authenticates users via RADIUS, and routes user onto the network. Identical in meaning to NAS.
return list	A list of attributes that Steel-Belted Radius must return to a RADIUS client after authentication of a user succeeds. The return list usually provides additional parameters that the RADIUS client needs to complete the connection.
roaming	The ability to move from one Access Point coverage area to another without interruption of service or loss of connectivity.

RSA SecurID	Security token system that allows remote-access users to generate a pseudorandom value they can forward as part of an authentication sequence.
session ID	Session Identifier. A string of characters uniquely identifying the session.
SHA-1	Secure Hash Algorithm-1. A one-way cryptographic function that takes a message of any length and produces a 160-bit message digest.
shared secret	An encryption key known only to the sender and receiver of data.
silent discard	The process of discarding a packet without further processing and without notification to the sender.
SNMP	Simple Network Management Protocol.
SSL	Secure Sockets Layer. Program layer that manages the security of messages on a network.
supplicant	The client in an 802.1X-authenticated network.
TACACS+	Terminal Access Controller Access Control System (with enhancements). An authentication protocol that allows a RAS to communicate with an authentication server to determine if a user should have access to a protected network.
TLS	Transport Layer Security.
trap	An SNMP message that reports a significant event, such as a problem, error, or change in state, that occurred within a managed device.
trap sink	The destination for trap messages sent by an SNMP agent on a managed device.
TTLS	Tunneled Transport Layer Security.
user database	A database where a RADIUS server keeps information about users, such as authentication information and network access permissions.
user profile	A record in the user database that describes how a particular user or class of users should be configured during authentication and authorization.
VSA	Vendor Specific Attributes.
WEP	Wired Equivalent Privacy. An encryption method designed to encrypt traffic between a WLAN client and an access point.
WLAN	Wireless Local Area Network.

Appendix B
When to Restart Steel-Belted Radius

This appendix explains the following topics when to stop and restart Steel-Belted Radius.

The *least* drastic action that causes this change to take effect is indicated by **Yes** in this table:

Table 59: When to Stop and Restart Steel-Belted Radius

Item changes:	Save the window or file	Issue a HUP signal	Stop/restart the server
Access window or object	Yes	Also works)	(Also works)
access.ini file	No	No	Yes
*.acc files	No	No	Yes
account.ini file	No	No	Yes
admin.ini file	No	No	Yes
*.aut files	No	(Sometimes)	Yes
blacklist.ini file	No	No	Yes
bounce.ini file (Windows only)	No	No	Yes
Authentication policy	Yes	(Also works)	(Also works)
*.dct files	No	No	Yes
*.dhc files	No	No	Yes
dhcp.ini file	No	No	Yes
*.dir files (see Notes below)	No	(Sometimes)	Yes
*.eap files	No	(Sometimes)	Yes
eap.ini file	No	No	Yes
events.ini file	No	No	Yes

Item changes:	Save the window or file	Issue a HUP signal	Stop/restart the server
filter.ini file	No	Yes	(Also works)
Import *.rif or users file	Yes	(Also works)	(Also works)
*.ini for directed accounting	No	No	Yes
IP Pools dialog or object	Yes	(Also works)	(Also works)
IPX Pools dialog or object	Yes	(Also works)	(Also works)
lockout.ini file	No	No	Yes
Log levels (in radius.ini file)	No	Yes	(Also works)
Profiles dialog or object	Yes	(Also works)	(Also works)
Proxy dialog or object	Yes	(Also works)	(Also works)
*.pro files	No	Yes	(Also works)
proxy.ini file (see Notes below)	No	(Sometimes)	Yes
radius.dct file (see Notes below)	No	No	Yes
radius.ini file	No	No	Yes
RADIUS Clients dialog or object	Yes	(Also works)	(Also works)
Servers dialog or object	Yes	(Also works)	(Also works)
services file	No	No	Yes
tacplus.ini file	No	No	Yes
TLS and TTLS	No	Yes	(Also works)
Trace levels	No	Yes	(Also works)
Tunnels panel or object	Yes	(Also works)	(Also works)

Item changes:	Save the window or file	Issue a HUP signal	Stop/restart the server
Users dialog or object	Yes	(Also works)	(Also works)
vendor.ini file	No	No	Yes

Appendix C
Technical Notes

This appendix contains the following technical bulletins:

- LDAP Support for Novell eDirectory
- Service Type Mapping
- CCA Support for 3COM
- Ascend Filter Translation
- ldapauth Extensions
- Ericsson's e-h235 Authentication Protocol
- Uniport Plug-In
- Windows Performance Monitor

LDAP Support for Novell eDirectory

The Steel-Belted Radius LDAP authentication plug-in contains features to enable greater interoperability with Novell eDirectory.

If you have configured eDirectory to limit the number of grace logins available to a user, Steel-Belted Radius can be configured to coordinate with eDirectory on this feature. Each time a user authenticates, the number of grace logins available is decremented until the account is locked out and needs an administrator to unlock it. A profile is assigned to these users — a profile that overrides their normal profile — when they are being authenticated using a grace login.

The features include:

- *Allowing Expired Accounts*: When enabled, this feature allows users to log in even after their account has expired.

  ⓘ Note: As eDirectory itself does not check the password, this feature should be used only if the administrator has configured the ProfileForExpiredUsers setting to assign an alternate profile to the user, one that would inform the user of their account status but not allow a usable connection to the network. For example, you can use http redirection to force the connection to a web page with relevant information.

  ⓘ Note: Administrators should contact their NAS vendors to determine the capabilities of their NAS equipment and what attribute-value pairs would be needed to create such a connection.

- *Grace Logins:* When grace logins are limited in eDirectory, Steel-Belted Radius can be configured to accept or reject a user whose password has expired. This user is said to be in grace login mode; the user can also be allowed to log in but is provided with an alternate profile.

  ⓘ Note: The grace login feature requires NetWare 6.0 or later with eDirectory 8.6 or later.

- *BindName*: You can use the BindName technique to search the eDirectory directory for a matching user and, having retrieved the user's DN, apply the Bind technique to authenticate the user's credentials. This combination allows the user to specify their common name (rather than the more cumbersome DN) when requesting authentication.

Note: This feature works only if the NetWare server accepts LDAP Bind requests for users who are in grace login state. Earlier versions of the NetWare server did not support this feature.

Note: You must configure the eDirectory server to 'Allow Clear_text passwords' to use these LDAP extensions. You can do this from the ConsoleOne application, in the Properties section of the LDAP group entry for your server.

Configuration

The [NDS] section (**Table 60**) has been added to the **ldapauth.aut** file to configure these features.

Table 60: [NDS] Settings

Field	Usage
Enable	Set to 1 to enable the Novell eDirectory (NDS) extensions. Default value is 0.
AllowExpiredAccountsForUsers	Set to 1 to allow users to authenticate even after their account has expired. Default value is 0. **Important**: This requires that the Netware server notify Steel-Belted Radius that the user's account has expired when the Netware server is attempting to Bind.
ProfileForExpiredUsers	The name of the profile to assign a user (as an override) if authenticated with an expired account. If you do not provide a value for this setting, the user is accepted with the usual profile and attributes. We recommend that you provide the user with a profile with restricted access.
AllowGraceLoginsForUsers	Set to 1 if users should be allowed to be authenticated in grace login mode. This decrements the grace login counter, and rejects the user when it has run out. Default value is 1.
ProfileForGraceLoginUsers	The name of the profile to assign a user (as an override) if authenticated in grace login mode. If you do not provide a value for this setting, the user is accepted with the usual profile and attributes.

If Enable is set to 1, Steel-Belted Radius works as follows:

- If the eDirectory directory is configured to operate without grace logins:
  - If **AllowExpiredAccountsForUsers** is set to 0 (the default), users with expired passwords are rejected.
  - If **AllowExpiredAccountsForUsers** is set to 1, users with expired accounts are accepted; the attributes returned in the Access-Accept response are either the attributes normally assigned to the user or, if the **ProfileForExpiredUsers** setting is specified, the attributes specified in that profile. If you enable this feature, we recommend that you configure the

ProfileForExpiredUsers setting.

- If the eDirectory directory is configured to operate with grace logins:
    - If **AllowGraceLoginsForUsers** is set to 0, users with correct but about-to-expire passwords are rejected.
    - If **AllowGraceLoginsForUsers** is set to 1 (the default), users with correct but about-to-expire passwords are accepted; the attributes returned in the Access-Accept are either the regular attributes assigned to the user or, if **ProfileForGraceLoginUsers** is specified, the attributes specified in that profile.

Sample ldapauth.aut file

```
[Bootstrap]
LibraryName=ldapauth.dll
Enable=1
InitializationString=LDAP

[Settings]
MaxConcurrent=1
Timeout=20
ConnectTimeout=25
QueryTimeout=10
WaitReconnect=2
MaxWaitReconnect=360
BindName=uid=<User-Name>,   ou=sales,   o=bigco.com
LogLevel = 0
UpperCaseName = 0
PasswordCase=original
PasswordFormat=0
Search = DoLdapSearch
SSL = 0

[Server]
s1=

[Server/s1]
Host=192.168.5.110
Port = 389

[Request]
%UserName = User-Name

[Response]
%profile= attThatContainsUserProfile

[Search/DoLdapSearch]
;Bind as a privileged user or someone that has the right to
; search the tree and retrieve the DN of users
bind=cn=administrator,o=netware6
Password= support
Base = o=netware6
Scope = 2
```

```
Filter = uid=<User-Name>
%DN = dn
;if the user is found perform search getprofile
onfound = GetProfile
;else reject the user
onnotfound=$reject

[Search/GetProfile]
; bind using the DN retrieved in doLdapSearch
Bind = <dn>
;You do not have to supply the password SBR knows to use the
;one received in the auth request.
;Setting base to the DN is most efficient
Base =<DN>
Scope = 2
Filter = uid=<User-Name>
Attributes = AttrList

[Attributes/AttrList]
attThatContainsUserProfile

[NDS]
Enable=1
AllowExpiredAccountsForUsers=1
ProfileForExpiredUsers=Expired
AllowGraceLoginsForUsers=1
ProfileForGraceLoginUsers=Grace

[Attributes/AttrList]
;Filter-Id
;Session-Timeout
;thepasswordis
```

Service Type Mapping

Service type mapping allows a single user to have multiple authorization attribute sets based on the service type the user is requesting. The service type is determined based on request attributes using rules that may differ depending on the NAS device.

Using static configuration parameters in the **servtype.ini** file, you can specify, on a NAS-by-NAS basis, a mapping of request attributes and/or values to service type strings. These strings can be attached to the username, either as a prefix or as a suffix. The elaborated username is used for both authentication and authorization, and for allowing different authorizations based on service type requested.

Configuration

Service type mapping is configured in the following way:

- Create multiple Local User entries in the database according to specific naming and mapping conventions. For example:

```
ppp:george
vpn:george
ppp:martha
```

isdn:martha

- Define a set of rules in the servtype.ini file mapping each incoming Access-Request packet to the appropriate database entry for the user.

Local User Database Entries

The Local User entries you define to support service type mapping must follow a consistent naming convention. However, you are free to use any convention you like.

For example, you can store entries for PPP users using the convention ppp:username (for example, ppp:george and ppp:martha) and entries for VPN users using the convention vpn:username (vpn:george and vpn:martha).

For the mapping to work, however, you must define users who do not have any of these mapped prefixes or suffixes in the local users database. For example, if you want to map vpn:emil and ppp:emil so that the appropriate profiles would be returned, you could enter three user entries in the local users database:

vpn:emil
ppp:emil
emil

Alternatively, you could omit emil from the local users database, authenticate emil against a non-local method and then apply the mapping. The mapped names would still have to be in the local user database for profiles to be returned.

You can support classes of service by varying the string you use in creating Local User entries. For example, if you offer three classes of VPN service, your VPN entries might use the conventions vpn1:username, vpn2:username, and vpn3:username (vpn3:george and vpn1:martha).

A delimiting character (such as a colon) in your service type string makes your user record names easier to read—for example, vpn:amar instead of vpnamar. When you design a service type string, consider whether it is a prefix (string+separator) or a suffix (separator+string) to the username.

Note: You can define Local User records using the Administration program or the LDAP configuration interface.

servtype.ini File

The servtype.ini configuration file controls service type mapping and contains the following sections:

Table 61: servtype.ini Syntax

servicetype.ini Section	Meanings
[Settings]	Indicates how the service type string should be attached to the username prior to look-up in the Local User database: by prefix, by suffix, or not at all. The two fields **Prefix** and **Suffix** may be enabled (set to 1) or disabled (set to 0) independently of each other. If both are set to 0 (the default) the service type feature is completely disabled. Using this example, if user **george** requests PPP service and the string for that service type is **ppp:**, the Local User record with the return list for this request has the name **ppp:george.**
[NAS]	Allows you to map NAS devices to [mapping] sections. The syntax for [NAS] is as follows: **[NAS]** **NASname = mapping** **NASname = mapping** . .

servicetype.ini Section	Meanings
	. **NASname = mapping** Each NASname in the [NAS] section must match the name of a RADIUS client entry in the database. When an Access-Request is received, its NAS-IP-Address attribute is matched to a RADIUS client entry in the database. If a match can be found, and the RADIUS client name matches a NASname in the [NAS] section, a corresponding [mapping] section will be found.
[mapping]	The following logic is applied to the [mapping] section: 1. The "next" (initially, the "first") **ServiceTypeString** in the [mapping] section is sought. It combines the **ServiceTypeString** with the username as defined in the [Settings] section, and tries to find a matching Local User entry. If a matching entry is found, each rule in the ServiceTypeString section is tried against the attributes in the incoming Access-Request packet. Otherwise (there was no next ServiceTypeString or no match could be found), the user is rejected. 2. In the following example, the rule syntax is: **RADIUSattribute = value** If the RADIUSattribute named is present in the Access-Request packet, and if it has the value shown, this rule is true. Evaluate the next rule. If there is no next rule, select this **ServiceTypeString**. If the RADIUSattribute named is not present in the Access-Request packet, or if it is present but does not have the value shown, then control returns to step 1. 3. In the following example, the rule syntax is: **RADIUSattribute** **NOTE**: The absence of a value is important. If the RADIUSattribute is present in the Access- Request packet, this rule is true. Evaluate the next rule. If there is no next rule, select this ServiceTypeString. If the **RADIUSattribute** is not present in the Access-Request packet, control returns to step 1. 4. In the following example, the rule syntax is: **~RADIUSattribute =value** **NOTE**: The tilde (~) operator. If the RADIUSattribute named is present in the Access- Request packet, and if it does not have the value shown, this rule is true. Evaluate the next rule. If there is no next rule, accept this **ServiceTypeString**. If the **RADIUSattribute** named is not present in the Access-Request packet, or if it is present but has the value shown, control returns to step 1. **NOTE**: The following is not valid syntax: **RADIUSattribute = ~value** 5. In the following example, the rule syntax is: ~RADIUSattribute **NOTE**: The tilde (~) operator and the absence of a value. If the RADIUSattribute named is not present in the Access-Request packet, this rule is true. Evaluate the next rule. If there is no next rule, accept this ServiceTypeString.

servicetype.ini Section	Meanings
	If the RADIUSattribute named is present in the Access-Request packet, control returns to step 1. 6. If no RADIUSattribute rules are provided and a ServiceTypeString section exists, but contains no rules, the ServiceTypeString is selected automatically. If no ServiceTypeString sections are provided and a [mapping] section exists, but is empty, the user is rejected automatically.

In addition to enabling a prefix or suffix, the [Settings] section of the servtype.ini file permits you to specify a default [mapping] section to be used when an Access-Request packet arrives from a NAS device that is not listed in the [NAS] section of servtype.ini. The syntax for setting this default is as follows:

```
[Settings]
Default = mapping
```

The Default field is optional. If you do not set up a default mapping, and the server cannot determine the mapping in any other way, the server ignores the service type and authenticates the user without it.

The following is a sample servtype.ini file:

```
[Settings]
Prefix=1
Suffix=0
Default=defaultmap

[NAS]
nas1=nas1map
nas2=nas2map

[nas1map]
ppp:Framed-Protocol=1
Service-Type=2

vpn:
Framed-Protocol=6
~Service-Type=2

other:
Framed-Protocol
Service-Type

[nas2map]
analog:
NAS-Port-Type=1

isdn:
NAS-Port-Type=2

[defaultmap]
ppp:
```

To permit simple and clear failure cases, any syntax error in the servtype.ini file prevents initialization of the file. If this occurs, service type mapping is disabled. This event is logged in the date.log file.

Routed Proxy

Routed Proxy provides the ability to consult an external database (SQL or LDAP) to determine the routing of an authentication request. This information can be used to pre-authenticate the user, to select a target realm for a subsequent proxy, to modify the User-Name in the proxy request, and to insert attributes into the response.

Normally, a proxy target may be configured statically (Proxy As Authentication Method), or may be determined based on information in the packet, such as NAI (decorated user-name), DNIS, or other computations (attribute mapping). Routed Proxy adds the ability to externalize the dynamic determination of a realm through an LDAP directory or a SQL database.

Typical uses for this feature might be:

- Allowing the User-Name to determine the realm without requiring decoration.
- Centralizing the mapping of attributes to the realm (the attribute mapping feature, but offloaded to LDAP or SQL).
- Allowing the User-Name to be decorated only with a final realm, mapping it to the Next-Hop realm through LDAP. This allows an enterprise to change their ISP without requiring reconfiguration of user PCs. (Also, by storing shared-secret information for the Next-Hop realm in LDAP, Secure Peer Discovery functionality is available.)

Operation

Routed Proxy occurs when certain information is returned from an external database (SQL or LDAP). Operation is governed by the following two variables, accessible by authentication methods through the SQL and LDAP authentication plug-ins:

- If %ProxyRealm is not set, Routed Proxy does not occur.
- If %ProxyRealm is set to a directed realm, then handle it as a directed realm request.
- If %ProxyRealm is set to a proxy realm, then send the request off to that proxy realm.
- %ProxyUserName is set to the User-Name attribute, which must be sent in the proxy request. If %ProxyUserName is not set, the User-Name from the original request packet is used.

Routed Proxy supports RADIUS authentication, including the RADIUS challenge process and RADIUS accounting. Password authentication may happen when the external database is accessed, or later by the proxy target itself, depending on whether the values of the %Password and %ProxyRealm variables returned from the database are blank or non-blank. Note that only one routed proxy is allowed per transaction; that is, they cannot be nested.

Table 62 lists the four cases for Routed Proxy.

Table 62: Routed Proxy – Four Cases

%Password	%ProxyRealm	Action
blank	blank	Authentication fails
blank	non-blank	Authenticate at proxy target
non-blank	blank	Authenticate password now; no proxy

%Password	%ProxyRealm	Action
non-blank	non-blank	Authenticate password now; direct to appropriate realm if successful

CCA Support for 3COM

Steel-Belted Radius can support the generation of 3Com CCA tunnel attributes.

Configuration

To enable the return of the required CCA tunnel attributes, the ccagw.ini file must be modified.

The ccagw.ini file contains information about gateways, which are stored in the [gateway] sections. A [gateway] section must be present for each gateway supported. The following table describes each field:

Table 63: ccagw.ini File

Setting	Meaning
Address	The address of the gateway.
TunnelRefresh	The number of seconds before the tunnel refreshes. The default value is 0.
Description	A text string describing the gateway.
Secret	The shared secret between Steel-Belted Radius and this gateway device.

For example:

```
[Jupiter-Gateway]
Address = 200.47.98.142
TunnelRefresh = 3600
Description = Jersey City facility, East Coast subscribers
Secret = Holland Tunnel
```

Setting User and Profile Attributes

To enable this functionality for a particular user, the return list for the user must contain the following attributes:

Tunnel-Authentication
VPN-Gateway

Both of these attributes are defined as strings. The value of each attribute must be set to the name of the gateway used in the ccagw.ini file. For example, the return list of a user would have to include:

```
Tunnel-Authentication=Jupiter-Gateway
VPN-Gateway=Jupiter-Gateway
```

It is important to make sure that both attributes name the correct gateway. If an unknown gateway is named, the request is rejected.

Note: Steel-Belted Radius is capable of returning multiple pairs of attributes for different gateways. For each gateway named in one of the attributes, a different random session key is generated.

Note: Please see your 3Com documentation for more information.

Ascend Filter Translation

Ascend defines two attributes — Ascend-Data-Filter (242) and Ascend-Call-Filter (243) — that contain structured binary data representing a filter to be applied to the NAS device.

Instead of entering hexadecimal strings to configure these attributes, users can configure these attributes as text strings. Steel-Belted Radius automatically converts the text strings to the proper binary representation. The original filter attributes are still supported, and these attributes still may be configured as hexadecimal strings.

The following attributes allow configuration as text:

Ascend-Data-Filter-String
Ascend-Call-Filter-String

When Steel-Belted Radius formats a response packet, it translates the string version of the attribute to the appropriate binary value, and returns the attribute in the Access-Accept message.

Configuration

These attributes may be entered as text strings through the SBR Administrator. The attributes may also be returned from an LDAP or SQL database during authentication.

No syntax validation is performed when the attribute is configured. The validation of syntax occurs only when the response packet is formatted. If the syntax is invalid, a reject response is issued and an error is logged.

Note: These attributes should be tested before configured on a production server.
Two types of filter are supported: “ip” and “generic”. “ipx” filters are not supported.

Syntax

In the syntax descriptions below, brackets [ ] indicate that the items enclosed are optional.

ip [direction] [action] [srcip address[/mask]] [dstip address[/mask]] protocol [srcport operator port] [dstport operator port]

Table 64: ip and generic Filter Support

Parameter	Values
direction	May be “in” or “out”. The default is “out”.
action	May be “forward” or “drop”. The default is “drop”.
address	An IP address in decimal dotted notation.
mask	The number of bits (decimal) in the network portion, from 0 through 32. The default is based on class of network.

Parameter	Values
protocol	The protocol number (decimal); for example, 6 for TCP or 17 for UDP. The following protocol names are translated to the proper number: • icmp(1) • tcp(6) • udp(17) • ospf(89).
operator	May be = (equal sign), != (exclamation and equal sign), < (less than), or > (greater than).
port	The port number (decimal). In addition, the following service names are translated to the proper port number: • ftp-data(20)•www(80) • ftp(21)•kerberos(88) • telnet(23)•hostname(101) • smpt(25)•nntp(119) • nameserver(42)•ntp(123) • domain(53)•exec(512) • tftp(69)•login(513) • gopher(70)•cmd(514) • finger(79)•talk(517)

Example:

ip out forward srcip 10.1.1.0/24 6 dstport = 80 srcport < 1023

Note: See your Ascend documentation for details about the syntax for these attributes.

ldapauth Extensions

This plug-in adds the ability to use two new LDAP attributes:

- ProfileData: Stores multiple RADIUS attribute value pairs within a single LDAP container, removing the need for multiple entries in a LDAP user object.
- GlobalProfile: Configures users based on a global profile, which can be specified as any username concatenated with the company name (such as Profile1@Company).

GlobalProfile Attribute

The GlobalProfile attribute takes the value from an LDAP attribute and parses it to match a profile. The format of the data is that of a DN attribute and should be stored as:

The GlobalProfile attribute takes the value from an LDAP attribute and parses it to match a profile. The format of the data is that of a DN attribute and should be stored as:

cn=profile-name, {optional ou's}, o=name,
{optional dc's \o's \c's}
profile-name and name are concatenated to build profilename@name. This value should then match a profile stored in Steel-Belted Radius.

For example:

cn=Global1, ou=Profile, ou=Radius, ou=IP Services,
o=acme, o=directoryroot
This value is parsed to form a new string: Global1@acme. This new string is then passed back as the profile by making the following entry in the response section:

[Response]
%profile= LDAP attribute that contains the global profile
This value, however, is ignored if:

- There is no o keyword value
- The string does not begin with the cn keyword
- %profile is not set to the name of the attribute that contains the Globalprofile data

An incorrect profile name results if the name parameter isn't the first value of the organization name (o).

ProfileData Attribute

This feature allows an administrator to store multiple RADIUS attribute-value pairs within a single LDAP container, removing the need for multiple entries in a LDAP user object.

For example, the values for framed-ip-address, service-type, and framed-protocol could all be stored in one attribute called stdDialin. Combining them saves space on the LDAP server.

The attribute should be of a string data-type (directory string or string case insensitive). The format for the data stored in this attribute is:

<r|R>;attribute-name;type;value&

- r or R specifies that the attribute may be single or multi-valued.
- attribute-name: Specifies the name of the attribute that is being added.
- value&: The value to be returned with this attribute, terminated with &.

Note: The type field is ignored (the values are interpreted according to the RADIUS dictionary).
For example:

stdDialin: r; service-type; integer; 1&;r; framed-protocol; integer; 2&; r;
framed-ip-address; string;192.168.2.2&

The Profiledata attribute is retrieved from the LDAP server in the same way in which other attributes are retrieved; they might be specified from the [Attributes\] section referenced in the relevant search.

The [Response] section of the ldapauth.aut file should list each attribute contained in the profiledata attribute.

The [Response] section should be configured as follows for Std_dial to operate:

[Response]
service-type=
framed-protocol=

framed-ip-address=

Note: If the ProfileData attribute stores multiple attribute-value pairs and one or more of those attributes appears in the applicable dictionary, then that attribute and its value are returned to the RADIUS client even if the attribute is not enumerated in the [Response] section of the file.

Modifying ldapauth.aut

The following explains how to modify ldapauth.aut to support the extensions:

1. Add the following field: [Settings] UpdateResponse = 1

2. Add a [GroupedAttributes] section to specify the GlobalProfile and/or ProfileData attributes.

```
[GroupedAttributes]
GlobalProfile = GlobalProfileLDAPAttribute
ProfileData = ProfileDataLDAPAttribute
```

3. In the appropriate [Attributes/name] section, add the actual LDAP attributes as specified above.

```
[Attributes/name]
GlobalProfileLDAPAttribute
ProfileDataLDAPAttribute
any other attributes
```

4. In the [Response] section, set %Profile to the GlobalProfile and list any attributes that are contained in the ProfileData attribute:

```
[Response]
%Profile = GlobalProfileLDAPAttribute
radiusattribute1=
radiusattribute2=
```

Ericsson Enhanced Token Caching

This section pertains only to Ericsson equipment that supports enhanced token caching.

Enhanced token caching allows the administrator to specify that particular users are authenticated with both an ordinary password and a SecurID passcode. For such users, the ordinary PAP or CHAP password is checked first. If this first authentication is successful, the user's SecurID passcode is authenticated. Only if both authentications succeed is the user allowed access.

Note: The enhanced token caching feature does not interact in any way with the ordinary token caching feature described in the above section. Enhanced token caching is required to support the newer firmware releases on Ericsson devices.

Enhanced Token Caching Configuration

To enable enhanced token caching, a file with extension .aut must be included in the server directory—typically named sidalt.aut. The file has the following format:

```
[Bootstrap]
LibraryName = sidalt.dll
Enable = { 0 | 1 }
InitializationString = string
```

```
[Settings]
TokenAttr = string
CacheTimeoutAttr = string MessageID
= { 0 | 1 } ChallengeTokenInPassword
= { 0 | 1 }
```

Table 65 describes each field in the sidalt.aut file.

Table 65: sidalt.aut Syntax

Sidalt.aut Field	Meaning
Enable	• Set to 0 to disable. • Set to 1 to enable. Default value is 0.
LibraryName	Identifies the dynamic link library used to process SecurID authentication. Default value is sidalt.dll.
InitializationString	Identifies the enhanced token caching software component for logging purposes. A typical value might be SecurID Alt. This field is required.
TokenAttr	The name of the Access-Request attribute containing the passcode or other information to passed to the RSA SecurID server. This attribute must match the corresponding dictionary (.dct file) entry. This field is required.
CacheTimeoutAttr	The name of the Access-Response attribute containing the number of seconds a passcode remains in the cache from the time it was first validated by the RSA SecurID server. This attribute must match the corresponding dictionary (.dct file) entry. This field is required.
MessageID	Controls the format of Reply-Message attribute response packets, which are used to prompt the user for information during a challenge, and to inform the user of results. • If set to 0, the Reply-Message attribute consists only of a text message. • If set to 1, the Reply-Message attribute consists of a 1-byte message ID followed by a text message. Default value is 0.
ChallengeTokenInPassword	Allow test clients to interpolate with the enhanced token caching plug-in. • If set to 0, passcode or other information entered by the user as a response to a challenge appears in the TokenAttr entry. • If set to 1, passcode or other information entered by the user as a response to a challenge appears in the User-attribute, rather than in the attribute specified by the TokenAttr entry.

Enhanced Token Caching Administration

To authenticate a user through the enhanced token caching component, the following must be true:

- The attribute specified by the TokenAttr entry must be present in the Access-Request.
- The return list specified for the user, either directly or through a profile, must include the attribute specified by the CacheTimeoutAttr entry.

If either attribute is not supplied, the user is assumed not to require enhanced token caching authentication, and is accepted.

Note: See your Ericsson product documentation for information about NAS and PPP client operation under the enhanced token caching authentication method.

Ericsson's e-h235 Authentication Protocol

Steel-Belted Radius now supports Ericsson's implementation of the h-235 Authentication Protocol.

Note: This protocol is not a complete or strict implementation of the standard and should be used only with Ericsson equipment that supports this feature.

Operation

The user has a password, which is also known to the server. The user sends a name, timestamp, and 20-byte hash to the AAA server. The server validates this digest as follows:

1. A 20-byte key is computed from the password. If the password is less than 20 bytes, the key is set to the password with null padding. If the password is greater than 20 bytes, the key is set to the first 20 bytes of the password XORed with the next 20 bytes and so on in wrap-around fashion, until you come to the end of the password.

2. The 20-byte hash sent by the user is computed via HMAC-SHA1, where the key is the 20-byte key described previously, and the input is the username concatenated with the low-order byte of the timestamp.

This authentication scheme is operative if User-Password (PAP) is active and if the TimeStamp field in the [e-h235] section of radius.ini has been set.

Configuration

The protocol is configured by specifying the attribute that carries the timestamp in the new [e-h235] section of radius.ini:

```
[e-h235]
TimeStamp = Integer-Attribute
```

Uniport Plug-In

3Com's Uniport project can operate with Steel-Belted Radius via a plug-in.

Operation

Uniport requires RADIUS call-type determination as a back-up for SIP call-type determination. To determine call-type, the HiPerARC system sends Steel-Belted Radius a request containing a Service-Type attribute of Call-Check (10) and a User-Name attribute in which the value is the same as the Called-Station-Id (DNIS) attribute. The type of call is then determined based on the User-Name (DNIS), and the appropriate Service-Type attribute returned in the Access-Accept packet.

A Uniport plug-in method is instantiated for each value of the Service-Type attribute which can be returned in the Access-Accept. The proper method is utilized using proxy mapping to a directed realm which specifies the method instance. The method then sets the configured profile in the response and indicates it was successful.

The Uniport methods return a Reject if a Service-Type attribute with a Call-Check value is not present in the request, if User-Name or Called-Station-Id attributes are not present, or if their values are not identical.

Configuration

The attribute(s) to be returned in the Access-Accept to identify the call-type are defined as Steel-Belted Radius profiles. For example, a FAX profile may be created which would return a value of 96 in a Service-Type attribute.

The Uniport methods are configured with *.AUT files which specify the profile to return. The method is identified and associated with a directed realm by the initialization string.

For example, a method to return the FAX profile may use a configuration file such as FAX.AUT, which would have the following settings:

```
[Bootstrap]
Enable = 1
InitializationString = UNIPORT FAX
Profile = FAX
LibraryName = Uniport.so
```

The corresponding directed realm would then identify the method in its *.DIR file. For example, in the FAX.DIR file the settings might be:

```
[Auth]
Enable = 1

[AuthMethods]
Uniport fax
```

The directed realms which refer to the Uniport methods are mapped in the [AuthAttributeMap] of PROXY.INI. A sample map might appear as:

```
[AuthAttributeMap]
Fax
Service-Type = 10
Called-Station-Id = 6175471047
FoIP
Service-Type = 10
Called-Station-Id = 617*
VoIP
Service-Type = 10
```

In the preceding example, the number 6175471047 would be directed to the Fax realm, which in turn would use the Uniport method which returns the Fax profile. Similarly, 6174976339 would result in a FoIP type and 5085551234 in a VoIP type (as the default).

Note: Wildcards must be listed after any numbers that they might "contain." In the preceding example, the 617* wildcard must appear after the number 6175471047, as this last number would be contained within the range of numbers described by the wildcard.

If you want a default profile, the map may be configured to direct the request to a Uniport method by default. The same result may be obtained by omitting the default from the map and setting the first method in the authentication method chain to the desired default.

If you do not want a default profile, configure the map to direct requests by default to a method which has no profile set; the method returns with a value to indicate a failure to authenticate.

Windows Performance Monitor

The Steel-Belted Radius service has information which can be viewed with the Performance Monitor on a

Windows administrative workstation or server. You can start multiple instances of the Windows Performance Monitor to display performance statistics graphs for more than one Steel-Belted Radius server simultaneously.

To view a graph of Steel-Belted Radius performance:

1. Run the Windows performance monitor by choosing **Start > Control Panel > Administrative Tools > Performance**.

2. Click the + button in the Performance window (or press CTRL+I).

The Add Counters window opens.

3. Pull down the **Performance Object** list and choose **Steel-Belted Radius**.

If you are running multiple Steel-Belted Radius servers, select the one you want to monitor from the **Select counters from computer** list.

4. If you want to select the counters that you want to graph, click the **Select counters from list** radio button, choose the counter you want, and click **Add**. Continue choosing counters and clicking Add until all desired counters have been selected.

If you want to display all counters for Steel-Belted Radius, click the **All counters** radio button and click **Add**.

Most perfmon counters relating to Steel-Belted Radius have self-explanatory names, such as Acct Failures - Insufficient Resources or Acct Failures - Invalid Shared Secret.

Of special interest is the Failed Auths - n counter, where n is a number between 1 and 16. You can set up as many as 16 Failed Auths - n counters, where each counter tracks the number of failed authentication requests that were encountered for all of the RADIUS clients that you have mapped to collection number n.

To set up the Failed Auths - n counter, you must configure the [FailedAuthOriginStats] section of radius.ini. For information on radius.ini, refer to the Steel-Belted Radius Reference Guide.

5. Click **Close** when you are finished adding counters.

The Performance Monitor window displays a graph of the counters you have selected. The graph updates itself at regular intervals until you close the Performance Monitor window.

6. Optionally, specify the color, scale, line width, and line style for one or more Steel-Belted Radius counters.

Right-click the name of a counter and choose **Properties** from the context menu. When the System Monitor Properties window opens, specify the characteristics you want the graph of the counter to use.

Table 66 describes the meaning of each perfmon counter.

Table 66: perfmon Counters

perfmon Counter	Meaning
Acct Failures - Insufficient Resources	The number of accounting requests that were discarded because the RADIUS server was unable to obtain sufficient system resources to process the request.
Acct Failures - Invalid Clients	The number of accounting requests that were discarded because the RADIUS client identified in the request was not defined in the RADIUS server database.
Acct Failures - Invalid Requests	The number of accounting requests that were discarded because the request was malformed or contained invalid attributes.
Acct Failures - Invalid Shared Secret	The number of accounting requests that were discarded because the request contained an invalid digital signature. This is usually due to a mismatch in the shared secrets defined on the

perfmon Counter	Meaning
	RADIUS client and the RADIUS server.
Acct Proxy Failures	The number of forwarded accounting requests for which failures were encountered.
Acct Requests Forwarded	The number of accounting requests that were forwarded to other RADIUS servers.
Acct Requests Retried	The number of unique accounting requests for which retries were received by the RADIUS server.
Acct Requests Retried/sec	The number of accounting requests per second for which one or more retries has been received by the RADIUS server.
Acct Retry Requests	The number of actual accounting request retries received by the RADIUS server.
Acct Retry Requests/sec	The number of accounting request retries per second received by the RADIUS server.
Acct Service Time	The number of seconds that elapsed from the time the last completed accounting request was received to the time the RADIUS server sent a response. Responses generated by proxies are not reflected in this statistic.
Acct Starts	The number of accounting start requests received by the RADIUS server. An accounting start signifies the granting of a connection to an end-user by the remote access server.
Acct Starts/sec	The number of accounting start requests received by the RADIUS server per second. An accounting start signifies the granting of a connection to an end-user by the remote access server.
Acct Stops	The number of accounting stop requests received by the RADIUS server. An accounting stop signifies that an end-user has disconnected from the remote access server.
Acct Stops/sec	The number of accounting stop requests received by the RADIUS server per second. An accounting stop signifies that an end-user has disconnected from the remote access server.
Auth Failure - Authentication Failures	Number of unique authentication requests to which the RADIUS server replied with a reject because no user specified in the database possessed a matching password. A mismatch in shared secrets would also cause this counter to be incremented.
Auth Failure - Checklist Mismatches	Number of unique authentication requests to which the RADIUS server replied with a reject because the request did not include required checklist information.
Auth Failure - Insufficient Resources	Number of unique authentication requests to which the RADIUS server replied with a reject because the RADIUS server ran into a system resource limitation.
Auth Failure - Invalid Clients	Number of unique authentication requests to which the RADIUS server replied with a reject because the request was from a RADIUS client not identified in the RADIUS server's database.
Auth Failure - Invalid Requests	Number of unique authentication requests to which the RADIUS server replied with a reject because the request was malformed or contained invalid attributes.
Auth Proxy Failures	The number of forwarded authentication requests for which failures were encountered.
Auth Proxy Rejects	The number of forwarded authentication requests for which rejects were received from the target RADIUS server.
Auth Requests	The number of unique authentication requests that the RADIUS server has received.
Auth Requests Forwarded	The number of authentication requests that were forwarded to another RADIUS server.
Auth Requests Retried	The number of unique authentication requests for which retries were received by the RADIUS server.
Auth Requests Retried/sec	The number of authentication requests per second for which one or more retries has been received by the RADIUS server.

perfmon Counter	Meaning
Auth Requests/sec	The number of unique authentication requests that the RADIUS server has received per second.
Auth Retry Requests	The number of actual authentication request retries received by the RADIUS server.
Auth Retry Requests/sec	The number of authentication request retries per second received by the RADIUS server.
Auth Service Time	The number of seconds that elapsed from the time the last completed authentication request was received to the time the RADIUS server sent an Accept response. Accept responses generated for tunnel requests or by proxies are not reflected in this statistic.
Auth SQL Disconnects	The number of times an existing connection to a SQL authentication database failed.
Auth SQL Failures	The number of times an attempt to connect to a SQL authentication database failed.
Auth SQL Records Not Found	The number of times no record was found in a SQL authentication database for the specified username.
Auth SQL Timeouts	The number of times a timeout occurred attempting to execute a SQL authentication request.
Auth Successes	The number of unique authentication requests to which the RADIUS server replied with an Accept.
Auth Successes/sec	The number of unique authentication requests to which the RADIUS server replied with an accept per second.
Concurrency Auth Failures	The number of times an authentication request was forwarded to the concurrency server and the concurrency server returned a reject for reasons other than users being over their port limits.
Concurrency Auth Service Time	The number of seconds elapsed from the last time an authentication request was sent to the concurrency server and a response was received.
Concurrency Over Port Limit	The number of times an authentication request was forwarded to the concurrency server and the concurrency server returned a reject because users were over their port limits.
Failed Auths - 1	The number of failed authentication requests that were encountered for clients categorized in collection number 1. To set up the Failed Auths - n counter, you must configure the [FailedAuthOriginStats] section of radius.ini. For information on radius.ini, refer to the Steel-Belted Radius Reference Guide.
Failed Auths - 2	The number of failed authentication requests that were encountered for clients categorized in collection number 2.
Failed Auths - 3	The number of failed authentication requests that were encountered for clients categorized in collection number 3.
Failed Auths - 4	The number of failed authentication requests that were encountered for clients categorized in collection number 4.
Failed Auths - 5	The number of failed authentication requests that were encountered for clients categorized in collection number 5.
Failed Auths - 6	The number of failed authentication requests that were encountered for clients categorized in collection number 6.
Failed Auths - 7	The number of failed authentication requests that were encountered for clients categorized in collection number 7.
Failed Auths - 8	The number of failed authentication requests that were encountered for clients categorized in collection number 8.

perfmon Counter	Meaning
Failed Auths - 9	The number of failed authentication requests that were encountered for clients categorized in collection number 9.
Failed Auths - 10	The number of failed authentication requests that were encountered for clients categorized in collection number 10.
Failed Auths - 11	The number of failed authentication requests that were encountered for clients categorized in collection number 11.
Failed Auths - 12	The number of failed authentication requests that were encountered for clients categorized in collection number 12.
Failed Auths - 13	The number of failed authentication requests that were encountered for clients categorized in collection number 13.
Failed Auths - 14	The number of failed authentication requests that were encountered for clients categorized in collection number 14.
Failed Auths - 15	The number of failed authentication requests that were encountered for clients categorized in collection number 15.
Failed Auths - 16	The number of failed authentication requests that were encountered for clients categorized in collection number 16.
Forwarded Requests Retried	The number of unique forwarded accounting and authentication requests for which retries were transmitted by the RADIUS server.
Forwarded Requests Retried/sec	The number of unique forwarded accounting and authentication requests for which retries were transmitted per second by the RADIUS server.
Forwarded Retry Requests	The number of actual retransmissions of forwarded accounting and authentication request performed by the RADIUS server.
Forwarded Retry Requests/sec	The number of actual retransmissions of forwarded accounting and authentication request per second performed by the RADIUS server.
Proxy Failures - Insufficient Resources	The number of authentication and accounting requests that were forwarded to other RADIUS servers for which the RADIUS server was unable to obtain sufficient system resources to process the request.
Proxy Failures - Invalid Response	The number of authentication and accounting requests that were forwarded to other RADIUS servers for which malformed or invalid responses were received.
Proxy Failures - Invalid Shared Secret	The number of authentication and accounting requests that were forwarded to other RADIUS servers for which responses were discarded because the response contained an invalid digital signature. This is usually due to a mismatch in the shared secrets defined on the RADIUS client and RADIUS server.
Proxy Failures - Time Out	The number of authentication and accounting requests that were forwarded to other RADIUS servers for which no response was received after the specified number of retries.
Seconds since started	Number of seconds Steel-Belted Radius has been running.
Sessions Online	The number of sessions currently active in the RADIUS server's Sessions list.
Static Acct Service Time	The number of seconds elapsed from the last time an accounting request was sent to the static accounting proxy server and a response was received.
Total Acct Failures	The number of unique accounting requests to which the RADIUS server did not reply. Reasons for the failures are identified in other statistics.
Total Acct Failures/sec	The number of unique accounting requests to which the RADIUS server did not reply per

perfmon Counter	Meaning
	second because of an error. Reasons for the failures are identified in other statistics.
Total Acct Offs	The number of accounting off requests received by the RADIUS server. An accounting off signifies that the accounting support in the RADIUS client has been disabled. This request is most often issued when a RADIUS client in being shut down.
Total Acct Offs/sec	The number of accounting off requests received by the RADIUS server per second.
Total Acct Ons	The number of accounting on requests received by the RADIUS server. An accounting on signifies that the accounting support in the RADIUS client has been enabled. This request is most often issued when a RADIUS client is powered on.
Total Acct Ons/sec	The number of accounting on requests received by the RADIUS server per second.
Total Auth Challenges	The number of authentication requests that resulted in a RADIUS challenge response.
Total Auth Failures	The number of unique authentication requests to which the RADIUS server replied with a reject. Reasons for the failures are identified in other statistics.
Total Auth Failures/sec	The number of unique authentication requests to which the RADIUS server replied with a reject, per second. Reasons for the failures are identified in other statistics.
Total Forwarded Request Failures	The number of forwarded authentication and accounting requests that encountered failures.
Total Forwarded Request Failures/sec	The number of forwarded authentication and accounting requests that encountered failures, per second.
Total Forwarded Requests	The number of authentication and accounting requests that were forwarded to other RADIUS servers.
Total Forwarded Requests/sec	The number of authentication and accounting requests per second that were forwarded to other RADIUS servers.
Users Online	The number of unique users represented in the RADIUS server’s Sessions List.

Appendix D
Authentication Protocols

This appendix provides a matrix of authentication methods and their supported authentication protocols.

Table 67: Authentication Protocols

Method	PAP	CHAP	MS-CHAP	MS-CHAP-V2	LEAP	EAPM-SCHAP-V2	EAP-MD5	PAP/ Token card	EAP/ Token card
Microsoft PEAP available inner authentication protocols	No	No	No	No	No	Yes	No	No	No
Cisco PEAP available inner authentication protocols	No	No	No	No	Yes	Yes	Yes	No	Yes
TTLS available inner authentication protocols	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Local	Yes	Yes	Yes	Yes	Yes	Yes	Yes	N/A	N/A
Local (password saved as Allow PAP only, {SHA} or {crypt}).	Yes	No	No	No	No	No	No	N/A	N/A
Windows Domain Authentication									
Windows Domain Group	Yes	No	Yes	Yes	Yes	Yes	No	N/A	N/A
Windows Domain User	Yes	No	Yes	Yes	Yes	Yes	No	N/A	N/A
UNIX authentication methods									
UNIX User	Yes	No	No	No	No	No	No	N/A	N/A
UNIX Group	Yes	No	No	No	No	No	No	N/A	N/A
Other authentication plug-ins									

Method	PAP	CHAP	MS-CHAP	MS-CHAP-V2	LEAP	EAPM-SCHAP-V2	EAP-MD5	PAP/ Token card	EAP/ Token card
RSA SecurID	Yes	No	No	No	No	No	No	N/A	Yes
TACACS+	Yes	Yes	No	No	No	No	Yes	N/A	N/A
LDAP									
BIND (this includes AD and eDirectory/NDS)	Yes	No	No	No	No	No	No	N/A	N/A
BINDNAME (password stored in clear text)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	N/A	N/A
BINDNAME (password stored in SHA Crypt text)	Yes	No	No	No	No	No	No	N/A	N/A
BINDNAME (password stored as MD4 hash of unicode value text)	Yes	No	No	Yes	No	Yes	No	N/A	N/A
BINDNAME (password stored as enc-md5)	Yes	Yes	No	No	No	No	No	N/A	N/A
SQL									
Password stored in clear text	Yes	Yes	Yes	Yes	Yes	Yes	Yes	N/A	N/A
Password password stored in SHA Crypt text	Yes	No	No	No	No	No	No	N/A	N/A
Password stored as {MD4} hash of unicode value text	Yes	No	No	Yes	No	Yes	No	N/A	N/A
Password stored as {enc-md5}	Yes	Yes	No	No	No	No	No	N/A	N/A

Appendix E
Importing and Exporting Data

This appendix describes how to export database information from one Steel-Belted Radius server to an XML file, and then selectively import database information into another Steel-Belted Radius server. The ability to export and import information facilitates configuration of multiple Steel-Belted Radius servers.

Steel-Belted Radius uses XML files with UTF-8 encoding to store exported information. Export files contain only those attributes that have been assigned values, including defaults. For example, if an exported local user item does not have a profile associated with it, the exported information will not identify a profile name, though it will include default values required by Steel-Belted Radius.

ℹ Note: Versions of Steel-Belted Radius earlier than Version 5.0 used a file format called RADIUS Import File (RIF) to export and import information. For backward compatibility, Steel-Belted Radius includes a utility to convert RIF files to XML format. For more information, see the Steel-Belted Radius Installation and Upgrade Guide.

Exporting to a RADIUS Information File via Legacy SBR Administrator

To export information from the Steel-Belted Radius database to an XML file:

1. Run the SBR Administrator.

2. Choose File > Export

3. When the Export dialog (Figure 240: Export Dialog) opens, select the information you want to export.

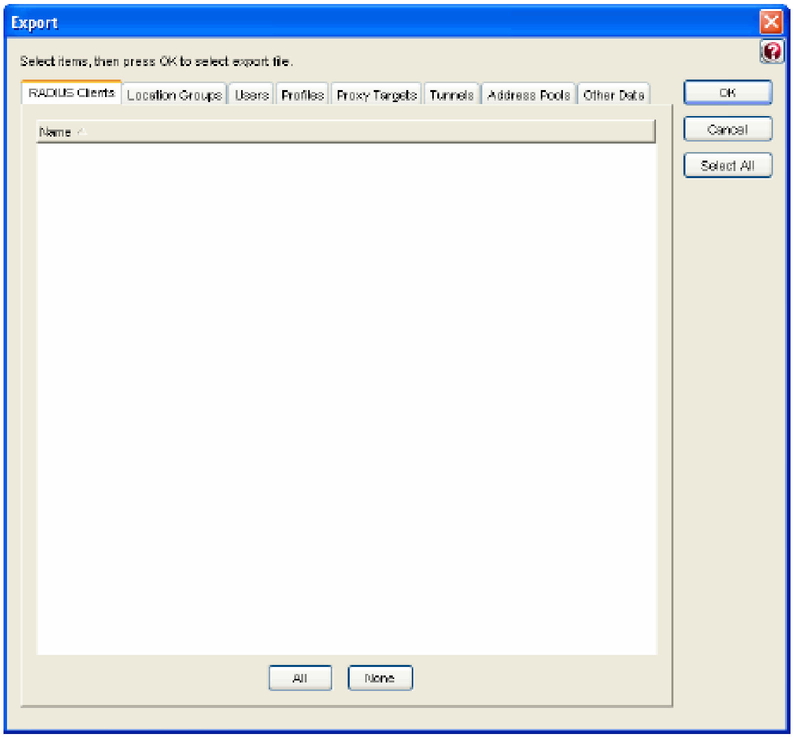
   Each tab in the dialog lists exportable items of particular category. For each category, select the appropriate tab and click each item you'd like to export. To select a contiguous range of items, select the first item in the range, hold down the Shift key, and click the last item in the range.

   To select a non-contiguous set of items, hold down the Ctrl key as you click each item you want.

   To select all items in a category, click **All**.

   To select all items in all categories, click **Select All**.

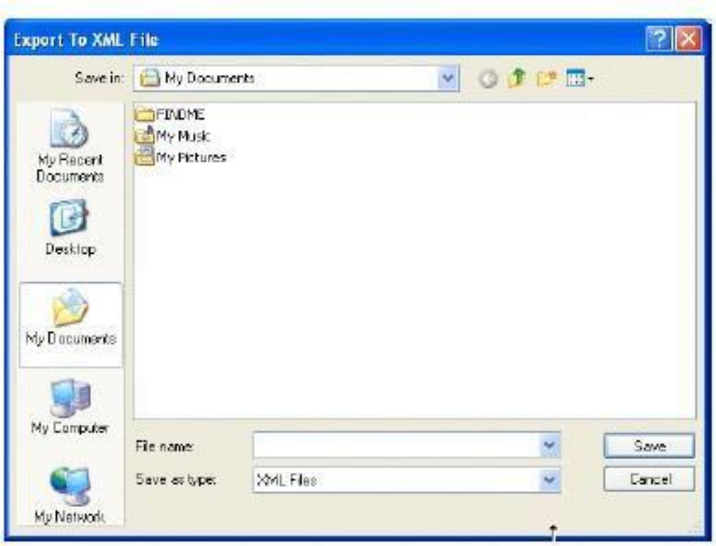
Figure 240: Export Dialog



4. After you have selected the items you want to export, click OK.

5. When the Export to XML dialog opens, specify a file name and click Save.

Figure 241: Export to XML Dialog



Exporting to a RADIUS Information File via WebGUI

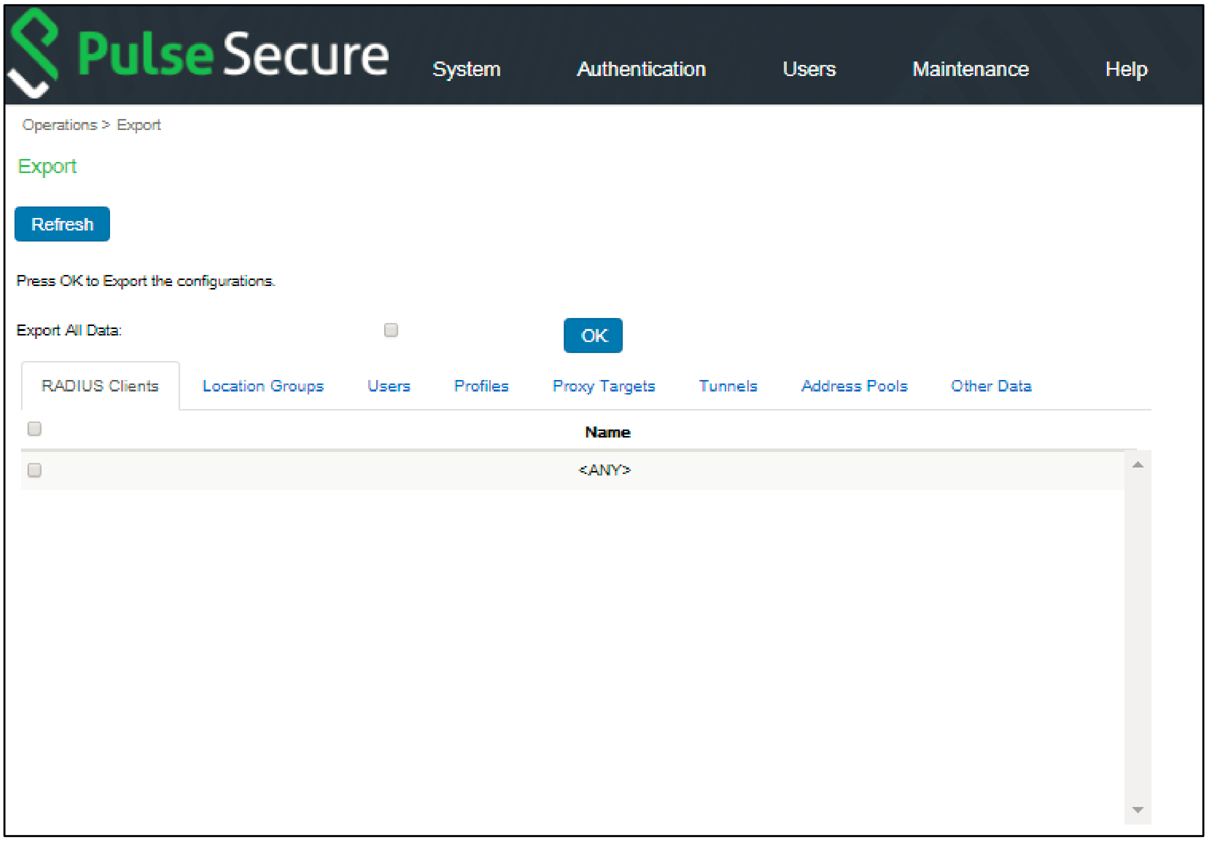
To export information from the Steel-Belted Radius database to an XML file:

1. From the WebGUI, choose Maintenance > Operations > Export.

2. When the Export page (Figure 242: Export Page) opens, select the information you want to export.

Each tab in the page lists exportable items of a particular category. For each category, select the appropriate tab and click each item you'd like to export.
To select all items in a category, select the check box in the table header in each category.
To select all items in all categories, select the Export All Data check box.

Figure 242: Export Page



Press OK to Export the configurations.

Export All Data: □

□	Name
□	<ANY>

3. After you have selected the items you want to export, click OK.

4. SBR-E_Export.xml file will be automatically downloaded.

Importing into the Steel-Belted Radius Database via Legacy SBR Administrator

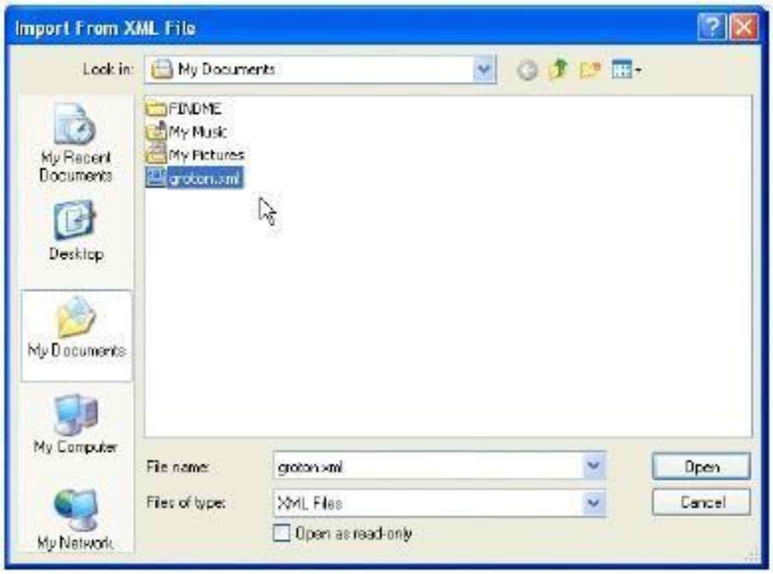
To import information from an XML file into the database on your Steel-Belted Radius server:

1. Run SBR Administrator.

2. Choose File > Import.

3. When the Import from XML dialog opens, select the XML file containing the information you want to import and click Open.

Figure 243: Import from XML File Dialog



Import From XML File

Look in: My Documents

File name: groton.xml

Files of type: XML Files

☐ Open as read-only

4. When the Import dialog Figure 244: Import Dialog opens, specify whether what the SBR Administrator should do when it finds an object with the same name in the Steel-Belted Radius database.

   - Click Skip if you want SBR Administrator to leave the item already in the database intact.
   - Click Replace if you want SBR Administrator to overwrite the item in the database with the imported information.

5. Select the information you want to import by clicking each tab and selecting items to import.

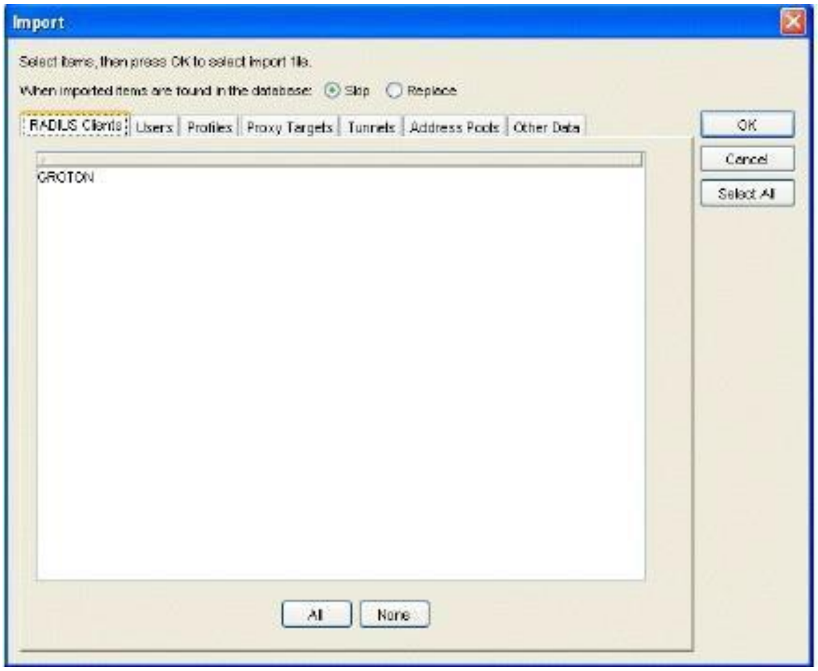
   To select a contiguous range of items, select the first item in the range, hold down the SHIFT key, and click the last item in the range.

   To select a non-contiguous set of items, hold down the Ctrl key as you click each item you want.

   To select all items in a category, click All.

   To select all items in all categories, click Select All.

Figure 244: Import Dialog



Import

Select items, then press OK to select import file.

When imported items are found in the database: ● Skip ○ Replace

GROTON

6. After you select all the items you want, click **OK**. The items you selected are added to the Steel-Belted Radius database.

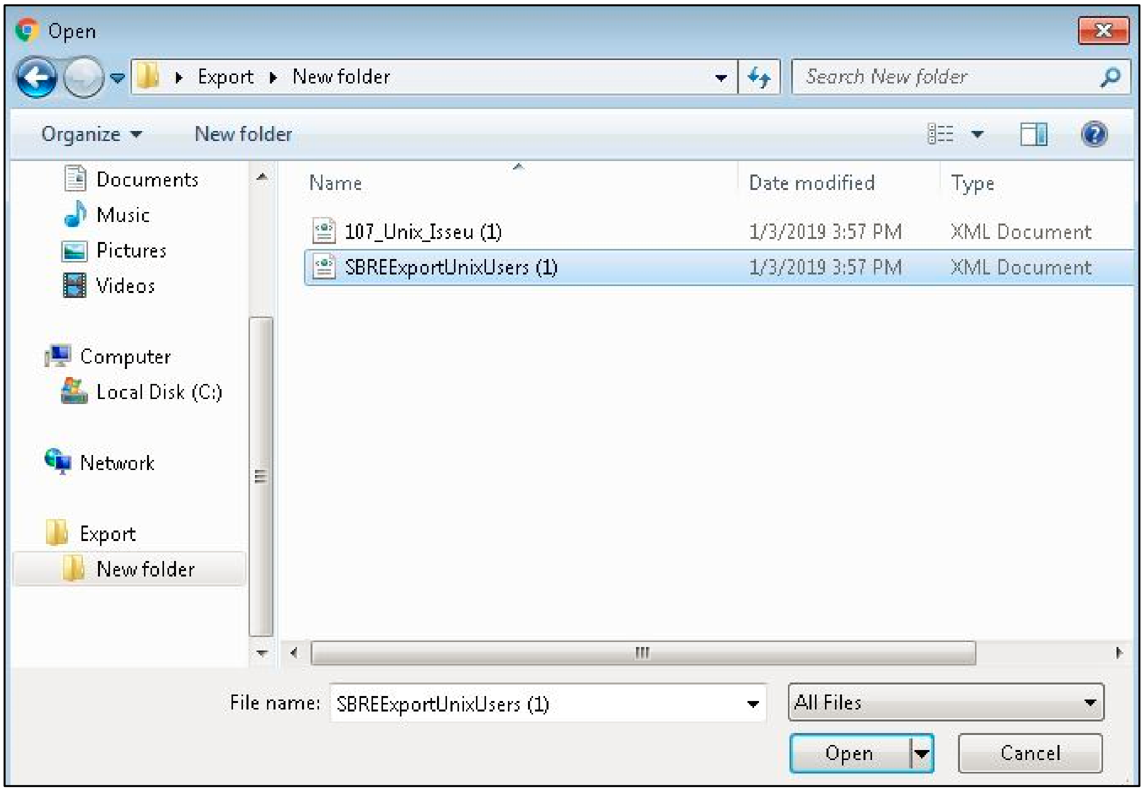
Importing into the Steel-Belted Radius Database via WebGUI

To import information from an XML file into the database on your Steel-Belted Radius server:

1. From the WebGUI, choose Maintenance > Operations > Import.

2. Click the Choose File/Browser button, select the xml file containing the information you want to import and click Import.

Figure 245: Import from XML File Page



Name	Date modified	Type
107_Unix_Isseu (1)	1/3/2019 3:57 PM	XML Document
SBREExportUnixUsers (1)	1/3/2019 3:57 PM	XML Document

File name: SBREExportUnixUsers (1)

All Files

3. When the Import page (Figure 246: Import Page) opens, specify whether what the SBR Administrator should do when it finds an object with the same name in the Steel-Belted Radius database.

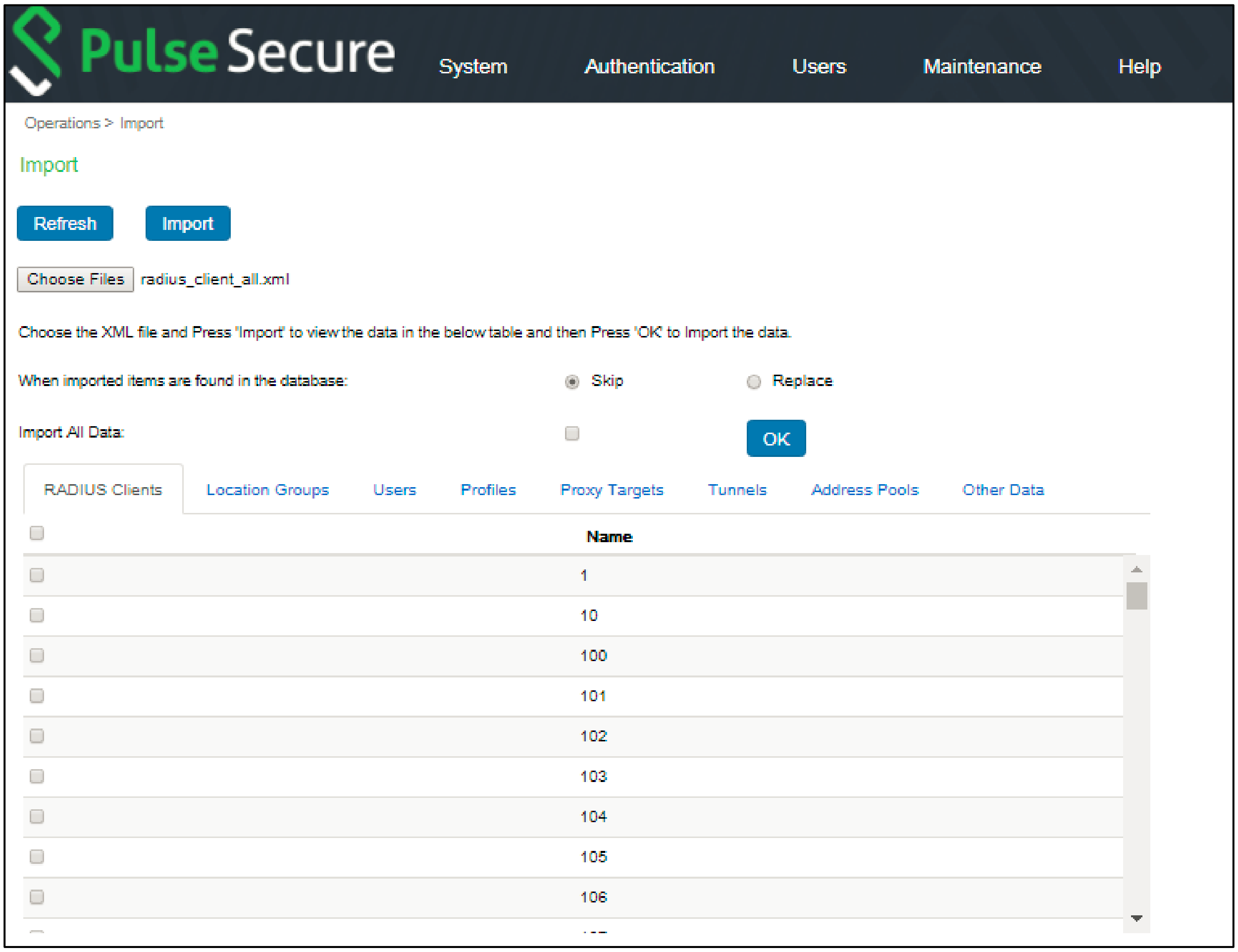
   - Click Skip if you want SBR Administrator to leave the item already in the database intact.
   - Click Replace if you want SBR Administrator to overwrite the item in the database with the imported information.

4. Select the information you want to import by clicking each tab and selecting items to import.

   To select all items in a category, select the check box in the table header in each category.

   To select all items in all categories, select the Import All Data check box.

Figure 246: Import Page



radius_client_all.xml

Choose the XML file and Press 'Import' to view the data in the below table and then Press 'OK' to Import the data.

When imported items are found in the database: ● Skip ○ Replace

Import All Data: □

□	Name
□	1
□	10
□	100
□	101
□	102
□	103
□	104
□	105
□	106

5. After you select all the items you want, click **OK**. The items you selected are added to the Steel-Belted Radius database.

Appendix F
Stopping and Starting Steel-Belted Radius

This appendix describes how to stop and restart the Steel-Belted Radius server.

Stopping the Steel-Belted Radius Server

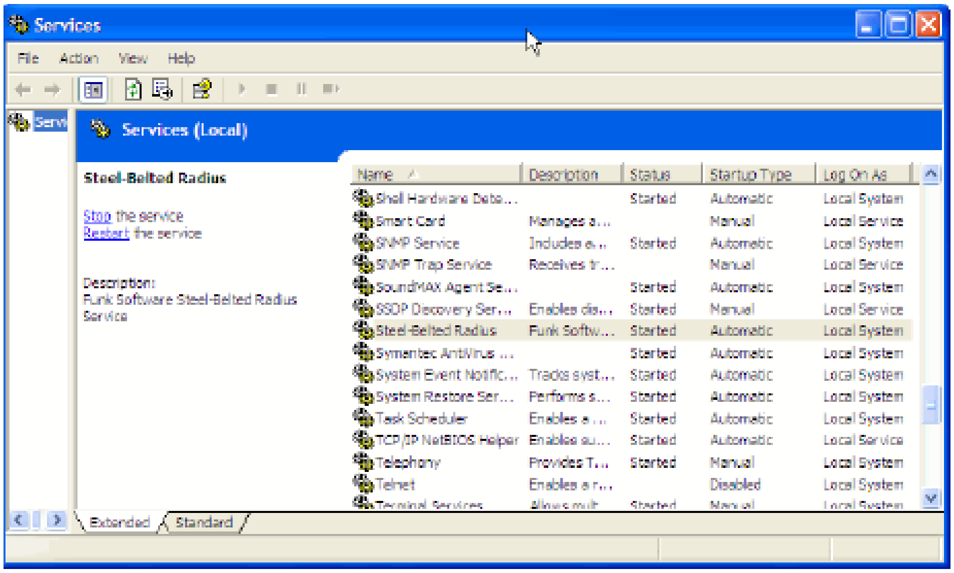
After the Steel-Belted Radius service (Windows) or RADIUS daemon (Linux) is installed, it stops and starts automatically each time you shut down or restart the server. If you modify the settings in the Steel-Belted Radius configuration files, you may need to restart the Steel-Belted Radius server manually before the server recognizes its new settings.

Windows

You can stop the Steel-Belted Radius service at any time by performing the following steps:

1. Choose **Start > Control Panel > Administrative Tools > Services**.
2. When the Services window opens, click the Steel-Belted Radius entry.

Figure 247: Services Window



Steel-Belted Radius

Stop the service
Restart the service

Description:
Funk Software Steel-Belted Radius Service

Name	Description	Status	Startup Type	Log On As
Shell Hardware Dete...		Started	Automatic	Local System
Smart Card	Manages a...		Manual	Local Service
SNMP Service	Includes a...	Started	Automatic	Local System
SNMP Trap Service	Receives tr...		Manual	Local Service
SoundMAX Agent Se...		Started	Automatic	Local System
SSDP Discovery Ser...	Enables dis...	Started	Manual	Local Service
Steel-Belted Radius	Funk Softw...	Started	Automatic	Local System
Symantec AntiVirus ...		Started	Automatic	Local System
System Event Notific...	Tracks syst...	Started	Automatic	Local System
System Restore Ser...	Performs s...	Started	Automatic	Local System
Task Scheduler	Enables a ...	Started	Automatic	Local System
TCP/IP NetBIOS Helper	Enables su...	Started	Automatic	Local Service
Telephony	Provides T...	Started	Manual	Local System
Telnet	Enables a r...		Disabled	Local System
Terminal Services	Allows mult	Started	Manual	Local System

3. Click the **Stop the service** button.

Linux

After the RADIUS daemon is installed on the server, it stops and starts automatically each time you shut down or restart the server. You can stop the RADIUS daemon on a Linux server at any time by issuing the following commands:

```
cdserver-directory
./sbrd stop
```

When you execute the sbrd stop command, Steel-Belted Radius allows its subsystems to complete outstanding work, release resources, and then stops the the radius service gracefully.

If Steel-Belted Radius fails to stop after you issue an sbrd stop command, you can use the optional force argument to terminate all subsystems immediately.

```
cd server-directory
./sbrd stop force
```

Starting the Steel-Belted Radius Server

You must restart the Steel-Belted Radius service (Windows) or RADIUS daemon (Linux) after you modify the Steel-Belted Radius configuration files.

Windows

To start the Steel-Belted Radius service on a Windows server after it has been stopped:

1. Choose **Start > Control Panel > Administrative Tools > Services**.
2. When the Services window Figure 247: Services Window opens, click the Steel-Belted Radius entry.
3. Click the **Start the service** button.

To restart the Steel-Belted Radius service without stopping it:

1. Choose **Start > Control Panel > Administrative Tools > Services**.
2. When the Services window Figure 247: Services Window opens, click the Steel-Belted Radius entry.
3. Click the **Restart the service** button.

Linux

Use the following command to start the RADIUS daemon after you have issued an sbrd stop command on a Linux server:

```
cd server-directory
./sbrd start
```

When you execute the sbrd start command, Steel-Belted Radius starts the radius service.

If you change configuration settings for your Steel-Belted Radius server, you may need to restart the server to make the changes effective. As an alternative to issuing an sbrd stop command immediately followed by an sbrd start command, you can use the sbrd restart command when you want to restart Steel-Belted Radius. When you issue the sbrd restart command, the system shuts down the radius service (if it is running), and then immediately starts the radius service.

```
cd server-directory
./sbrd restart
```

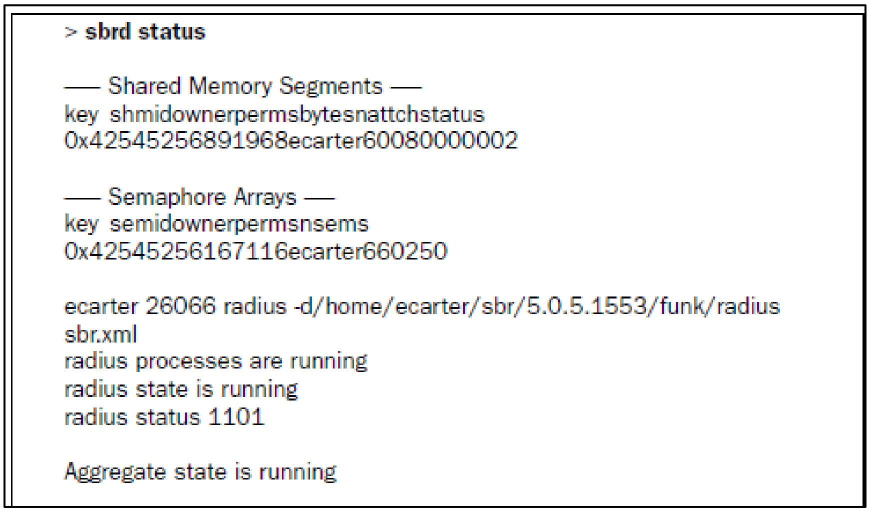
Displaying RADIUS Status Information (Linux)

You can use the sbrd status command to display status information for the RADIUS daemon.

```
cd server-directory
./sbrd status
```

Figure 248: Output of sbrd status Command illustrates the output of the sbrd status command.

Figure 248: Output of sbrd status Commandd



```
> sbrd status

—— Shared Memory Segments ——
key  shmidownerpermsbytesnattchstatus
0x42545256891968ecarter60080000002

—— Semaphore Arrays ——
key  semidownerpermsnsems
0x42545256167116ecarter660250

ecarter 26066 radius -d/home/ecarter/sbr/5.0.5.1553/funk/radius
sbr.xml
radius processes are running
radius state is running
radius status 1101

Aggregate state is running
```

Appendix G
Stopping and Starting "Steel-Belted Radius Jetty Server"

Windows

Once you start Steel-Belted Radius Service, another service for Java Web server (Jetty) will be created automatically. You can use the entry "Steel-Belted Radius Jetty Server" service to start/restart and stop the Java Web server.

Note: When you stop the "Steel-Belted Radius" Service, the "Steel-Belted Radius Jetty Server" Service will be stopped and uninstalled automatically. Similarly, when you restart the "Steel-Belted Radius" Service, the "Steel-Belted Radius Jetty Server" Service will be restarted automatically.

Stopping Jetty Server

You can stop the "Steel-Belted Radius Jetty Server" service at any time by performing the following steps:

1. Choose **Start** > **Control Panel** > **Administrative Tools** > **Services**.
2. When the Services window opens, click the Steel-Belted Radius Jetty Server entry.
3. Click **Stop the service** button.

Starting Jetty Server

To start the "Steel-Belted Radius Jetty Server service" on a Windows server after it has been stopped, perform the following steps:

1. Choose **Start** > **Control Panel** > **Administrative Tools** > **Services**.
2. When the Services window Figure 247: Services Window opens, click the Steel-Belted Radius Jetty Server entry.
3. Click **Start the service** button.

Restarting Jetty Server

To restart the "Steel-Belted Radius service" without stopping it, perform the following steps:

1. Choose **Start** > **Control Panel** > **Administrative Tools** > **Services**.
2. When the Services window Figure 247: Services Window opens, click the Steel-Belted Radius Jetty Server entry.
3. Click **Restart the service** button.

Linux

The Steel-Belted radius Java Web server (Jetty) will be started/stopped/restarted automatically along with the radius server. However, you can start/stop/restart the SBR Jetty server separately by using the following steps:

Stopping Jetty Server

Run the following commands to stop the jetty server

cd server-directory
./sbrd jettystop

Starting Jetty Server

Run the following commands to start the jetty server

cd server-directory
./sbrd jettystart

Restarting Jetty Server

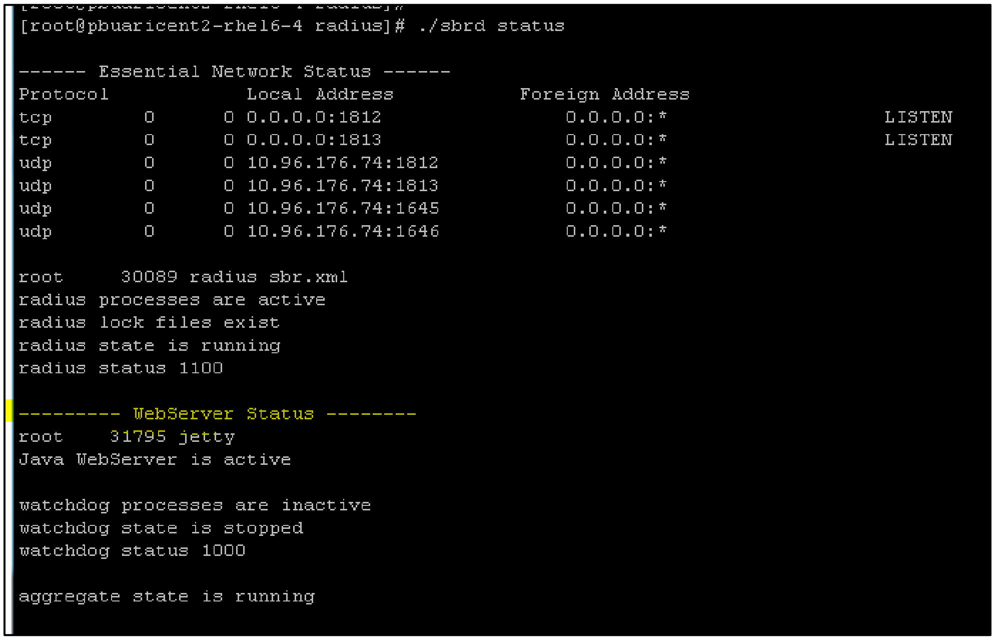
Run the following commands to restart the jetty server

cd server-directory
./sbrd jettyrestart

Viewing the Status of Jetty Server

To view the status of Java Web server, run the following command

Figure 249: Viewing status of Jetty Server



```
[root@pbuaricent2-rhel6-4 radius]# ./sbrd status

------ Essential Network Status ------
Protocol            Local Address           Foreign Address
tcp        0      0 0.0.0.0:1812                0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:1813                0.0.0.0:*                   LISTEN
udp        0      0 10.96.176.74:1812           0.0.0.0:*
udp        0      0 10.96.176.74:1813           0.0.0.0:*
udp        0      0 10.96.176.74:1645           0.0.0.0:*
udp        0      0 10.96.176.74:1646           0.0.0.0:*

root     30089 radius sbr.xml
radius processes are active
radius lock files exist
radius state is running
radius status 1100

--------- WebServer Status --------
root    31795 jetty
Java WebServer is active

watchdog processes are inactive
watchdog state is stopped
watchdog status 1000

aggregate state is running
```

Appendix H
Use Custom SSL Certificate for Launching SBR-E Web UI

By default, Java Web server (Jetty) uses a self-signed certificate in its keystore (for https). To use a custom certificate, edit configurations in the **application.properties** file.

Perform the following steps to edit:

1. Go to **<SBR_INSTALLED_DIR>\Service\Website\jetty** (in Windows) or **<SBR_INSTALLED_DIR/radius/website/jetty** (in Linux).

2. Take a backup of "**application.properties**" file as "application.properties-orig".

3. Open the "**application.properties**" file to edit.

4. Default configurations will be as follows:

   ```
   # The format used for the keystore
   server.ssl.key-store-type=PKCS12 (Default keystore)
   # The path to the keystore containing the certificate
   server.ssl.key-store=classpath:SBRWebServer.p12  (self-signed certificate inside Web server)
   # The password of the certificate
   server.ssl.key-store-password=pulsesbr (password of the self-signed certificate).
   ```

5. If you have a .pfx or .p12 certificate then configure the key store type as follows:

   server.ssl.key-store-type=PKCS12
   If you have a .jks certificate, then configure the type as follows,
   server.ssl.key-store-type=JKS

6. Set the path to the keystore containing the certificate (server.ssl.key-store)
   Configure the path of the certificate keystore as follows:

   server.ssl.key-store=<absolute path of certificate>
   Example:
   server.ssl.key-store=/opt/certs/customSSLCert.p12

7. Set the password of the certificate (server.ssl.key-store-password)
   server.ssl.key-store-password=<password to open the certificate>

8. After setting all the properties, save the file and restart Java Web server (Jetty).

   Linux: ./sbrd jettyrestart
   Windows: Restart "Steel-Belted Radius Jetty Server" service from Services.msc

9. When you launch SBR Web UI, you will be able to see the custom certificate in the web browser.

Index