



Pulse Secure Desktop Client

Administration Guide

Product Release 9.1R4

Document Revision 1.3.1
Published: February 2020

2700 Zanker Road, Suite 200
San Jose, CA 95134
<https://www.pulsesecure.net>

© 2020 by Pulse Secure, LLC. All rights reserved

Pulse Secure and the Pulse Secure logo are trademarks of Pulse Secure, LLC in the United States. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Pulse Secure, LLC assumes no responsibility for any inaccuracies in this document. Pulse Secure, LLC reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Pulse Secure Desktop Client Administration Guide

The information in this document is current as of the date on the title page.

END USER LICENSE AGREEMENT

The Pulse Secure product that is the subject of this technical documentation consists of (or is intended for use with) Pulse Secure software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <https://www.pulsesecure.net/support/eula>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Revision History

The following table lists the revision history for this document.

Feature	Add/Update/Remove	Document Published Date/ Document Version	Effective Release	Notes
		February 2020/1.3.1	9.1R4	Cosmetic Changes
<ul style="list-style-type: none">PSAM Details Tab	Updated " Pulse Secure Client for Windows " section.	January 2020/1.3	9.1R4	
<ul style="list-style-type: none">Yubikey Support	Updated " Configuring Client Certificate Selection Option " section.			
		October 2019/1.2.1	9.1R3	Cosmetic Changes
<ul style="list-style-type: none">Configuring Client Certificate Selection Option	Added " Configuring Client Certificate Selection Option " section.	October 2019/1.2	9.1R3	
<ul style="list-style-type: none">Manage Pulse Secure Client Versions	Updated " Manage Pulse Secure Client Versions " section.	July 2019/1.1	9.1R2	
<ul style="list-style-type: none">Launching Pulse Desktop Client using URLPulse SAM IPv6 SupportPulse Secure Client and Software Defined Perimeter	Added " Launching Pulse Desktop Client using URL " section. Added " Pulse SAM IPv6 Support " section. Added " Pulse Secure Client and Software Defined Perimeter (SDP) " section.	May 2019/1.0	9.1R1	

Table of Contents

Revision History	1
List of Tables	8
About This Guide	9
Objectives	9
Audience	9
Document Conventions	9
Related Documentation	9
Requesting Technical Support	10
<i>Self-Help Online Tools and Resources</i>	10
<i>Opening a Case with PSGSC</i>	10
PART 1 Pulse Secure Client	11
CHAPTER 1 Pulse Secure Client Overview	12
Introducing Pulse Secure Client	12
<i>Pulse Secure Client for Windows</i>	12
<i>Pulse Secure Client for macOS</i>	18
<i>User Experience</i>	19
<i>L3 and Pulse SAM Coexistence</i>	25
<i>HVCI Compatibility</i>	26
<i>Pulse SAM IPv6 Support</i>	26
<i>Location Awareness</i>	27
<i>Centralized Pulse Configuration Management</i>	27
<i>Session Migration</i>	28
<i>Smart Connections - List of URLs</i>	28
<i>Security Certificates</i>	29
<i>Compliance and Remediation</i>	29
<i>Two Factor Authentication</i>	30
<i>Captive Portal Detection</i>	30
<i>Pulse Collaboration Suite Integration</i>	30
<i>Sign In Notifications</i>	31
<i>Automatic Software Updates</i>	31
<i>Pulse Secure Client Customization and Rebranding</i>	31
Pulse Secure Client and Software Defined Perimeter (SDP)	32
Pulse Secure Client Configuration Overview	33
Pulse Secure Client Status Icons	34
Installation Requirements	35
Pulse Secure Client Error Messages Overview	35
Accessing Pulse Secure Client Error Messages on macOS Endpoints	35
Pulse Secure Client Log Files	36
Deleting the Pulse Secure Client Log Files	38
Uploading the Pulse Secure Client Log Files	38
Migrating from Odyssey Access Client to Pulse Secure Client	39
Migrating from Network Connect to Pulse Secure Client	41
Predictable Pulse Server Hostname Resolution with IPv6	42
CHAPTER 2 Configuring Pulse Policy Secure	43
Before You Begin	43
Pulse Policy Secure Overview	44
Pulse Policy Secure and Pulse Connect Secure Deployment Options	44
SRX Series Gateway Deployment Options	45

Configuring a Role for Pulse Policy Secure	45
Client Connection Set Options for Pulse Policy Secure	48
<i>Pulse Secure Connection Set Options</i>	48
<i>UAC 802.1X Connection Type Options</i>	49
<i>Trusted Server List (for UAC 802.1X Connection)</i>	50
<i>Connect Secure or Policy Secure (L3) Connection Type Options</i>	50
<i>SRX (for Dynamic VPN) Connection Type Options</i>	52
<i>Pulse Connection is Established Options</i>	52
<i>Pulse Connection is Established Examples</i>	53
<i>Location Awareness Rules</i>	54
<i>Machine Connection Preferences</i>	55
<i>User Connection Preferences</i>	55
Creating a Client Connection Set for Pulse Policy Secure.....	56
Pulse Secure FIPS Mode Overview for Pulse Policy Secure	59
<i>Windows Endpoint Requirements</i>	59
<i>Configuration Overview</i>	59
Securing the Connection State on the Pulse Secure Client	61
Machine Authentication for Pulse Policy Secure Overview	61
Configuring Machine-Only Machine Authentication for a Pulse Secure Connection	62
Configuring User-After-Desktop Machine Authentication for a Pulse Secure Connection.....	63
Preferred Realm and Role for Pulse Secure Client Machine Authentication.....	64
Remote Desktop Protocol Compatibility with Pulse Secure 802.1X Machine Authentication Connection	66
Credential Provider Authentication for Pulse Policy Secure Overview	66
Configuring User-at-Credprov Credential Provider Authentication for a Pulse Secure Connection	68
Configuring Machine-Then-User-at-Credprov Credential Provider Authentication for a Pulse Secure Connection	69
Configuring a Pulse Credential Provider Connection for Password or Smart Card Login	71
Machine and User Authentication Through a Pulse Connection for Pulse Policy Secure	73
Configuring Location Awareness Rules for Pulse Secure Client.....	74
Pulse Policy Secure Component Set Options.....	76
Creating a Client Component Set for Pulse Policy Secure	76
Endpoint Security Monitoring and Management for Pulse Policy Secure.....	77
<i>Remediation Options</i>	78
Issuing a Remediation Message with Pulse Policy Secure	79
Using SMS/SCCM Remediation with Pulse Policy Secure	80
Patch Management Info Monitoring and Patch Deployment	81
<i>Configuration and Migration Options for Deprecated Custom: Patch Assessment Rules</i>	81
<i>Using a System Management Server</i>	82
Pushing Pulse Secure Client Configurations Between Pulse Servers of the Same Type	82
Enabling or Disabling Automatic Upgrades of the Pulse Secure Client	84
Upgrading Pulse Secure Client.....	84
Using Device Certificates	85
<i>Understanding Device Certificates</i>	86
<i>Understanding Self-Signed Certificates</i>	86
<i>Importing a Device Certificate and Private Key</i>	87
<i>Creating a Certificate Signing Request</i>	87
<i>Importing a Signed Certificate Created from a CSR</i>	88
<i>Understanding Intermediate Certificates</i>	88
<i>Importing Intermediate CA Certificates</i>	89
<i>Importing a Renewed Certificate That Uses the Existing Private Key</i>	89
<i>Downloading a Device Certificate</i>	89
<i>Using Device Certificates with Virtual Ports</i>	89
CHAPTER 3 Configuring Pulse Connect Secure	91
Before You Begin	91
Pulse Connect Secure Overview	91
<i>Pulse Secure Client and IVS</i>	92
<i>Pulse Secure Client and Traffic Enforcement</i>	92

<i>Advanced Client Configuration Feature</i>	93
About Sign-In Notifications	95
Configuring and Implementing Sign-in Notifications	95
Pulse Connect Secure Split Tunneling Overview	97
<i>Split Tunneling Disabled</i>	97
<i>Split Tunneling Enabled</i>	98
<i>Pulse Split Tunneling Summary</i>	98
<i>Split Tunneling Notes</i>	103
Configuring a Role for Pulse Connect Secure	103
<i>Configuring General Role Options for Pulse Connect Secure</i>	104
<i>Configuring Role Options for Host Checker for Pulse Connect Secure</i>	105
Machine Authentication for Pulse Connect Secure Overview	106
Credential Provider Authentication for Pulse Connect Secure Overview	107
<i>Configuring Role Options for Pulse Connect Secure</i>	107
Configuring User-at-Credprov Credential Provider Authentication for a Pulse Secure Connection	110
Configuring Machine-Then-User-at-Credprov Credential Provider Authentication for a Pulse Secure Connection	111
Machine and User Authentication through a Pulse Connection for Pulse Connect Secure	112
Stealth Mode	113
Configuring Pulse Secure Client for Secure Application Manager	120
Pulse Connection Set Options for Pulse Connect Secure	125
<i>Pulse Secure Connection Set Options</i>	125
<i>Configuring Client Certificate Selection Option</i>	127
<i>Always-on VPN</i>	129
<i>Configuring Always-on Options</i>	129
<i>Configuring Always-on VPN Options using Wizards</i>	130
<i>Requirement to set up the appropriate GPOs</i>	140
<i>Always-on with Lock-down Mode</i>	141
<i>Lock-down Exception</i>	142
<i>Program-based Resource Access</i>	144
<i>Port-based Resource Access</i>	145
<i>Custom-based Resource Access</i>	146
<i>Retry Button</i>	146
<i>Captive Portal Remediation with Pulse Client Embedded Mini-Browser</i>	147
<i>Policy Secure 802.1X Connection Type Options</i>	148
<i>Trusted Server List (for Policy Secure 802.1X Connection)</i>	148
<i>Connect Secure or Policy Secure (L3) Connection Type Options</i>	149
<i>SRX (for Dynamic VPN) Connection Type Options</i>	150
<i>Pulse Connection is Established Options</i>	151
<i>Pulse Connection is Established Examples</i>	151
<i>Location Awareness Rules</i>	153
<i>Machine Connection Preferences</i>	154
<i>User Connection Preferences</i>	154
Securing the Connection State on the Pulse Secure Client	155
Creating a Client Connection Set for Pulse Connect Secure	155
Pulse Secure Client FIPS Mode for Pulse Connect Secure Overview	156
<i>Endpoint Requirements</i>	157
<i>Configuration Overview</i>	157
Configuring Location Awareness Rules for Pulse Secure Client	159
Component Set Options for Pulse Connect Secure	160
<i>Creating a Client Component Set for Pulse Connect Secure</i>	161
<i>Manage Pulse Secure Client Versions</i>	161
Endpoint Security Monitoring and Management for Pulse Connect Secure	163
<i>Remediation Options</i>	164
Issuing a Remediation Message with Pulse Connect Secure	165
Using SMS/SCCM Remediation with Pulse Connect Secure	166
Pushing Pulse Configurations Between Pulse Servers of the Same Type	167
Enabling or Disabling Automatic Upgrades of the Pulse Secure Client	168

Upgrading Pulse Secure Client.....	168
Pulse Collaboration Suite Overview	169
<i>Task Summary: Configuring Pulse Collaboration Suite on Pulse Connect Secure</i>	170
<i>Configuring Pulse Connections to Support Meetings</i>	170
<i>Scheduling Meetings Through the Pulse Connect Secure User Web Portal</i>	171
<i>Scheduling Meetings Through Microsoft Outlook</i>	171
CHAPTER 4 Configuring Pulse Secure Client on SRX Series Gateways	173
Pulse Secure Client and SRX Series Gateways	173
Pulse Secure Client and Dynamic VPN Configuration Overview	174
CHAPTER 5 Session Migration	175
Understanding Session Migration.....	176
<i>Session Migration Overview</i>	176
<i>Session Migration and Session Timeout</i>	177
<i>How Session Migration Works</i>	177
<i>Session Migration and Session Lifetime</i>	178
<i>Session Migration and Load Balancers</i>	178
<i>Authentication Server Support</i>	178
Task Summary: Configuring Session Migration	178
Configuring Session Migration for the Pulse Client	179
Configuring an IF-MAP Federated Network for Session Migration.....	179
CHAPTER 6 Deploying Pulse Secure Client.....	181
Pulse Secure Client Installation Overview	181
Adding a Pulse Configuration to a New Pulse Installation	183
Installing Pulse Secure Client from the Web.....	187
Launching Pulse Secure Client from the Pulse Server Web Portal.....	188
<i>Usage Notes</i>	188
Launching Pulse Desktop Client using URL	188
Installing the Pulse Secure Client on Windows Endpoints Using a Preconfiguration File	196
<i>Installing the Pulse Client Using Advanced Command-Line Options</i>	197
<i>Examples</i>	197
<i>Repairing a Pulse Installation on a Windows Endpoint</i>	197
Installing the Pulse Secure Client on OS X Endpoints Using a Preconfiguration File	198
<i>Installing the Pulse Client on OS X Endpoints Using Command-Line Options</i>	198
Pulse Secure Command-line Launcher	199
Using jamCommand to Import Pulse Secure Connections	201
jamCommand Reference.....	202
CHAPTER 7 Customizing the Pulse Secure Desktop Client	204
Customizing Pulse Secure Client Overview.....	204
<i>BrandPackager Usage Notes</i>	205
BrandPackager Workflow	206
Setting Up the Pulse Secure Client Customization Environment	206
Initializing the Pulse Secure Client Customization Environment	207
Importing an Existing Customized Pulse Secure Client Package	208
Editing Pulse Secure Client User Interface Labels	209
Editing Pulse Secure Client Messages.....	212
Adding Custom Graphics to Pulse Secure Client.....	213
Customizing Pulse Secure Client for Apple OS X Online Help	214
Validating Customizations to the Pulse Secure Client	215
Building the New Pulse Secure Client Package.....	215
Testing the Pulse Secure Client Package.....	215
Installing or Upgrading Pulse for Windows with a Branding Package	216
Installing or Upgrading Pulse for Apple OS X with a Branding Package	216

Installing a Branding Package Only	217
PART 2 Pulse Secure Client Compatibility	219
CHAPTER 8 Client Software Feature Comparison	220
Comparing Odyssey Access Client and Pulse Secure Client	220
Comparing Network Connect and Pulse Secure Client	223
Pulse Split Tunneling	225
PART 3 Pulse Secure Client for Mobile Devices	226
CHAPTER 9 Pulse Secure Client for Mobile Devices	227
Pulse Secure Client for Mobile Devices Overview	227
Pulse Secure Mobile Clients and User Agent Strings	227
CHAPTER 10 Pulse Secure Client for Apple iOS	229
Pulse Secure Client for Apple iOS Overview	229
Before You Begin	230
Configuring a Role and Realm for Pulse Secure Client for Apple iOS	231
Allowing Pulse Secure Client for iOS Users to Save Webmail Password	233
Host Checker for Pulse iOS Clients	233
Configuring Host Checker for Pulse Secure iOS Clients	234
Implementing Host Checker Policies for Pulse Secure Client for iOS Devices	235
Installing the Pulse Secure Client for Apple iOS App	236
Using iPhone Configuration Utility Profiles for Pulse Secure Client for iOS	236
Collecting Log Files from Pulse Secure Client for iOS	237
Launching the Pulse Secure Client for iOS App with a Command	237
Pulse Secure Client for iOS Error Message Reference	238
Configuring Secure Mail	239
Enabling Secure Mail at the Role Level	239
Defining the Secure Mail Resource Profile	240
Obtaining an S/MIME Certificate	241
Configuring an Authorization-Only Policy for ActiveSync	242
CHAPTER 11 Pulse Secure Client for Google Android	244
Pulse Secure Client for Android Overview	244
Configuring a Role and Realm for Pulse for Android	245
Application Acceleration on Pulse Secure Client Mobile VPN Connections	246
Configuring Application Acceleration for Pulse Mobile VPN Connections	247
Allowing Pulse Secure Client for Android Users to Save Webmail Password	247
Host Checker for Pulse Android Clients	247
Configuring Host Checker for Pulse Secure Client Android Clients	248
Implementing Host Checker Policies for Pulse Secure Client for Android Clients	249
Pulse Secure Client for Android Error Message Reference	250
Box-Introduction to the Android Client API	251
API Access Requirement	251
Package Name	251
AIDL File	251
Service intent filter action name	252
JSON String Format for createConnection	252
Single Username/Password Authentication	252
Certificate Authentication	252
Username/Password and Certificate Authentication	253
Dual Username/Password Authentication	253
XML Command Format for doCommand (Deprecated)	254
Return Codes for doCommand	254

API Functions.....	255
<code>getVersion()</code>	255
<code>createConnection(String jsonProfile)</code>	256
<code>removeConnection(String profileName)</code>	256
<code>getAllConnections()</code>	256
<code>getConnection(String profileName)</code>	257
<code>startConnection(String profileName)</code>	257
<code>stopConnection(String profileName)</code>	257
<code>getState(String profileName)</code>	258
<code>getErrorString(String profileName)</code>	258
<code>doCommand(String commandXML)</code>	258
<code>addVPNConnection</code>	259
<code>checkVPNConnection</code>	260
<code>updateVPNConnection</code>	261
<code>deleteVPNConnection</code>	261
<code>addToKeyStore</code>	262
Launching the Pulse Secure Client for Android App through a Browser Link.....	262
CHAPTER 12 Pulse Secure Client for Windows Phone.....	264
Pulse Secure Client for Windows Phone Overview.....	264
<i>Pulse Secure Client for Windows Phone Supported Platforms</i>	264
<i>Pulse Secure Client for Windows Phone Supported Features</i>	265
<i>Pulse Secure Client for Windows Phone Limitations</i>	265
Configuring Pulse Connect Secure for Pulse Secure Client for Windows Phone VPN Connections.....	266
Configuring a Pulse Secure Connection for Windows Phone – Manual Configuration.....	267
Host Checker for Pulse Secure Client for Windows Phone.....	269
Configuring Host Checker for Pulse Secure Client for Windows Phone.....	269
Implementing Host Checker Policies for Pulse Secure Client for Windows Phone.....	270
PART 4 Windows In-Box Pulse Secure Client.....	271
CHAPTER 13 Windows Pro and Windows RT In-Box Pulse Secure Client.....	272
Pulse Secure VPN App for Chrome.....	272
Pulse Secure Universal App for Windows.....	272
<i>Microsoft Windows In-Box Pulse Secure Client Supported Platforms</i>	273
<i>Microsoft Windows In-Box Pulse Secure Client Supported Features</i>	273
<i>Microsoft Windows In-Box Pulse Secure Client Limitations</i>	274
Microsoft Windows In-Box Pulse Secure Client User Interface.....	274
Windows PowerShell Script Examples for Microsoft Windows In-Box Pulse Secure Client.....	279
Host Checker for Windows In-Box Pulse Secure Client Overview.....	281
Configuring Host Checker for Windows In-Box Pulse Secure Client.....	282
Implementing Host Checker Policies for Windows In-Box Pulse Secure Client.....	283
Host Checker Statement of Health for Pulse Connect Secure Overview.....	284
Configuring a Statement of Health Host Checker Policy for Pulse Connect Secure.....	284
Enabling Statement of Health on the Windows Endpoint for the Windows In-Box Pulse Secure Client.....	285
Viewing Windows In-Box Pulse Secure Client Log Messages.....	285
PART 5 Index.....	287

List of Tables

Table 1: Notice Icons	9
Table 2: Pulse Secure Documentation	9
Table 3: Pulse for Windows Connection Status.....	18
Table 4: Connection Status in the System Tray Icon.....	18
Table 5: Deployment Scenarios	27
Table 6: Pulse Icon States (Windows Tray and OS X Menu Bar)	34
Table 7: Pulse Secure Client Event Log Messages.....	36
Table 8: Configuration Options for Credential Provider Login.....	73
Table 9: Split Tunneling Options Summary	98
Table 10: Split Tunneling Deployment Scenarios.....	102
Table 11: Pulse/SAM Client Version Summary	120
Table 12: Variables Description	128
Table 13: Always-on options Settings	129
Table 14: Pulse Client Embedded Mini-Browser Settings	147
Table 15: Enabled and Disabled features	147
Table 16: Parameters details of URL	189
Table 17: Parameter details.....	190
Table 18: Pulse Launcher Arguments	199
Table 19: Pulse Launcher Return Codes.....	200
Table 20: Odyssey Access client and Pulse Secure Client Feature Comparison	220
Table 21: Network Connect and Pulse Secure Client Feature Comparison.....	223
Table 22: Pulse Split Tunneling.....	225
Table 23: User Agent String Client Type Pairings for Mobile Devices	228
Table 24: Pulse Secure Client for iOS Error Messages	238
Table 25: Pulse Secure Client for Android Error Messages.....	250
Table 26: Return Codes for doCommand.....	254
Table 27: Schema Options.....	281

About This Guide

- Objectives
- Audience
- Document Conventions
- Related Documentation
- Requesting Technical Support

Objectives

The *Pulse Secure Client Administration Guide* describes Pulse and includes procedures for network administrators who are responsible for setting up and maintaining network access using Pulse Secure client software.



Audience

The *Pulse Secure Client Administration Guide* is for network administrators who are responsible for setting up and maintaining network access using Pulse Secure client software. This guide describes the procedures for configuring Pulse Secure as the access client.

Document Conventions

Table 1 defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.

Related Documentation

Table 2 describes related Pulse Secure documentation.

Table 2: Pulse Secure Documentation

Title	Description
Pulse Secure for Mobile Devices	Describes how to set up and manage security on mobile devices.
Pulse Connect Secure	Describes how to configure and maintain Pulse Connect Secure.
Pulse Policy Secure	Describes how to configure and maintain Pulse Policy Secure.
SRX Series Services Gateways	Describes how to use and configure SRX Series Gateways running Junos OS.

Requesting Technical Support

Technical product support is available through the Pulse Secure Global Support Center (PSGSC). If you have a support contract, then file a ticket with PSGSC.

- Product warranties—For product warranty information, visit <https://www.pulsesecure.net/>

Self-Help Online Tools and Resources

For quick and easy problem resolution, Pulse Secure, LLC has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <https://www.pulsesecure.net/support>
 - Search for known bugs: <https://www.pulsesecure.net/support>
- Find product documentation: <https://www.pulsesecure.net/techpubs>
- Find solutions and answer questions using our Knowledge Base: <https://www.pulsesecure.net/support/>
 - Download the latest versions of software and review release notes: <https://www.pulsesecure.net/support>
 - Search technical bulletins for relevant hardware and software notifications: <https://www.pulsesecure.net/support>
 - Open a case online in the CSC Case Management tool: <https://www.pulsesecure.net/support>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool:
<https://www.pulsesecure.net/support>

Opening a Case with PSGSC

You can open a case with PSGSC on the Web or by telephone.

- Use the Case Management tool in the PSGSC at <https://www.pulsesecure.net/support>.
- Call 1- 844-751-7629 (toll-free in the USA).

For international or direct-dial options in countries without toll-free numbers, see <https://www.pulsesecure.net/support>

PART 1 Pulse Secure Client

- [Pulse Secure Client Overview](#)
- [Configuring Pulse Policy Secure](#)
- [Configuring Pulse Connect Secure](#)
- [Configuring Pulse Secure Client on SRX Series Gateways](#)
- [Session Migration](#)
- [Deploying Pulse Secure Client Software](#)
- [Customizing the Pulse Secure Desktop Client](#)

CHAPTER 1 Pulse Secure Client Overview

- [Introducing Pulse Secure Client](#)
- [Pulse Secure Client and Software Defined Perimeter \(SDP\)](#)
- [Pulse Secure Client Configuration Overview](#)
- [Pulse Secure Client Status Icons](#)
- [Installation Requirements](#)
- [Pulse Secure Client Error Messages Overview](#)
- [Accessing Pulse Secure Client Error Messages on macOS Endpoints](#)
- [Pulse Secure Client Log Files](#)
- [Deleting the Pulse Secure Client Log Files](#)
- [Migrating from Odyssey Access Client to Pulse Secure Client](#)
- [Migrating from Network Connect to Pulse Secure Client](#)
- [Predictable Pulse Server Hostname Resolution with IPv6](#)

Introducing Pulse Secure Client

Pulse Secure client is an extensible multiservice network client that supports integrated connectivity and secure location-aware network access. Pulse Secure client simplifies the user experience by letting the network administrator configure, deploy, and control the Pulse client software and the Pulse connection configurations that reside on the endpoint.

Pulse Secure comprises client and server software. The client enables secure authenticated network connections to protected resources and services over local and wide area networks. The Pulse Secure client software can connect with Pulse Connect Secure to provide remote access to enterprise and service provider networks. Pulse also delivers secure, identity-enabled network access control (NAC) for LAN-based network and application access when it is deployed with Pulse Policy Secure. Pulse also integrates with Pulse Collaboration Suite for online meeting services.

Users of mobile devices (smart phones and tablets) can install the Pulse mobile device app from the respective app stores for secure connectivity to Pulse Connect Secure. Windows 8.1 (Pro and RT) introduced a Pulse Secure VPN client as part of the operating system.

Pulse Secure Client for Windows

The Pulse Secure client for Windows user interface (see [Figure 1](#)) lists the deployed Pulse connections. Each connection is a set of properties that enables network access through a specific Pulse server. The user can expand a connection to see more details about the connection.



Note: From 5.3R2, the Pulse Client connects to PSA device through proxy at the first attempt and then try connecting directly upon failure.

VPN Connection Details

To view the VPN connection details:

1. Click the VPN connection to select it.
2. Click **File > Connections**. Refer to the following figure.

Figure 1: Pulse Secure Client for Windows - VPN Connection Details

To view the Advanced Connection Details dialog:

1. Click the VPN connection to select it.
2. Click **File > Connections > Advanced Connection Details**.

The connection detail information is not updated automatically. For example, the session time remaining shows how much time remains when you open the dialog. To update advanced detail information, click **Refresh** or click the check box labeled automatically refresh.

Figure 2: Advanced Connection Details - VPN



The Advanced Connection Details window gives the following information:

Field Name	Description
Session time remaining	The duration that the current VPN session will remain active before credentials must be re-entered or the session manually extended.
Session Duration	
Tunnel type	This describes that the connection is a VPN tunnel.
VPN type	The protocol used to create the tunnel (SSL or ESP).
Assigned IPv4	The IPv4 address assigned to the Pulse virtual adapter.
Bytes in	Number of bytes received through the tunnel.
Bytes out	Number of bytes sent through the tunnel.

Field Name	Description
Connection Source	<p>This describes how the Pulse client received the connection entry:</p> <p>If the value is Preconfigured, then the connection entry came from a Connection Set that was downloaded from a gateway.</p> <p>If the value is User-added, then the connection entry was manually added by the end user with the Pulse client UI.</p> <p>And if the value is Dynamic, then it means that the connection entry was resulted from launching the Pulse client by connecting a web browser to a Pulse Secure gateway and pressing the “Start” button on the web page.</p>

PSAM Connection Details

To view the PSAM connection details:

1. Click the connection to select it.
2. Click **File > Connections**. Refer to the following figure.

Figure 3: Pulse Secure Client for Windows – PSAM Connection Details



To view the Advanced Connection Details dialog:

1. Click the connection to select it.
2. Click **File > Connections > Advanced Connection Details**.

The connection detail information is not updated automatically. For example, the session time remaining shows how much time remains when you open the dialog. To update advanced detail information, click **Refresh** or click the check box labeled automatically refresh.

Figure 4: Advanced Connection Details - PSAM



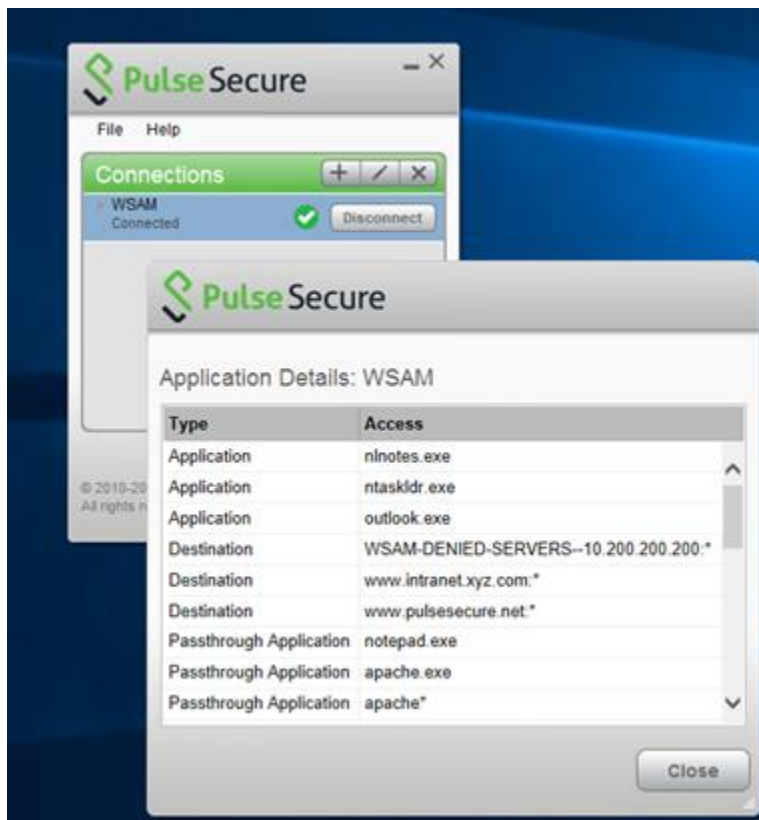
The Advanced Connection Details window gives the following information:

Field Name	Description
Session time remaining	The duration that the current VPN session will remain active before credentials must be re-entered or the session manually extended.
Session Duration	
Tunnel type	This describes that the connection is a port/application mapping through SAM (Secure Access Manager).
VPN type	The protocol used to create the tunnel (SSL or ESP).
Bytes in	Number of bytes received through the tunnel.
Bytes out	Number of bytes sent through the tunnel.

To view the Application Details:

1. Click the PSAM connection to select it.
2. Click **File > Connections > Application Details**. Refer to the following figure.







Figure 5: Application Details - PSAM



Pulse Secure client also displays a system tray icon that provides connection status, and can allow the user to connect and disconnect and enables quick access to the program interface. One tray icon provides status for all active connections.








Typically, the network administrator defines and deploys the Pulse connections, but you can also enable users to define, edit, and remove their own connections.

Table 3: Pulse for Windows Connection Status

	Connected.
	Connecting.
	Connected with limitations
	Connection attempt failed.
	Connection suspended.
	Connected to the local network but no Internet access available. Public WiFi locations often deploy a captive portal that requires the user to enter authentication information or to accept terms of service before network access is granted. Pulse Secure client detects the presence of captive portals and does not initiate a connection to a Pulse Secure client server until Internet access is granted.

Pulse Secure client supports the Federal Information Processing Standard (FIPS), which defines secure communication practices for the U.S. government. If FIPS is enabled on the endpoint, **FIPS On** appears near the bottom the Pulse window. A single system tray icon indicates the status of all active Pulse Secure client connections. You can right-click the system tray icon to control Pulse Secure client connections, to access Pulse Collaboration Suite meeting functions, to open the Pulse Secure client interface, or to exit from Pulse Secure client. The following table shows the connection status indicated by the system tray icon.

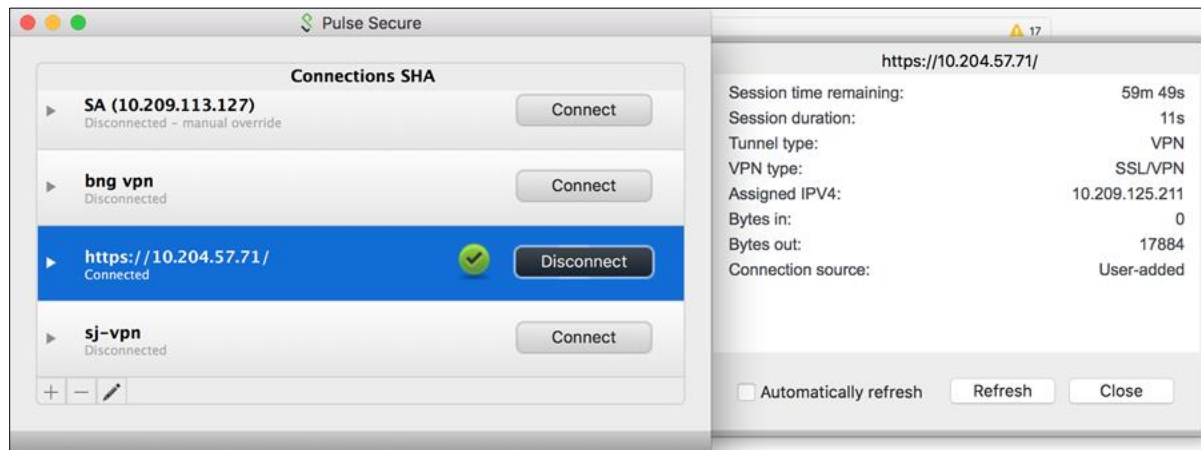
Table 4: Connection Status in the System Tray Icon.

	No connection
	Connecting. A connection stays in this state until it fails or succeeds.
	Suspended
	Connected with issues
	Connection failed
	Connected
	Connected to the local network but no Internet access available. Public Wi-Fi locations often deploy a captive portal that requires the user to enter authentication information or to accept terms of service before network access is granted. Pulse Secure client detects the presence of captive portals and does not initiate a connection to a Pulse Secure client server until Internet access is granted.

Pulse Secure Client for macOS

Pulse supports Apple computers running macOS. You deploy Pulse to Mac endpoints the same way you deploy the Windows client. [Figure 6](#) shows the Pulse for Mac client interface.

Figure 6: Pulse Secure client for Mac client Interface



Pulse for Mac endpoints supports the following:

- Connections to Pulse Policy Secure
- Connections to Pulse Connect Secure

Pulse clients connect to the Pulse Connect Secure in SSL fallback mode.

- Connections to Juniper Networks SRX Series gateways.
 - macOS endpoints can connect to SRX Branch series SRX100-SRX650 gateways that are running a Junos OS release between v10.2 and v12.3, and that have dynamic VPN access enabled and configured. SRX gateways do not support deployment of the Pulse Secure client.
 - Requires Pulse Secure client for macOS 5.0R3 or later and macOS 10.8 or later.
 - Pulse Secure for Mac clients connect to the gateway as an IPsec IKEv1 VPN connection.
 - Pulse Dynamic VPN functionality is compatible with SRX-Branch (SRX100-SRX650) devices only. SRX Data Center (SRX1400-SRX5800 – also called SRX HE or High End) devices do not support Pulse Dynamic VPN from either Windows or Mac clients.
 - On macOS clients, Pulse IPsec connections to SRX are unable to use the DNS IP address supplied by the SRX.
- Host Checker

Host Checker for macOS supports the following rules and remediation actions:

- Port
- Process
- File
- Custom IMC
- Enable Custom Instructions
- Kill Processes
- Delete Files
- Send reason strings

User Experience

From the user perspective, Pulse Secure client presents a clean, uncomplicated interface. The user can enter credentials, select a realm, save settings, and accept or reject the server certificate. When you configure the client, you can specify whether to permit end users to modify settings, such as by adding connections.

Security Assertion Markup Language (SAML) Authentication

Pulse environment can do SAML authentication for a Single Sign-on (SSO) in following two ways:

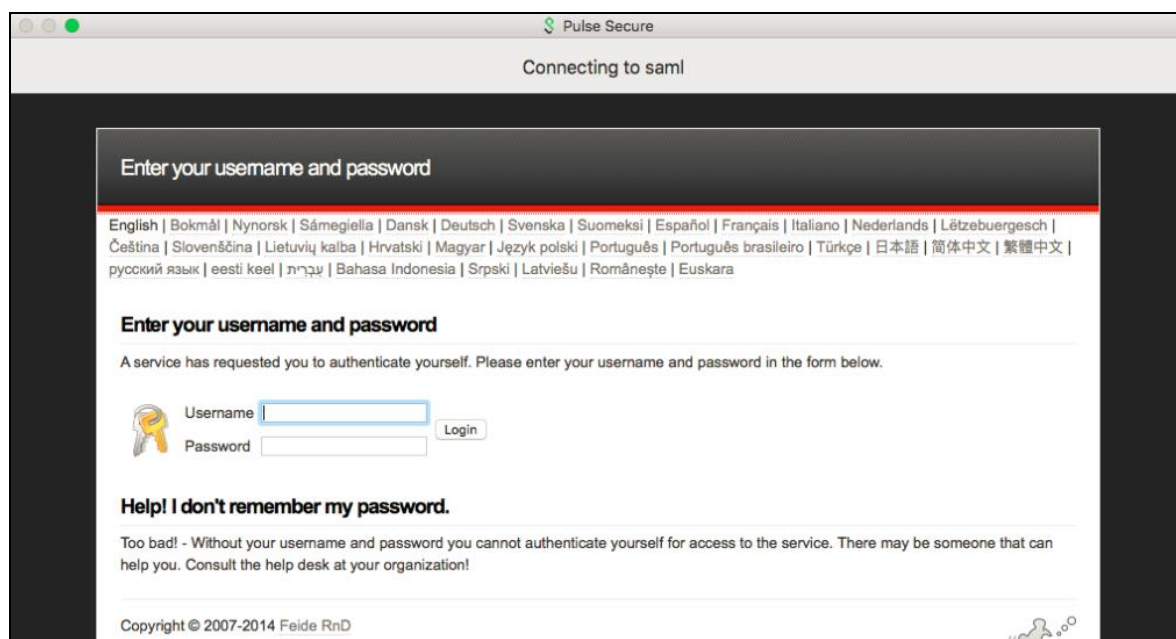
- Pulse user sees an embedded browser (Refer [Figure 7](#)) - if **Enable embedded browser for authentication** is enabled in [Pulse Secure Connection Set Options](#).

Pulse client will close the embedded browser, once the SAML authentication is done.



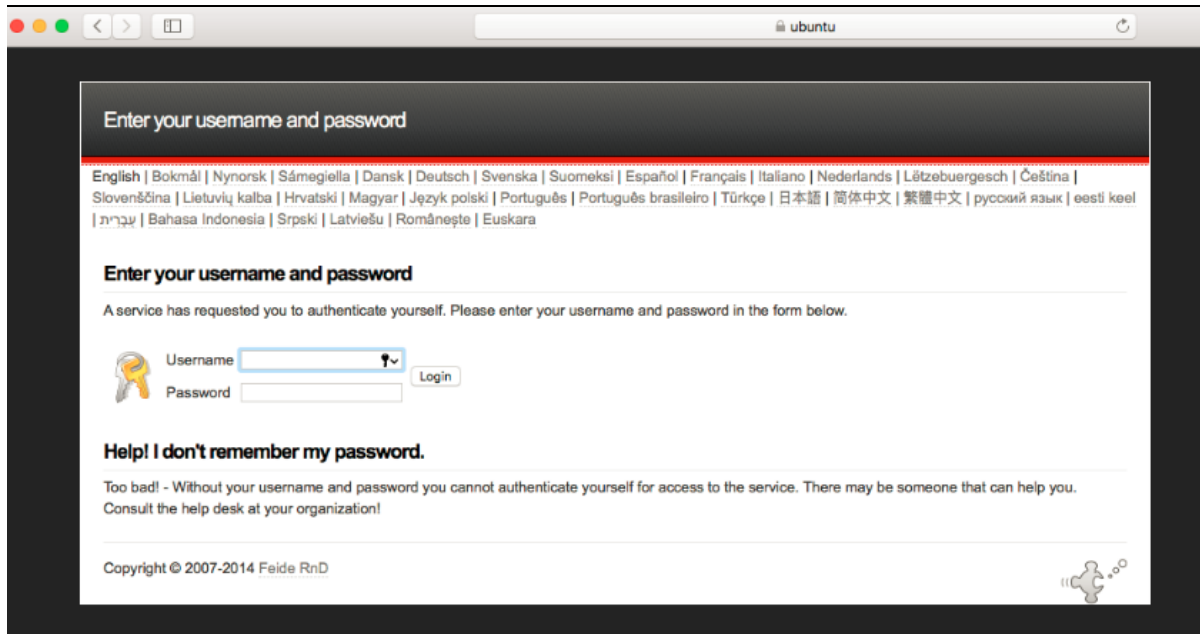
Note: If user resizes the Embedded browser window, size will remain same even if user reconnects to Pulse Desktop Client. Embedded browser window size will remain as pre-selected size which was set by the user for the first time, until user resizes it again.

Figure 7: SAML Authentication with Embedded browser



- Pulse user sees an external browser (Refer [Figure 8](#)). if **Enable embedded browser for authentication** is disabled in [Pulse Secure Connection Set Options](#).

Figure 8: SAML Authentication (External Browser)



Custom Sign-in Page in Embedded browser

To upload custom sign-in page in Pulse Desktop Client, admin needs to perform the following steps:

1. Login in to PCS/PPS as admin.
2. Go to **Authentication > Signing-In > Sign-In Pages > Upload Custom Sign-In Pages**.
3. Select the option **"Use Custom Page for the Pulse Desktop Client Logon"**.

Figure 9: Use Custom Page for the Pulse Desktop Client Logon - PCS

Pulse Secure System **Authentication** Administrators Users Maintenance Wizards

Signing In > Sign-In Pages > Upload Custom Sign-In Pages

Upload Custom Sign-In Pages

Custom sign-in pages allow you to provide customized templates for various pages that may appear during the sign-in process. Refer to the documentation for information about creating valid templates.

Sign-In Pages

Name:
Label to reference the custom sign-in pages.

Page Type: ☒ Access ☐ Meeting

☒ **Use Custom Page for the Pulse Desktop Client Logon**
The Pulse Desktop Client will open a web browser and use custom pages for authentication instead of standard login prompts.

☐ **Prompt the secondary credentials on the second page**
These labels appear when a realm using this sign-in page specifies a secondary authentication server that requires user input. These are only applicable to user sign-in pages.
 This option is not applicable when TOTP authentication server is selected as secondary auth-server for this realm, in which case token input from user is always taken from the second page.

Templates File: RSA_RBA_authentication.zip
Zip file containing the custom templates and assets.

☐ Skip validation checks during upload

Figure 10: Use Custom Page for the Pulse Desktop Client Logon - PPS

Pulse Secure System **Authentication** Administrators Users Endpoint Policy Maintenance Wizards

Signing In > Sign-In Pages > Upload Custom Sign-In Pages

Upload Custom Sign-In Pages

Custom sign-in pages allow you to provide customized templates for various pages that may appear during the sign-in process. Refer to the documentation for information about creating valid templates.

Sign-In Pages

Name:
Label to reference the custom sign-in pages.

Page Type: ☒ Access

☒ **Use Custom Page for the Pulse Desktop Client Logon**
The Pulse Desktop Client will open a web browser and use custom pages for authentication instead of standard login prompts.

☐ **Prompt the secondary credentials on the second page**
These labels appear when a realm using this sign-in page specifies a secondary authentication server that requires user input. These are only applicable to user sign-in pages.
 This option is not applicable when TOTP authentication server is selected as secondary auth-server for this realm, in which case token input from user is always taken from the second page.

Templates File: No file chosen
Zip file containing the custom templates and assets.

☐ Skip validation checks during upload

4. Click **Browse** and select the custom sign-in page file and click **Upload Custom Pages**.
5. Go to **Signing In > Sign-In Policies > New Sign-In Policy** to create the new Sign-In policy.
6. Under Sign-In page, select the uploaded custom page ([step-2](#)) from the drop-down box to associate custom Sign-In page with the Sign-In Policy.

Figure 11: Use Custom Page for the Pulse Desktop Client Logon – PPS

Pulse Secure System **Authentication** Administrators Users Endpoint Policy Maintenance Wizards

Spring In > Sign-In Policies > New Sign-In Policy

New Sign-In Policy

User type: ☒ Users ☐ Administrators

Sign-in URL: Format: https://path/ Use "x" as wildcard in the beginning of the host name.

Description:

Sign-in page: Sign-in pages

Authentication realm

Specify what realms will be available when signing in:

Available realms	Authentication protocol set	
<input type="text" value="Cert Auth"/>	<input type="text" value="- Not applicable -"/>	<input type="button" value="Add"/>

If more than one realm appears above, Onlyway Access Client in the Policy Secure sign-in page will ask the user to choose. Other endpoints cannot choose a realm, the Policy Secure will assign the first available realm from the list. If no realm appear above, sign-in will fail.

☐ **User may specify the realm name as a Username suffix**
When this option is selected, the Username suffix will be used to specify a realm.

☐ **Remove realm suffix before passing to authentication server**
When this option is selected, the username suffix will be stripped from the Username prior to authenticating with an authentication server.

☒ **Fail if suffix does not match any of the realms**
When this option is selected, the user should provide one of the realms suffix. If not, the user will be denied sign-in.

Configure Guest Settings

☐ Use this sign policy for Guest and Guest admin to use specific pages.

Configure Sign Notifications

Pulse environment can open Custom Sign-In page in following two ways:

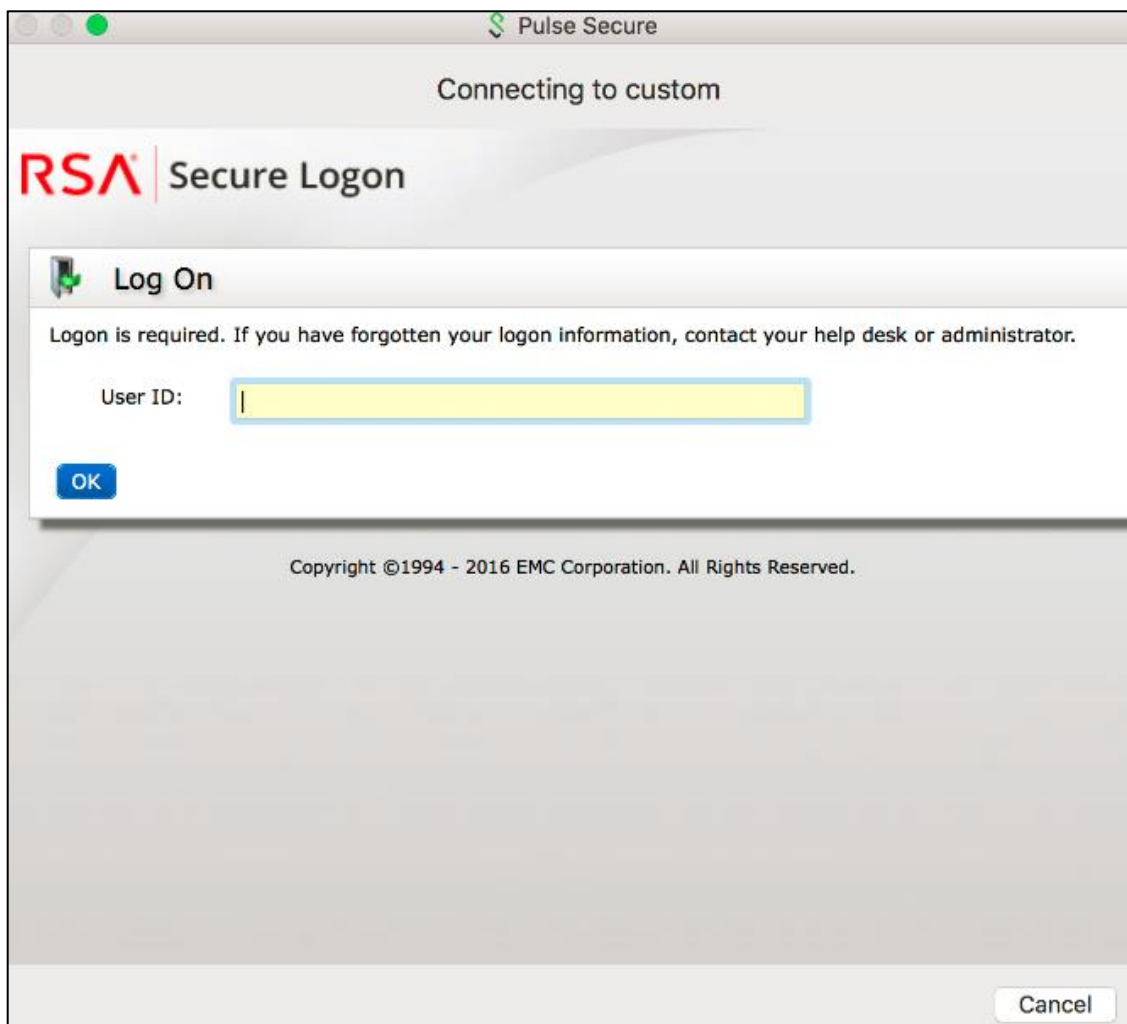
- Pulse user sees an embedded browser (Refer [Figure 12](#)) - if **Enable embedded browser for authentication** is enabled in [Pulse Secure Connection Set Options](#).

Pulse client will close the embedded browser once the authentication is done.

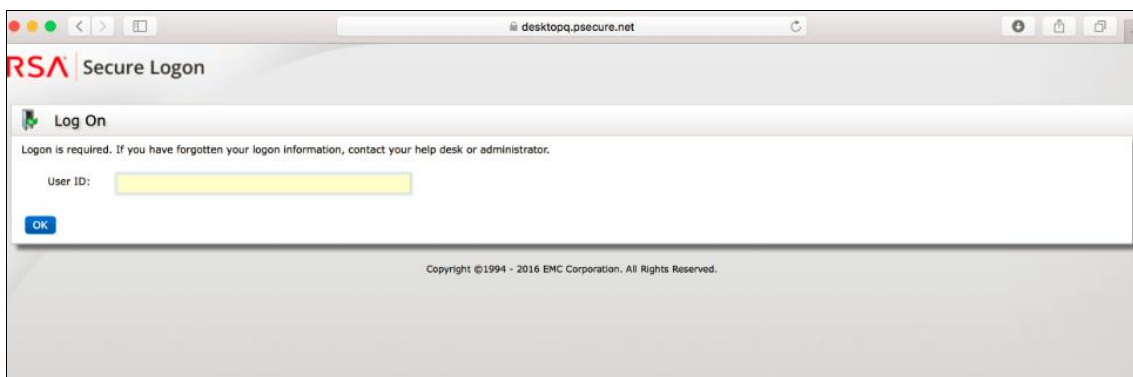


Note: If user resizes the Embedded browser window, size will remain same even if user reconnects to Pulse Desktop Client. Embedded browser window size will remain as pre-selected size which was set by the user for the first time, until user resizes it again.

Whenever user logs into the custom sign-in URL from Pulse Desktop Client, embedded browser will be launched with custom sign-in pages uploaded into it.

Figure 12: Custom Sign-In page support for Embedded browser

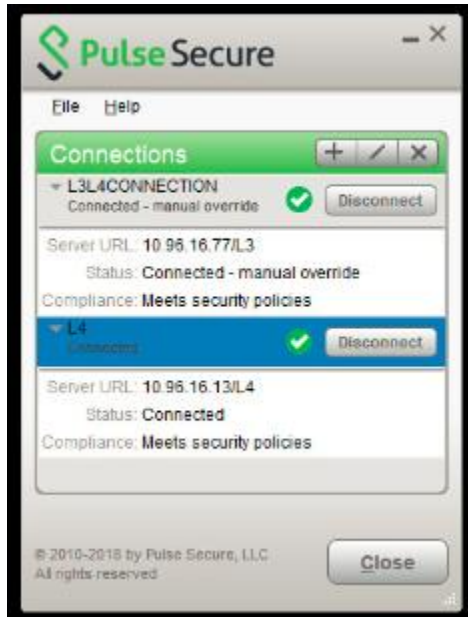
- Pulse user sees an external browser (Refer [Figure 13](#)). if **Enable embedded browser for authentication** is disabled in [Pulse Secure Connection Set Options](#).

Figure 13 : Custom Sign-In page uploading in External browser

L3 and Pulse SAM Coexistence

L3 and Pulse SAM coexistence (supported on Windows only) enables the user to establish Layer 3 connection to Pulse Connect Secure and Pulse SAM connection simultaneously (refer [Figure 14](#)). This feature is available from 9.0R3 onwards.

Figure 14: L3 and PSAM Connection Coexistence



To achieve, L3 and PSAM coexistence, Pulse desktop client should have minimum two Pulse Connect Secure connections, each for L3 and PSAM. Also, maximum three active user connections are allowed at once.

Limitation for L3 and Pulse SAM coexistence:

- At any given point, for any user only one L3 and one L4 is supported.

With L3 and PSAM coexistence, the way the packet is tunneled, depends on how the L3 and PSAM tunnel are configured. It can be done in following two ways:

Following are the 2 scenarios, where L3 and PSAM coexistence is supported.

Scenario-1: PSAM is behind L3

PCS1 has L3 tunnel configuration and PCS2 is behind PCS1.

If specific set of resources is not accessible on PCS1 server and needs to access from PCS2 server, which is accessible through PCS1 server, then additional authentication is needed to access PCS2 server. As access to PCS2 server is possible only after making connection to PCS1 server, it is the case of PSAM tunnel inside L3 tunnel.

Scenario-2: L3 and PSAM are independent

PCS1 has L3 tunnel configuration and PCS2 has Pulse SAM configuration.

L3 Connection for Pulse Connect Secure is established, split tunneling should be enabled and exclude the PCS2 IP from the split tunneling networks.

If single user needs to access two different set of resources available on PCS1 and PCS2, then one specific set of resources is under PCS1 and another set of resources is under PCS2.

As PCS1 and PCS2 are at different locations and user can not establish two L3 connections to access both set of resources on PCS1 and PCS2, so PSAM can provide the secure access to set of resources on PCS2.



Note: L3 based FQDN Split Tunneling feature with PSAM coexistence is not supported.

HVCI Compatibility

The new Pulse Desktop Client on Windows is now compatible with Microsoft Windows 10 HVCI settings. Windows 10 HVCI settings are part of Windows Device Guard security features for mitigating cybersecurity threats. When HVCI is enabled, Windows OS performs code integrity checks and allows only secured applications. Pulse Desktop Client on Windows is compatible with these settings which would help customers adopt the latest security features of Windows.

Pulse SAM IPv6 Support

Pulse SAM IPv6 support is available for Windows 7, Windows 8.1 and Windows 10.

Internet Protocol Version 6 (IPv6) is the protocol designed to succeed Internet Protocol Version 4 (IPv4). From 9.1R1 release onwards, Pulse SAM (PSAM) will support IPv6 Pulse SAM tunneling along with IPv4 Pulse SAM tunneling with the help of new option for internet traffic filtering, Windows Filtering Platform (WFP) driver.

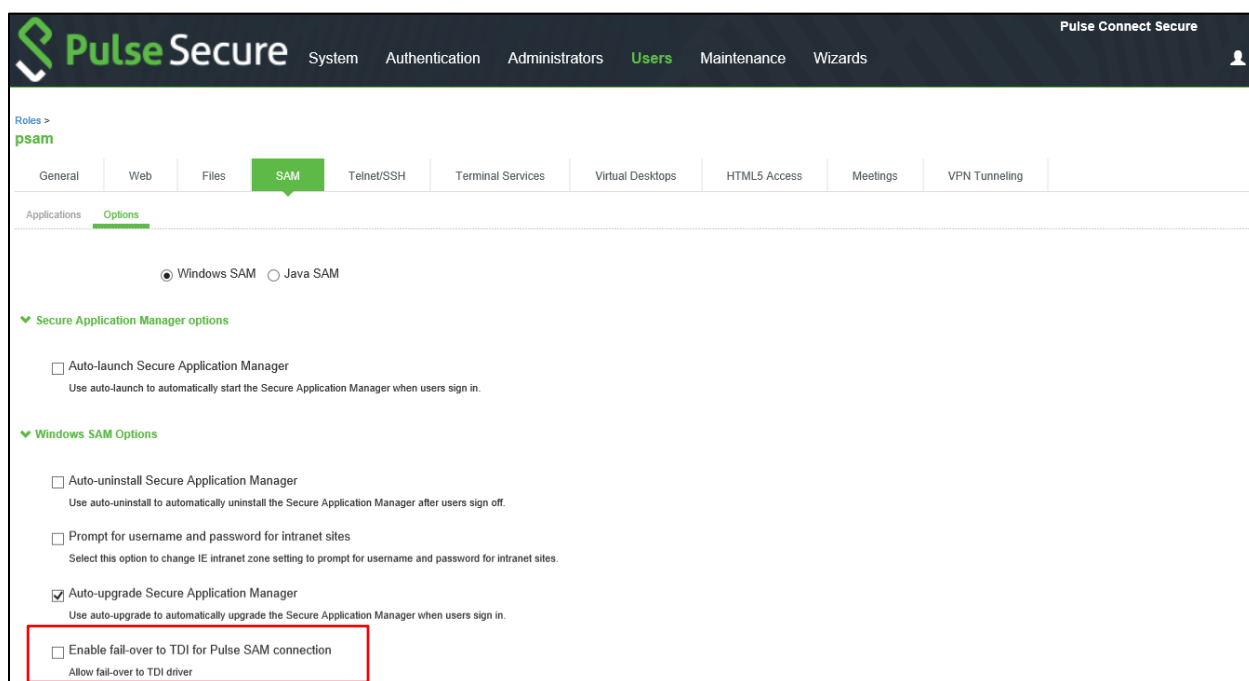
WFP driver supports both IPv6 and IPv4, however TDI driver supports only IPv4. WFP driver allows the user to provide a deeper inspection and control of packets by modifying or examining TCP/IP traffic at any TCP/IP stack layer.

Administrator can switch from WFP driver (supporting both IPv6 and IPv4) to classic TDI driver (supporting IPv4 only) with fallback mechanism, in case of any issue due to WFP driver installation.

Following are the steps to switch from WFP to TDI:

1. Go to **Users > User Role**.
2. Select the role.
3. Go to **SAM > Options**. Following screen appears:
Check the checkbox **"Enable fail-over to TDI for Pulse SAM connection"**.

Figure 15: Enable fail-over to TDI for Pulse SAM connection



Benefits:

Following are the benefits of this feature:

- PSAM will be able to filter the traffic from Windows 10 and Windows 8.1 Metro Mode Applications.
- PSAM will be able to filter the traffic from Internet Explorer 11 with Enhanced Protected mode.
- PSAM will support Dual Stack (both IPv6 and IPv4).

Deployment Scenarios:

The following table summarizes the IPv6 in IPv6, IPv4 in IPv6 and IPv6 in IPv4 scenarios:

Table 5: Deployment Scenarios

PDC	Endpoint	PCS External Interface	PCS Internal Interface	Tunnel	Description of the Connection
Dual Stack or IPv6 only	Dual Stack (IPv6 and IPv4) or IPv6 only	IPv6	Dual Stack or IPv6 only	IPv6-in-IPv6	IPv6 resource on IPv6 PSAM session.
Dual Stack or IPv6 only	Dual Stack (IPv6 and IPv4) or IPv6 only	IPv6	IPv4	IPv4-in-IPv6	IPv4 resource on IPv6 PSAM session.
Dual Stack or IPv4 only	Dual Stack (IPv6 and IPv4) or IPv6 only	IPv4	Dual Stack or IPv4	IPv6-in-IPv4	IPv6 resource on IPv4 PSAM session.



Note: Pulse Desktop Client 9.0R1 Pulse SAM connection fails with Pulse Connect Secure 9.1R1 version. For more details, refer to 9.1R1 Pulse Secure Desktop Client Release Notes document on the following website:
<https://www.pulsesecure.net/techpubs>.

Location Awareness

The location awareness feature enables you to define connections that are activated automatically based on the location of the endpoint. Pulse determines the location of the endpoint by evaluating rules that you define. For example, you can define rules to enable Pulse Secure client to automatically establish a secure tunnel to the corporate network through Pulse Connect Secure when the user is at home, and to establish a Pulse Policy Secure connection when the user is in the office and connected to the corporate network over the LAN. Pulse does not re-establish a VPN tunnel when the endpoint re-enters the trusted/corporate network. Location awareness rules are based on the client's IP address and network interface information.

Centralized Pulse Configuration Management

Centralized configuration management is a key feature of Pulse Secure client. Pulse connection sets (the configurations that define how and when a Pulse client connects), are *bound* to a particular Pulse server. The binding server is the one that provides the initial Pulse configuration to the Pulse client. For example, if you create a Pulse connection set on Server A, and then distribute those connections to endpoints, those clients are bound to Server A.

A bound client is managed by its particular Pulse server. The Pulse administrator defines the Pulse Secure client connections and software components that are installed on the endpoint. When the Pulse client connects to the Pulse server that is managing it, the server automatically provisions configuration and software component updates. The administrator

can permit the user to add, remove, and modify connections. The administrator can also allow dynamic connections (connections that are added by Pulse servers when the user logs into the server using a browser). A dynamic connection enables a bound client to add connections from Pulse servers other than the one the client is bound to. Dynamic connections are created as manual rather than automatic connections, which means that they are run only when the user initiates the connection or the user browses to a Pulse server and launches Pulse from the server's Web interface. Dynamic connections create the connection with the minimum configuration required to make the connection, which means that the URL used to install or launch Pulse from the Pulse server's Web interface is used as the Connection URL and connection name. Binding Pulse Secure clients to a particular server ensures that the client does not receive different configurations when it accesses other Pulse servers. A bound endpoint receives connection set options and connections from its binding server, but it can have its Pulse client software upgraded from any Pulse server that has the automatic upgrade option enabled. (SRX gateways do not support Pulse software updates.)



Note: A Pulse Secure client can be bound to only one Pulse server connection set at a time. The Pulse Secure client can receive updates and changes to that bound connection set from other Pulse servers only if the connection set is exported from the Pulse server and then imported to another Pulse server.

A Pulse client does not need to be bound to a Pulse server. An *unbound client* is managed by its user. If the Pulse Secure client software is installed without any connections, the user must add connections manually. Dynamic connections can be added by visiting the Web portals of Pulse servers. An unbound client does not accept configuration updates from any Pulse server.

“Adding a Pulse Configuration to a New Pulse Installation” explains the binding process in more detail.

Session Migration

If you configure your access environment to support the Pulse Secure client session migration feature, users can log in once through a Pulse server on the network, and then securely access additional Pulse servers without needing re-authentication. For example, a user can connect from home through Pulse Connect Secure, and then arrive at work and connect through Pulse Policy Secure without having to log in again. Session migration also enables users to access different resources within the network without repeatedly providing credentials. IF-MAP Federation is required to enable session migration for users.

Smart Connections - List of URLs

Each Pulse connection that connects to Pulse Policy Secure or Pulse Connect Secure can be configured with a list of Pulse servers. The Pulse client attempts to connect to each of the servers in the URL list until it succeeds. You can choose different modes to control the behavior of a Pulse connection that is starting from a disconnected state, start at the top of the list, start with the most recently connected URL, or choose randomly. The random option helps distribute the connection load across different Pulse servers. If a Pulse connection that is already established gets disconnected, for example, the wireless connection is interrupted, Pulse always tries to connect to the most recently connected URL. If that connection fails, then Pulse uses the server list. The Pulse user can also choose a connection from the list as shown in [Figure 16](#).

Figure 16: Pulse for Windows client with a List of Connection URLs



Security Certificates

Users cannot add CA servers or manage the server list. The Pulse client handles certificates in the same way that a browser handles certificates. If the Pulse dynamic certificate trust option is enabled for a connection, the user can accept or reject the certificate that is presented if it is not from a CA that is defined in the endpoint's certificate store.

Compliance and Remediation

Pulse supports the Host Checker application to assess endpoint health and update critical software. Host Checker is a client-side agent that is based on Trusted Network Connect standards. You configure rules in Host Checker policies for Pulse Connect Secure and Pulse Policy Secure to specify the minimum criteria for the security compliance of endpoints that are allowed to enter the network. Endpoints that fail can be connected through a remediation role that provides limited access.

Host Checker can be deployed from a Pulse server to Pulse clients on Windows and macOS endpoints. It will be downloaded and run when a browser is used on a Windows or macOS endpoint to connect to the Pulse server Web portal. You can use Host Checker policies at the realm or role level.

Host Checker for mobile clients (iOS, Android, and Windows Phone) is included as part of the Pulse app. Host Checker runs on the mobile client if Host Checker policies are configured and enabled on the server.



Note: Checker is not supported in the use case where the user employs a browser on the mobile device to connect to the Pulse server Web portal.

For Windows and OS X clients, you can use Host Checker to perform the following:

- **Virus signature monitoring**
You can configure Host Checker to monitor and verify that the virus signatures, operating systems, and software versions installed on client computers are up to date. You can configure automatic remediation for those endpoints that do not meet the specified criteria.
- **Patch management information monitoring and patch deployment**
You can configure Host Checker policies that check for Windows endpoints' operating system service pack, software version, or desktop application patch version compliance.
- **Patch verification remediation options**

Pulse and Host Checker support endpoint remediation through Microsoft System Management Server or Microsoft System Center Configuration Manager (SMS/SCCM). With SMS/SCCM, Pulse triggers a preinstalled SMS/SCCM client to get patches from a pre-configured server.
- **Endpoint configuration**

You can configure custom rules to allow Host Checker to check for third-party applications, files, process, ports, registry keys, and custom DLLs.

Pulse mobile clients support a set of Host Checker functions that vary from one OS to the next. For complete information on Host Checker for mobile clients, see [“Implementing Host Checker Policies for Pulse Secure Client for iOS Devices”](#), [“Implementing Host Checker Policies for Pulse Secure Client for Android Clients”](#), and [“Host Checker for Pulse Secure Client for Windows Phone”](#).

Two Factor Authentication

Pulse supports RSA SecurID authentication through soft token, hard token, and smart card authenticators. The SecurID software (RSA client 4.1 and later) must already be installed on the client machine.

Captive Portal Detection

Public WiFi locations often deploy a captive portal that requires the user to enter authentication information or to accept terms of service before network access is granted. Pulse detects the presence of captive portals and does not initiate a connection to a Pulse Connect Secure or Policy Secure server until internet access is granted. Pulse displays appropriate status information to enable the user to establish the portal and network connections.

Captive portal detection notes:

- Captive portal detection is supported on Pulse for both Windows and macOS. Captive portal detection is not supported on Windows In-Box Pulse Secure client or Pulse Secure client for mobile devices.
- If Pulse connects through a proxy in Captive Portal scenario, the captive portal detection algorithm is disabled and Pulse client tries connecting directly to PCS.
- SRX connections do not support captive portal detection.

Pulse Collaboration Suite Integration

Pulse Collaboration Suite is accessible through the Pulse interface on Windows, macOS, Android, and iOS. (Android clients must be R4.0 or later. iOS clients must be R3.2 or later.) Pulse Collaboration Suite enables users to schedule and attend secure online meetings. In meetings, users can share their desktops and applications with one another over a secure connection. Meeting attendees can collaborate by enabling remote control of their desktops and through text chatting.

Sign In Notifications

The notifications feature on Pulse Connect Secure and Pulse Policy Secure allows the network administrator to display notifications to Pulse client users prior to the user logging in and after the user has already logged in. For example, you could display a legal statement or a message stating who is allowed to connect to the server before you display the Pulse credentials dialog. After the user has connected, you could display a message that notifies the user of scheduled network or server maintenance or of an upcoming company meeting.

Automatic Software Updates

After you deploy Pulse Secure client software to endpoints, software updates occur automatically. If you upgrade the Pulse Secure client configuration on the server, updated software components are pushed to a client the next time it connects. You can disable this automatic upgrade feature.



Note: The automatic update feature is supported on Pulse Connect Secure and Pulse Policy Secure servers only. SRX gateways do not support automatic Pulse software updates.



Note: If you configure Pulse Secure client to make 802.1X-based connections, a reboot might be required on Windows endpoints.

Pulse Secure Client Customization and Rebranding

The Pulse Secure client customization tool (**BrandPackager**) enables you to customize the appearance of the Pulse Secure Windows and Apple OS X clients. You can add your own identity graphic to the Pulse splash screen, to the program interface, and to Windows credential provider tiles. [Figure 17](#) shows graphic customizations applied to the Pulse for Windows client. You can also customize error and informational message text, the text that appears in dialog boxes and on buttons, and make limited changes to Pulse online Help. For example, you might want to add your help desk phone number to Pulse error messages and the Pulse online Help.

BrandPackager is available for download from pulsesecure.net.

Figure 17: Pulse Secure Client Interface and Splash Screen with Branding Graphics



Related Documentation

- [Pulse Secure Client Status Icons](#)
- [Pulse Secure Client for Mobile Devices Overview](#)
- [Customizing Pulse Secure Client Overview](#)

Pulse Secure Client and Software Defined Perimeter (SDP)

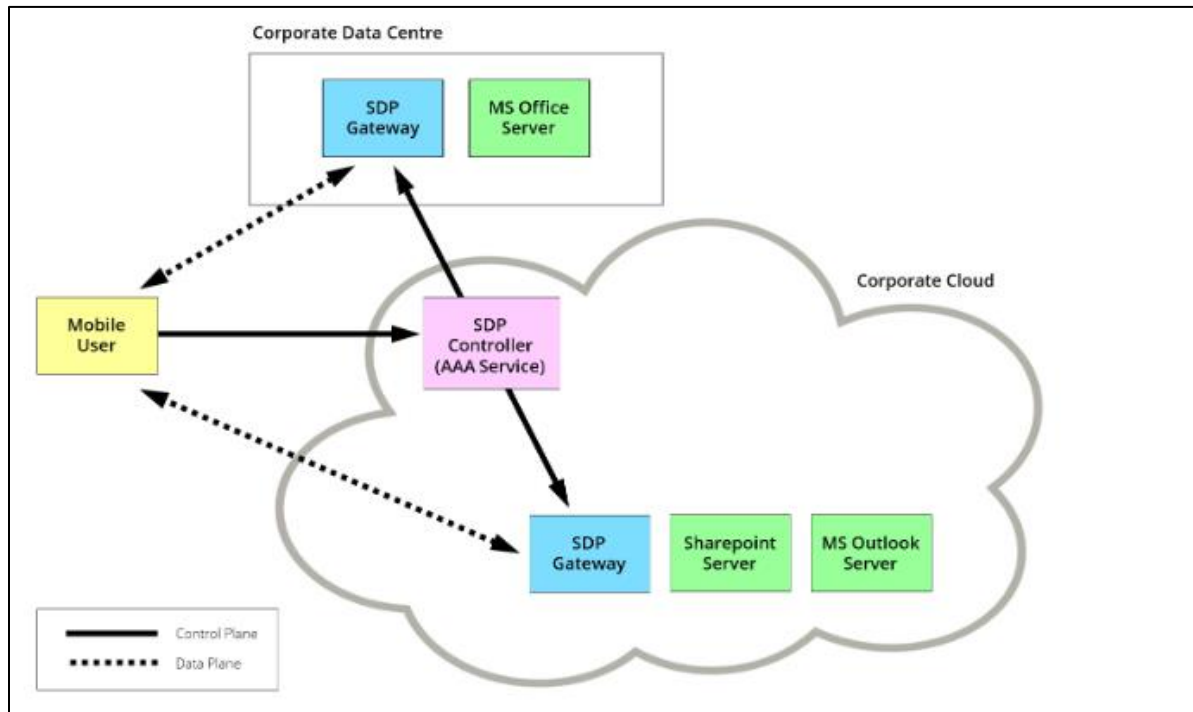
Traditional network-based security (Network Defined Perimeter) architectures use firewalls on the network perimeter to limit access to public IP addresses. This exposes the network to a variety of network-based attacks.

Connectivity in a Software Defined Perimeter (SDP) system is based on a need-to-know model, in which mobile devices are verified and authorized before access to application infrastructure is granted. Application infrastructure cannot be detected remotely and has no visible DNS information or exposed IP addresses. This protects networked resources from many common network-based attacks.

Pulse Secure SDP uses PCS appliances which individually act as either an SDP controller or an SDP gateway. Mobile users of the Pulse Secure Client perform authentication on an SDP controller which runs an Authentication, Authorization and Accounting (AAA) Service. The SDP controller then enables direct communication between the user and the SDP gateways that protect the user's authorized resources and enables requested encryption. This does not require the general exposure of public IP addresses. It also separates the control plane and the data plane.

Pulse Secure SDP supports a number of networks topologies, and can include both cloud-based and data center-based resources. For example:

Figure 18: Software Defined Perimeter Example



Note: For full details of installation and configuration of SDP, see the Software Defined Perimeter documentation on <https://www.pulsesecure.net/techpubs>.

Pulse Secure Client Configuration Overview

You configure the Pulse Secure client settings on the Pulse server so that when users request authentication, they are assigned a role based on the role mappings and optional security profile that you create. Access to specific resources is permitted only for users and devices that provide the proper credentials for the realm, that are associated with the appropriate roles, and whose endpoints meet security restrictions. If a user attempts to connect to the network from an endpoint that does not comply with the security restrictions you have defined, the user cannot access the realm or role.

As you plan your Pulse configuration, be sure you know how you want to deploy Pulse Secure client. You can use one or more of the following Pulse Secure client deployment options:

- Use the defaults or make changes to the Pulse Secure client default component set and default connection set, and then download and distribute Pulse by having users log in to the gateway's user Web portal and be assigned to a role. After the installation is complete, users have all the connections they need to access network resources.
- Create connections that an endpoint needs for connectivity and services, download the Pulse settings file (.pulsepreconfig), download default Pulse .msi installation program, and then run the .msi installation program by using an msiexec command with the settings file as an option. You can use the msiexec command to deploy Pulse using a standard software distribution process, such as SMS/SCCM.
- Distribute Pulse Secure client with no preconfiguration. You can download the default Pulse Secure Desktop Client installation file (Mac or Win) from the device, and then distribute the file to endpoints using your organization's standard software distribution methods. Because the installer does not contain preconfigured connections, users must define

network connections manually. Or you can create dynamic connections on each access gateway. These connections are automatically downloaded to the installed Pulse client when users provide their login credentials to the gateway's user Web portal.

The following tasks summarize how to configure Pulse Secure client on the device:

- Create and assign user roles to control who can access different resources and applications on the network. If you are converting your access environment from agentless or a VPN Tunneling environment, you should create new roles that are specific for Pulse Secure client.
- Define security restrictions for endpoints with Host Checker policies.
- Define user realms to establish authentication domains. If you are converting your access environment from agentless or a Pulse Secure Client environment, typically you can use your existing realms.
- Associate the roles with appropriate realms to define your access control hierarchy using role mapping.
- Define Pulse Secure client component sets, connection sets, and connections.
- Deploy Pulse Secure client to endpoints.







Related Documentation

- [Introducing Pulse Secure Client](#)
- [Client Connection Set Options for Pulse Policy Secure](#)
- [Creating a Client Connection Set for Pulse Connect Secure](#)
- [Creating a Client Component Set for Pulse Connect Secure](#)
- [Configuring a Role for Pulse Connect Secure](#)

Pulse Secure Client Status Icons

The Pulse Secure client interface (Windows and OS X) displays a system tray icon (Windows) or a menu bar icon (OS X) that indicates connection status, provides access to menu items that let the user connect and disconnect from networks and meetings, and enables quick access to the program interface. Only one icon is visible even when there are multiple connections. One icon provides the status for all connections by indicating the most important connection state information.

Table 6: Pulse Icon States (Windows Tray and OS X Menu Bar)

	Connected.
	Connecting.
	Connected with limitations
	Connection attempt failed.
	Connection suspended.
	Connected to the local network but no Internet access available. Public WiFi locations often deploy a captive portal that requires the user to enter authentication information or to accept terms of service before network access is granted. Pulse Secure client detects the presence of captive portals and does not initiate a connection to a Pulse Secure client server until Internet access is granted.

Related Documentation

- [Installation Requirements](#)
- [Introducing Pulse Secure Client](#)

Installation Requirements

For detailed information about supported platforms and installation requirements, see the *Pulse Secure Supported Platforms Guide*, which is available at <http://www.pulsesecure.net/support>.

Related Documentation

- [Introducing Pulse Secure Client](#)
- [Pulse Secure Client for Mobile Devices Overview](#)

Pulse Secure Client Error Messages Overview

Pulse Secure client error and warning messages reside in message catalog files on the endpoint. Each message includes a short description that states the problem and a long description that provides more details and suggests actions the user can take to resolve the issue.

You can edit Pulse messages by using the optional Pulse branding tool, BrandPackager. See [“Editing Pulse Secure Client Messages”](#) for more information.

All message catalog files are localized. The filename indicates the language. For example, MessageCatalogConnMgr_EN.txt is the English-language version of the file. The following filename conventions indicate the language:

- DE—German
- EN—English
- ES—Spanish
- FR—French
- IT—Italian
- JA—Japanese
- KO—Korean
- PL—Polish
- ZH—Chinese (Traditional)
- ZH-CN—Chinese (Simplified)

Related Documentation

- [Introducing Pulse Secure Client](#)
- [Customizing Pulse Secure Client Overview](#)

Accessing Pulse Secure Client Error Messages on macOS Endpoints

Pulse Secure client error and warning messages reside in message catalog files on the OS X endpoint. Each message includes a short description that states the problem and a long description that provides more details and suggests actions to resolve the issue.

You can edit Pulse messages by using the optional Pulse branding tool, BrandPackager. See [“Editing Pulse Secure Client Messages”](#) for more information.

All message catalog files are localized. The filename indicates the language. For example, MessageCatalogPulseUI_EN.txt is the English-language version of the file. The following filename conventions indicate the language:

- DE—German
- EN—English
- ES—Spanish
- FR—French
- IT—Italian
- JA—Japanese

- KO—Korean
- PL—Polish
- ZH—Chinese (Traditional)
- ZH-CN—Chinese (Simplified)

To view Pulse catalog files on macOS endpoint, use Finder to display the package contents of the Pulse application.

Related Documentation

- [Introducing Pulse Secure Client](#)
- [Pulse Secure Client Log Files](#)
- [Customizing Pulse Secure Client Overview](#)

Pulse Secure Client Log Files

The Pulse Secure client writes information to Pulse log files on Windows and Apple OS X endpoints. If you need to investigate a problem with Pulse connectivity on a Pulse client endpoint, you can instruct the user to save the client logs and e-mail them to you.

The user saves logging information by opening Pulse and then clicking File > Logs > Save As. All relevant log files are added to a single file, **LogsAndDiagnostics.zip**. The user saves the .zip file and then makes it available to you.

Pulse maintains its own log files on all supported platforms. On Windows clients, the Pulse client also logs its major operational events into Windows Event Log. Network administrators can review the Pulse event log to help troubleshoot problems. Table 7 lists the Pulse messages that can appear in the Windows event log.

To view the Pulse messages:

1. Open the Windows Event Viewer.
2. Click **Applications and Services > Pulse Secure > Operational**.

Table 7: Pulse Secure Client Event Log Messages

ID	Level	Message	Description
600	error	The connection <ID> failed authentication: Error <ID>.	802.1X EAP authentication failure.
601	informational	User has canceled authentication of the connection <ID>.	The user canceled 802.1X EAP authentication.
602	error	Failure writing wireless LAN profile for connection <ID> Error <ID>: Reason <ID>: Profile: <ID>.	A failure occurred while a wireless LAN profile was being created or modified.
603	error	Failure writing wireless LAN profile for connection <ID> Error <ID>.	A failure occurred while a wireless LAN profile was being deleted.
604	error	Failure writing wired LAN profile for connection <ID> Error <ID>: Profile: <ID>.	A failure occurred while a wired LAN profile was being created or modified.
605	error	Failure writing wired LAN profile for connection <ID> Error <ID>.	A failure while a wired LAN profile was being deleted.
500	informational	Pulse servicing has completed successfully. All components are up to date.	Pulse servicing was successful.
501	informational	Pulse servicing has completed successfully. All components are up to date.	Servicing was requested but all components were up to date.
502	error	Pulse servicing has failed. Failure summary:	Pulse servicing failed.
100	informational	User requested connection <ID> to start.	The user initiated a connect request.
101	informational	User requested connection <ID> to stop.	The user initiated a disconnect request.

ID	Level	Message	Description
102	informational	Connection <ID> is starting because its policy requirements have been met. Connection Policy: <ID>.	A connection was started because of a policy evaluation.
103	informational	Connection <ID> is stopping because of its policy requirements. Connection Policy: <ID>.	A connection was stopped because of a policy evaluation.
104	informational	Connection <ID> is stopping because of transition to context <ID>.	The machine-to-user connection was disconnected to transition to another identity.
105	informational	Connection <ID> is starting because of transition to context <ID>.	The machine-to-user connection was connected as part of the transition to another identity.
106	informational	Connection <ID> is disconnected due to computer suspend.	The connection to Pulse Connect Secure was disconnected because the computer is being suspended.
107	informational	Connection <ID> is disconnected due to login error.	A credential provider connection was disconnected because of a login error.
108	informational	Connection <ID> is disconnected because it was modified.	A connection was disconnected because it was modified.
109	informational	User requested connection <ID> to suspend.	The user initiated a suspend request.
110	informational	User requested connection <ID> to resume.	The user initiated a resume request.
1	informational	The Pulse Secure service version <ID> has successfully started.	The Pulse service started.
2	informational	The Pulse Secure service has stopped.	The Pulse service stopped.
200	error	No connections matching URL <ID> were found in Pulse database. Request to start a connection from the browser has failed.	Pulse failed to resume a connection from the browser.
400	error	The host check for connection <ID> has failed. Failed policies: <ID>.	Host Checker failed one or more policies.
300	informational	The connection <ID> was established successfully through web-proxy <ID>.	Pulse established a connection to Pulse Connect Secure or Pulse Policy Secure through a Web proxy.
301	informational	The connection <ID> was established successfully to address <ID>.	Pulse established a direct (nonproxy) connection to Pulse Connect Secure or Pulse Policy Secure.
302	informational	The connection <ID> was disconnected.	The Pulse connection was disconnected from the Pulse server.
303	error	The connection <ID> encountered an error: <ID> Peer address: <ID>.	A connection encountered an error.
304	error	The connection <ID> was disconnected due to change in routing table. Interface address changed from <ID> to <ID>.	Pulse detected a change in the route to the Pulse server.
305	informational	VPN tunnel transport for connection <ID> switched from ESP to SSL mode due to missing ESP heartbeat.	ESP to SSL fallback occurred because of missing ESP heartbeats.
306	informational	VPN tunnel for connection <ID> is switched to ESP mode.	Tunnel transport switched to ESP mode.
307	error	The connection <ID> encountered an error: System error: <ID> Peer address: <ID>.	The Pulse connection failed because of a system error.
308	error	The server disconnected connection <ID> Reason <ID>: Peer address: <ID>.	The server disconnected a connection.

Related Documentation

- [Deleting the Pulse Secure Client Log Files](#)
- [Uploading the Pulse Secure Client Log Files](#)

Deleting the Pulse Secure Client Log Files



Note: Pulse Secure, LLC recommends that you do not delete Pulse client log files.

The Pulse client controls log file size automatically. When a current log file reaches 10MB, a new one is created and the oldest log file is deleted. If you need to delete Pulse client log files, do not delete the file without first moving it to the Recycle Bin or renaming it.

To safely delete Pulse client log files on a Windows endpoint:

1. Use a command line or Windows Explorer to locate and delete debuglog.log and, optionally, debuglog.log.old. When prompted if you want to move the file to the Recycle Bin, answer Yes. Do not press Shift+Delete, which permanently deletes a file without moving it to the Recycle bin.

The file location varies depending on which version of Windows the endpoint is running. For example, the following path is valid for a Windows 7 Enterprise 64-bit endpoint: **C:\ProgramData\Pulse Secure\Logging**.

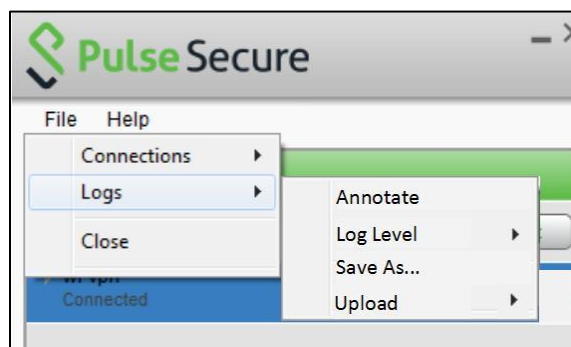
2. Empty the Recycle Bin.

Alternatively, you could first rename debuglog.log and then delete it. After you delete the log file, the Pulse client creates a new one. However, that operation might take some time depending on the activities of the Pulse client.

Uploading the Pulse Secure Client Log Files

The Pulse Secure desktop client for Windows makes it easy to transmit diagnostic log bundles to PCS gateways for analysis by system administrators. To send a log bundle to the PCS, when a VPN connection is active, run the following from the desktop client user interface: File -> Logs -> Upload.

Figure 19: Uploading Pulse Secure Client Log Files



The user must select the server to send the logs to. A dialog will appear that shows the progress of the upload.

Note that a system administrator must enable this feature on the server side before an end user can upload log files to the Pulse Secure gateway. To do this, the system administrator must launch the Pulse Secure administrative console and go to:

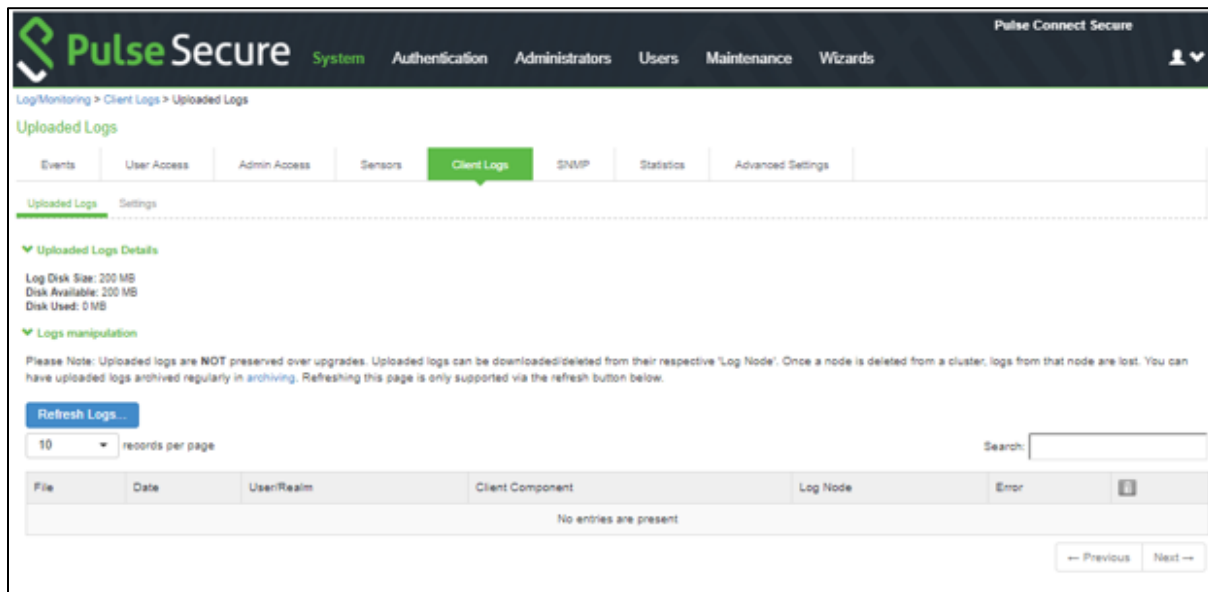
Users > Roles > General > Session Options > Enable Upload Logs

The admin must check the "Enable Upload Logs" checkbox, as shown below:

Figure 20: Enable Upload Logs

System > Log/Monitoring > Client-Side Log > Uploaded Logs

Figure 21: Viewing Upload logs



Related Documentation

- [Pulse Secure Client Error Messages Overview](#)

Migrating from Odyssey Access Client to Pulse Secure Client

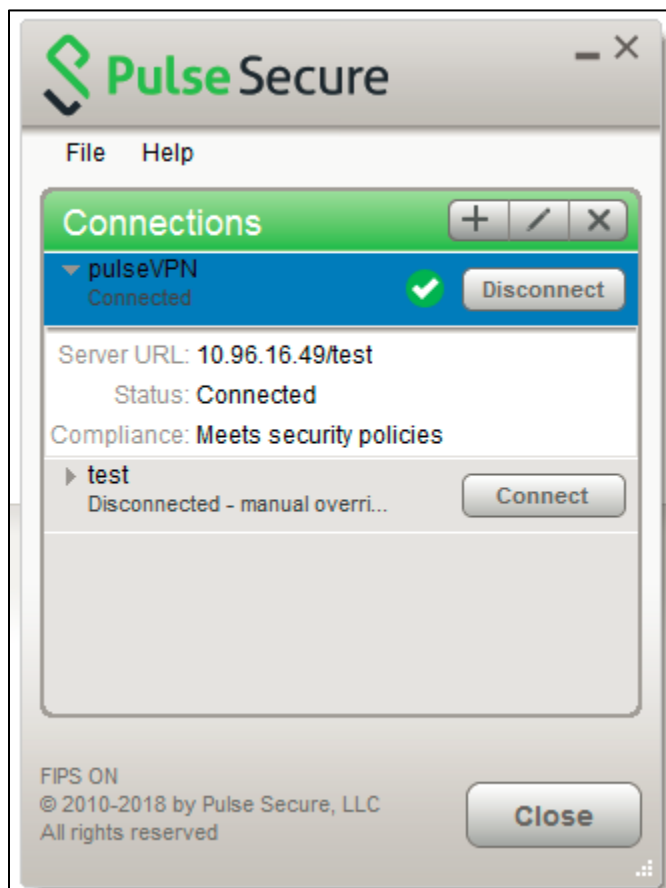
Odyssey Access Client® (OAC) is 802.1X network access client software that supports the Extensible Authentication Protocol (EAP) for secure wireless LAN access. Together with an 802.1X-compatible authentication server, OAC secures WLAN communications. OAC also serves as a client for enterprises that are deploying identity-based (wired 802.1X) networking. OAC provides wireless access to enterprise networks, home Wi-Fi networks, and public hotspots.

Pulse Secure client an extensible multiservice network client that supports integrated connectivity and secure location-aware network access. Pulse Secure client simplifies the user experience by letting the network administrator configure, deploy, and control the Pulse client software and the Pulse connection configurations that reside on the endpoint. Pulse can provide 802.1X authentication and Layer 3 access services.

Like OAC, Pulse client software is bundled with Pulse Policy Secure software. However, there are significant differences between OAC and Pulse and you should be aware of these differences when you plan a migration from OAC to Pulse. The following list includes planning considerations and best-practices for a migration project. See the related topics list for details about the Pulse configuration tasks.

- The 802.1X communication protocol that you use with OAC might need to be changed to support Pulse. OAC supports the full range of 802.1X protocols; Pulse supports only EAP-TTLS/EAP-JUAC. See [“Comparing Odyssey Access Client and Pulse Secure Client”](#), which lists the 802.1X protocols supported by OAC and Pulse.
- One common migration practice is to create new sign-in policies, user realms, and user roles for Pulse Secure client, and then control the cut-over to Pulse by enabling Pulse sign-in policies and disabling OAC sign-in policies. The new policies, realms, and roles can be clones of the existing OAC policies, realms, and roles as a starting point. However, Pulse has more robust connection decision capabilities so you will probably want to edit your Pulse roles to take advantage of the Pulse capabilities. For example, you can replace OAC with Pulse and use one client for authenticated LAN access and secure SSL VPN access. Location awareness rules allow Pulse to detect the network environment and choose a network connection based on current location.
- How many OAC configuration do you use? You need a pulse configuration for each of the OAC configurations you currently use. A Pulse access configuration is called a connection. It comprises properties that define how, when, and where a connection is established with a Pulse gateway. When you create the Pulse connections that you distribute to Pulse clients, you configure how the connection can be established. Pulse connections support machine authentication and credential provider authentication. [Figure 22](#) shows a Pulse Windows client that includes multiple connections.

Figure 22: Pulse Secure Client Interface (Windows Version)



- Odyssey Access Client is a wireless supplicant. Pulse, by design, is not a wireless supplicant. Pulse uses the underlying wireless supplicant on the endpoint, which is typically provided by the endpoint's OS X or Windows operating system. When you migrate to Pulse and uninstall OAC, you remove the OAC wireless supplicant and the endpoint falls back to using wireless connectivity provided by the OS. You define 802.1X authentication connections for the Pulse client to enable authenticated 802.1X connectivity in the enterprise network. Any custom network configurations that users added to their local OAC configuration are lost when OAC is removed. For example, if a user added connection information to connect to a home wireless network, the user will need to redefine that connection in the endpoint's wireless supplicant. A best practice is to mention this needed configuration to users as part of the Pulse roll-out. In OAC, network auto-scan lists are defined on the client. With Pulse, you can define an auto-scan list as part of an 802.1X connection that is pushed to the Pulse client.
- Do you use wireless suppression in your OAC environment? Wireless suppression disables wireless connections as long as the client has a wired network connection. You enable wireless suppression as part of a Pulse connection set. Pulse connection set properties define the decision process that Pulse uses to establish network connections.
- If you are using OAC FIPS Edition, you need to deploy Pulse 5.0 or later to support the same level of FIPS compliance that is supported by OAC.
- Do you allow users to modify configuration settings after you deploy them in your OAC environment? When you create a Pulse connection, you can define whether users can override the connection decision that has been defined by the Pulse administrator as part of the Pulse connection. You can also disable the user's ability to create new connections. Connections created by users are manual connections, that is, the connection is not tried unless the user opens Pulse and selects it.
- Do you allow OAC users to add, remove, or modify trusted servers and certificates? Pulse does not expose this functionality to users. Pulse handles certificates in the same fashion as a browser. When you define a Pulse connection you can allow users to choose to accept an unverified certificate, which allows users to connect to servers that use a self-signed certificate.

Related Documentation

- [Comparing Odyssey Access Client and Pulse Secure Client](#)
- [Client Connection Set Options for Pulse Policy Secure](#)
- [Machine Authentication for Pulse Policy Secure Overview](#)
- [Configuring Location Awareness Rules for Pulse Secure Client](#)
- [Machine and User Authentication Through a Pulse Connection for Pulse Policy Secure](#)
- [Remote Desktop Protocol Compatibility with Pulse Secure 802.1X Machine Authentication Connection](#)
- [Pulse Secure Client Installation Overview](#)

Migrating from Network Connect to Pulse Secure Client

Pulse Secure client and Network Connect (NC) can run at the same time on an endpoint.



Note: The Pulse installation program checks for NC. If the installation program finds NC Release 6.3 or later, the Pulse installation proceeds. If NC is not at least Release 6.3, the program displays a message telling the user to upgrade NC. For detailed information about supported platforms and installation requirements, see the Pulse Secure Supported Platforms Guide, which is available at <http://www.pulsesecure.net/support>.

On endpoints that connect to Pulse Connect Secure, if Pulse is running on the Windows main desktop, you cannot launch Pulse within Secure Virtual Workspace (SVW). SVW is not supported with Pulse.



Note: SVW is not supported by Pulse Policy Secure 5.1 and later and Pulse Connect Secure 8.1 and later. If a Pulse server has SVW policies configured, those policies are removed during the upgrade.

Related Documentation

- [Comparing Odyssey Access Client and Pulse Secure Client](#)

- [Comparing Network Connect and Pulse Secure Client](#)

Predictable Pulse Server Hostname Resolution with IPv6

When connecting to a Pulse Secure client server, the Pulse client uses the services of the endpoint operating system to resolve the hostname to an IP address. If a Pulse server hostname resolves to both IPv4 and IPv6 addresses, an IPv4 or an IPv6 address is presented to Pulse as the preferred IP address. The behavior depends on the operating system and how it is configured. For example, Windows 7 adheres to IETF standards that define how to establish the default address selection for IPv6. macOS 10.6 does not support that standard. Additionally, Windows 7 default settings can be changed by netsh commands, so RFC compliance can be modified on the endpoint. For these and other reasons, it is difficult to predict which Pulse server IP address would get resolved to on a given client machine.

For predictable hostname resolution, we recommend that you use different Pulse server hostnames for IPv6 and IPv4 addresses. For example, configure `myserver1.mycompany.com` for IPv4 addresses and `myserver1-v6.mycompany.com` for IPv6 addresses. The Pulse server administrator can publish `myserver1-v6.mycompany.com` to the Pulse users who are expected to connect over IPv6, and others will continue using `myserver1.mycompany.com`.

CHAPTER 2 Configuring Pulse Policy Secure

- [Before You Begin](#)
- [Pulse Policy Secure Overview](#)
- [Pulse Policy Secure and Pulse Connect Secure Deployment Options](#)
- [SRX Series Gateway Deployment Options](#)
- [Configuring a Role for Pulse Policy Secure](#)
- [Client Connection Set Options for Pulse Policy Secure](#)
- [Creating a Client Connection Set for Pulse Policy Secure](#)
- [Pulse Secure client FIPS Mode Overview for Pulse Policy Secure](#)
- [Securing the Connection State on the Pulse Secure Client](#)
- [Machine Authentication for Pulse Policy Secure Overview](#)
- [Configuring Machine-Only Machine Authentication for a Pulse Secure Client Connection](#)
- [Configuring User-After-Desktop Machine Authentication for a Pulse Secure Client Connection](#)
- [Preferred Realm and Role for Pulse Secure Client Machine Authentication](#)
- [Remote Desktop Protocol Compatibility with Pulse Secure 802.1X Machine Authentication Connection](#)
- [Credential Provider Authentication for Pulse Policy Secure Overview](#)
- [Configuring User-at-Credprov Credential Provider Authentication for a Pulse Secure Client Connection](#)
- [Configuring Machine-Then-User-at-Credprov Credential Provider Authentication for a Pulse Secure Client Connection](#)
- [Configuring a Pulse Credential Provider Connection for Password or Smart Card Login](#)
- [Machine and User Authentication Through a Pulse Connection for Pulse Policy Secure](#)
- [Configuring Location Awareness Rules for Pulse Secure Client](#)
- [Pulse Policy Secure Component Set Options](#)
- [Creating a Client Component Set for Pulse Policy Secure](#)
- [Endpoint Security Monitoring and Management for Pulse Policy Secure](#)
- [Issuing a Remediation Message with Pulse Policy Secure](#)
- [Using SMS/SCCM Remediation with Pulse Policy Secure](#)
- [Patch Management Info Monitoring and Patch Deployment](#)
- [Pushing Pulse Configurations Between Pulse Servers of the Same Type](#)
- [Enabling or Disabling Automatic Upgrades of the Pulse Secure Client](#)
- [Upgrading Pulse Secure Client](#)
- [Using Device Certificates](#)

Before You Begin

Before you begin configuring Pulse Secure client, be sure you have already configured your device network settings. Also be sure that you have defined the authentication settings, including the authentication servers and sign-in settings. Authentication Host Checker settings can directly affect a Pulse Secure Desktop client installation because you can define the conditions that an endpoint must meet to be allowed access to protected resources.

Related Documentation

- [Introducing Pulse Secure Client](#)
- [Specifying Host Checker Access Restrictions](#)

Pulse Policy Secure Overview

To enable Pulse clients to connect to Pulse Policy Secure, you configure the service so that when users request authentication, they are assigned a role based on the role mappings and optional security profile that you create. Access to specific resources is permitted only for users and devices that provide the proper credentials for the realm, that are associated with the appropriate roles, and whose endpoints meet security restrictions. If a user attempts to connect to the network from an endpoint that does not comply with the security restrictions you have defined, the user cannot access the realm or role.

As you plan your Pulse configuration, be sure you know how you want to deploy Pulse. You can use one or more of the following Pulse Secure client deployment options:

- Use the defaults or make changes to the Pulse Secure default component set and default connection set, and then download and distribute Pulse by having users log in to the Pulse server's user Web portal. After the installation is complete, users have all the connections they need to access network resources.
- Create the connections that an endpoint needs for connectivity and services, download the settings file (.pulsepreconfig), and download default Pulse installation program. For Windows endpoints you run the Pulse installation program by using an msixexec command with the settings file as an option. For OS X endpoints, you run the default installer and then import the pulsepreconfig file using a separate command.
- Distribute Pulse Secure with no preconfiguration. You can download the default PulseSecure Desktop client installation file, and then distribute the file to endpoints using your organization's standard software distribution methods. Because the installer does not contain preconfigured connections, users must define network connections manually. Users can also automatically download a Pulse server's dynamic connection by browsing to and logging into the Pulse Server's Web portal. Dynamic connections are created as manual rather than automatic connections, which means that they are run only when the user initiates the connection or the user browses to a Pulse Server and launches Pulse from the server's Web interface.

Related Documentation

- [Installing the Pulse Secure Client on Windows Endpoints Using a Preconfiguration File](#)
- [Installing the Pulse Secure Client on OS X Endpoints Using a Preconfiguration File](#)

Pulse Policy Secure and Pulse Connect Secure Deployment Options

For Pulse Policy Secure and Pulse Connect Secure, you can deploy all of the connections required for Windows and macOS clients to connect to any Pulse server.



Note: Pulse clients for mobile devices are distributed through the app stores.

Pulse Policy Secure and Pulse Connect Secure support the following deployment options:

- Web install—Create all of the settings that a Windows or macOS endpoint needs for connectivity and services, and install the software on endpoints that connect to the Pulse server's Web portal. Pulse servers include a default client connection set and client component set. The default settings enable you to deploy Pulse Secure client to users without creating new connection sets or component sets. The default settings for the client permit dynamic connections and install only the components required for the connection.
- Default installer—A default Pulse Secure installer package (in .msi format for Windows and .dmg for macOS) is included in the Pulse server software. You can distribute this default installer to endpoints, run it, and then let users create their own connections or have users browse to the Pulse server and authenticate through the server's Web portal to receive the initial configuration and bind the client to the server for future configuration updates. Users can automatically install connections to other Pulse servers (if the Pulse client's configuration allows dynamic connections)

by browsing to the user Web portal of a Pulse server where a dynamic connection has been made available. A dynamic connection is a predefined set of connection parameters that enables a client to connect to the host server. If the user is able to log in to the Pulse server's user Web portal and start Pulse from the Web interface, the connection parameters are downloaded and installed on the Pulse Secure client. Dynamic connections are created as manual rather than automatic connections, which means that they are run only when the user initiates the connection or the user browses to a Pulse Server and launches Pulse from the server's Web interface.

- **Preconfigured installer**—Create the connections that an endpoint needs for connectivity and services, download the settings file (.pulsepreconfig), and download default Pulse installation program. For Windows endpoints you run the Pulse installation program by using an msixexec command with the settings file as an option. For OS X endpoints, you run the default installer and then import the .pulsepreconfig file using a separate command.



Note: Pulse Secure client for mobile devices uses a different deployment model than Pulse for Windows and Pulse for Mac endpoints.

Related Documentation

- [Pulse Secure Client for Mobile Devices Overview](#)
- [Comparing Odyssey Access Client and Pulse Secure Client](#)
- [Comparing Network Connect and Pulse Secure Client](#)

SRX Series Gateway Deployment Options

Windows and macOS endpoints can use Pulse Secure client software to connect to SRX Series gateways that are running a Junos OS release between v10.2 and v12.3, and that have dynamic VPN access enabled and configured. For SRX Series devices running Junos OS Release 10.2 through 10.4, Pulse Secure client is supported but must be deployed separately. You can download the Pulse Secure installer from a Pulse server or the Pulse Secure Licensing and Download Center @ my.pulsesecure.net and install it using local distribution methods such as SMS/SCCM. By using preconfiguration file, you can add preconfigured connections when you install Pulse. After installing Pulse for Windows or Pulse for OS X, users can also create a connection to an SRX gateway. In Junos OS Release 11.1 and later, if the Pulse client does not exist on the client machine, the Pulse client is automatically downloaded and installed when you log into an SRX Series device.



Note: Pulse Secure Dynamic VPN functionality is compatible with SRX-Branch (SRX100-SRX650) devices only. SRX-HE (SRX1400-SRX5800 – also called SRX Data Center) devices do not support Pulse Secure Dynamic VPN from either Windows or Mac clients.

Related Documentation

- [Understanding Session Migration](#)
- [Installing Pulse Secure Client from the Web](#)
- [Installing the Pulse Secure Client on Windows Endpoints Using a Preconfiguration File](#)
- [Installing the Pulse Secure Client on OS X Endpoints Using a Preconfiguration File](#)

Configuring a Role for Pulse Policy Secure

A user role defines session settings and options, personalization settings (user interface customization and bookmarks), and enabled access features (Web, file, application, Telnet/SSH, Terminal Services, network, meeting, and e-mail access). A user role does not specify resource access control or other resource-based options for an individual request. For example, a user role can define whether or not a user can perform Web browsing. However, the individual Web resources that a user may access are defined by the Web resource policies that you configure separately.

To configure a role for Pulse endpoints:

1. From the admin console, select **Users > User Roles > New User Role**.
2. Enter a name for the role and, optionally, a description.

3. **Click Save Changes.** The role configuration tabs appear.
4. Set the following options:

General > Overview

- **Options**—Select the **Pulse Secure** check box.

General > Restrictions

- **Source IP**—Source IP options allow you to make an assignment to this role dependent on the endpoint's IP address or IP address range. To enable source IP address restrictions, select **Allow or deny users from the following IP addresses**, and then add IP addresses or address ranges. Select **Allow** to allow users to sign in from the specified IP address, or **Deny** to prevent users from signing in from the specified IP address. Then click **Add**. When you are finished making changes, click **Save Changes**.
- If you add multiple IP addresses, move the highest priority restrictions to the top of the list by selecting the check box next to the IP address, and then clicking the up arrow button. For example, to deny access to all users on a wireless network (10.64.4.100) and allow access to all other network users (0.0.0.0), move the wireless network address (10.64.4.100) to the top of the list and move the (0.0.0.0) network below the wireless network.
- **Browser**—Browser options allow you to enforce the use of a particular type of browser for Web access to Pulse Policy Secure. Browser options apply only to operations that involve accessing Pulse Policy Secure through its user Web portal, such as acquiring a dynamic connection or installing Pulse through a role. Normal connection operations between the Pulse Secure client and Pulse server are not affected by browser restrictions.
- **Certificate**—Certificate options allow you to require users to sign in from an endpoint that possesses the specified client-side certificate from the proper certificate authority. Before you enable this option, be sure that you have configured the client-side certificate on the Trusted Client CAs page of the admin console.
- **Host Checker**—Host Checker options allow you to enable Host Checker policies, to choose one or more policies for the role, and specify whether the endpoint must meet all or just one of the selected Host Checker policies. The Host Checker policies that appear as Available Policies must be previously defined as part of the Endpoint Security settings in the Authentication section of the admin console.

General > Session Options

- **Session lifetime**—Session lifetime options allow you to set timeout values for user sessions. You can change the defaults for the following:
 - **Max. Session Length**—Specify the number of minutes a user session might remain open before ending. During a user session, prior to the expiration of the maximum session length, Pulse prompts the user to re-enter authentication credentials, which avoids the problem of terminating the user session without warning.
 - **Heartbeat Interval**—Specify the frequency at which the Pulse client should notify Pulse Connect Secure to keep the session alive. You should ensure that the heartbeat interval of the agent is greater than the Host Checker interval, otherwise performance could be affected. In general, the heartbeat interval should be set to at least 50% more than the Host Checker interval.
 - **Heartbeat Timeout**—Specify the amount of time that Pulse Connect Secure should wait before terminating a session when the endpoint does not send a heartbeat response.
 - **Auth Table Timeout**—Specify a timeout value for the auth table entry to be provisioned as needed. Based on user identity and endpoint status, Pulse Policy Secure assigns the user a set of roles that specify which resources the user can access. The Pulse server pushes the roles associated with each endpoint's source IP address (called auth table entries) to the Infranet Enforcer. The Infranet Enforcer allows traffic between the endpoint and the protected resources based on resource access policies.

- **Reminder Time**—When the Enable Session Extension feature is enabled, the Reminder Time specifies the number of minutes prior to a session end when the server sends a notice through Pulse and notifies the user that the session will end soon.
- **Use Session/Idle timeout values sent by the primary Radius authentication Server**—The session takes its timeout values from the Radius server Idle-timeout setting.
- **Enable Session Extension**—You can select the Enable Session Extension check box to allow Pulse users to continue a session beyond the maximum session length. If this feature is enabled, users can extend a session through the Pulse client user interface.
- **Allow VPN Through Firewall**—Enable this option to allow Infranet Enforcer traffic to act as a heartbeat and keep the session alive. This option is useful for iOS devices.
- **Roaming session**—Roaming allows user sessions to work across source IP addresses. Roaming session options include the following:
 - **Enabled**—Select this option to enable roaming for users mapped to this role. A roaming user session works across source IP addresses, which allows mobile users with dynamic IP addresses to sign in to Pulse Connect Secure from one location and continue working from other locations.
 - **Limit to subnet**—Select this option to limit the roaming session to the local subnet specified in the Netmask box. Users can sign in from one IP address and continue using their sessions with another IP address as long as the new IP address is within the same subnet.
 - **Disabled**—Select this option to disable roaming user sessions for users mapped to this role. Users who sign in from one IP address cannot continue an active Infranet Controller session from another IP address; user sessions are tied to the initial source IP address.

General > UI Options

The UI options allow you to define options that a user sees after a successful login to the Pulse Policy Secure server by means of a browser.

5. Select the **Agent** tab. The agent is the client program for a user assigned to this role. When a user connects to the system using a Web browser, the user can click a button to download and install the selected agent if it is not already installed on the user's endpoint. Configure the following options.
 - Select **Install Agent for this role**.

Agent options appear only after you select this check box.

 - Select **Install Pulse Secure**.
6. In the **Session scripts** area, optionally specify a location for the following:
 - **Windows: Session start script**—Specify a script (.bat, .cmd, or .exe) to run for users assigned to the role after Pulse connects with Pulse Policy Secure. For example, you can specify a script that maps network drives on an endpoint to shares on protected resources. The script must be in a location (either local or on the network) that is accessible by the user.
 - **Windows: Session end script**—Specify a script (.bat, .cmd, or .exe) to run for users assigned to the role after Pulse Secure client disconnects from Pulse Policy Secure. For example, you can specify a script that disconnects mapped network drives. If there is no start script defined, or the start script has not been run, the end script does not run. The script must be in a location (either local or on the network) that is accessible by the user.
7. Click **Save Changes**, and then select **Agent > Pulse Secure Settings**.
8. Select a component set that you have created, use the Default component set or select **none**. You would select **none** only if you are creating this role to distribute new or updated connections to existing Pulse users.
9. Click **Save Changes**.
10. Select **Users > User Realms > Select Realm > Role Mapping > New Rule** to configure role mapping rules that map Pulse Secure client users to the role you configured.

Related Documentation

- [Pulse Policy Secure Overview](#)
- [Endpoint Security Monitoring and Management for Pulse Policy Secure](#)

- *Creating Global Host Checker Policies*

Client Connection Set Options for Pulse Policy Secure

A Pulse Secure client connection set contains network options and allows you to configure specific connection policies for client access to any Pulse server that supports Pulse Secure client. The following sections describe each of the configuration options for a Pulse connection set.

Pulse Secure Connection Set Options

The following items apply to all connections in a connection set.

- **Allow saving logon information**—Controls whether the Save Settings check box is available in login dialog boxes in the Pulse client. If you clear this check box, the Pulse client always requires users to provide credentials. If you select this check box, users have the option of saving their credentials.

The Pulse Secure client can retain *learned user settings*. These settings are retained securely on the endpoint, evolving as the user connects through different Pulse servers. The Pulse Secure client can save the following settings:

- Certificate acceptance
- Certificate selection
- Realm
- Username and password
- Proxy username and password
- Secondary username and password
- Role



Note: If the authentication server is an ACE server or a RADIUS server and authentication is set to Users authenticate using tokens or one-time passwords, Pulse ignores the Allow saving logon information option. If the user sees a username and token prompt and the Save settings check box is disabled. Pulse supports soft token, hard token, and smart card authentication.

When a user opts to save settings, that information is used for each subsequent connection without prompting. If a setting changes (for example, if a user changes a password), the saved setting is invalid and connection attempts fail. In this case, the user must use the client's Forget Saved Settings feature, which clears all user-saved settings.

- **Allow user connections**—Controls whether connections can be added by the user.
- **Display splash screen**—Clear this check box to hide the Pulse splash screen that normally appears when the Pulse client starts.
- **Dynamic certificate trust**—Determines whether users can opt to trust unknown certificates. If you select this check box, a user can ignore warnings about invalid certificates and connect to the target Pulse server.



Note: By default, **Dynamic certificate trust** check box will be unchecked.

- **Dynamic connections**—Allows connections within this connection set to be automatically updated or added to a Pulse Secure client when the user connects to the Pulse server through the user Web portal, and then clicks the Pulse button. Dynamic connections are created as manual rather than automatic connections, which means that they are run only when the user initiates the connection or the user browses to a Pulse server and launches Pulse from the server's Web interface.

If dynamic connections are disabled, and the user logs in through the Web portal of a Pulse server that is not already included in the Pulse client's connection set, then starting Pulse from the Web portal does not add a new Pulse connection for that Pulse server. If you choose to disable dynamic connections, you can still allow users to manually create connections by enabling **Allow User Connections**.

- **FIPS mode enabled**—Enable FIPS mode communications for all Pulse connections in the connection set. The Federal Information Processing Standard (FIPS) defines secure communications for the U.S. government. When a Pulse connection is operating in FIPS mode, **FIPS On** appears in the lower corner of the Pulse client interface. If the Pulse server hardware does not support FIPS mode operations, FIPS mode configuration options are not present in the admin console interface. FIPS mode operations are supported on PSA-V Series Pulse Secure Gateways and some SA series appliances. The device must be running Pulse Policy Secure R5.0 or later or Pulse Connect Secure R8.0 or later.



Note: Users cannot enable FIPS mode from within the Pulse client. You must create FIPS-enabled connections on the server and deploy them.

- **Wireless suppression**—Disables wireless access when a wired connection is available. If the wired connection is removed, Pulse enables the wireless connections with the following properties:
 - Connect even if the network is not broadcasting.
 - Authenticate as computer when computer information is available.
 - Connect when this network is in range.



Note: Wireless suppression occurs only when the wired connection is connected and authorized. If you enable wireless suppression, be sure to also configure a connection that enables the client to connect through a wired connection.

UAC 802.1X Connection Type Options

Use this connection type to define authenticated connectivity to 802.1X devices, wired or wireless. Users cannot create 802.1X connections from the Pulse client interface. Users see 802.1X connections in the Pulse interface only when the connection has been deployed from the server and the specified network is available.



Note: When configuring an 802.1x connection, Pulse Policy Secure will force Pulse client to connect using Client IP Address of the same family as of the specified Radius Client's IP Family. When you specify Radius Client as IPV6, Pulse Policy Secure will allow Pulse Client to connect only using IPv6 address in 802.1x scenario. Also Pulse Client does not consider link-local IPv6 Addresses to complete an 802.1x Connection in a scenario where it needs an IPv6 address to connect to Pulse Policy Secure. For more information see PPS Admin Guide.

- **Adapter type**—Specifies the type of adapter to use for authentication: wired or wireless.
- **Outer username**—Enables a user to appear to log in anonymously while passing the actual login name (called the inner identity) through an encrypted tunnel. As a result, the user's credentials are secure from eavesdropping, and the user's inner identity is protected. In general, enter **anonymous**, which is the default value. In some cases, you might need to add additional text. For example, if the outer identity is used to route the user's authentication to the proper server, you might be required to use a format such as [anonymous@acme.com](#).



Note: If you leave the box blank, the client passes the users or the machine's Windows login name as the outer identity.

- **Scan list**—If you selected wireless as the adapter type, the scan list box is available to specify the SSIDs, including non-broadcast SSIDs, to connect to in priority order. If you leave the list empty, the user can connect to any available wireless network.

- **Support Non-broadcast SSID**—Allows a user to connect to a non-broadcast wireless network from within the Pulse interface. Selecting this field enables the following options:
 - **Wireless Security Algorithm**—Specify wireless authentication:
 - WPA
 - WPA2
 - **Wireless Security Cipher**—Specify the type of encryption used by the non-broadcast network:
 - TKIP
 - AES

If the non-broadcast SSID options are configured, the Pulse connection configuration includes the values and they are used to configure the wireless profile on the endpoint.

Trusted Server List (for UAC 802.1X Connection)

FQDN criteria for 802.1X/EAP server certificates (with wildcard support) can be specified in the Trusted Server List of the PPS admin console. In the name field, you can enter a fully-qualified-domain name (FQDN) that can be either an exact FQDN or an FQDN that begins with a "." and/or can contain wildcards ("*").

Note the Following:

- The "ANY" entry matches any server certificate name.
- An entry that contains "=" requires an exact Subject:DN (Distinguished Name) match.
- An entry that is neither "ANY" nor contains "=" is an FQDN. It can be either an exact value or include wildcards and/or begin with a "." character. This value will be checked against FQDNs in the server's certificate (Subject:DN:CN=..., SAN:DNS=...).
 - An entry that begins with "." will wildcard only the first subdomain (domain component) in the FQDN. For example, ".mycompany.com" will match "foo.mycompany.com" but not "foo.bar.mycompany.com". As such, a FQDN beginning with "." is equivalent to the same FQDN beginning with "." (e.g., ".mycompany.com" is equivalent to ".mycompany.com"). Note that this mechanism is more restrictive than what is described in RFC 5280.
 - FQDN may contain at most one wildcard per domain component (DC). For example, "a.mycompany.com" is not allowed and will always result in authentication failure.
 - A wildcard matches 1 or more characters (but not zero characters). For example, "f*r.mycompany.com" will match "foo-bar.mycompany.com" but not "fr.mycompany.com".
 - See RFC 2818 and RFC 6125 for more details and security implications of wildcards.
 - Be careful when mixing wildcard FQDN entries with certificates that contain wildcards in their names. For example, the entry "foo*.mycompany.com" will match a certificate with the name "foo*bar.mycompany.com".
 - This wildcarding mechanism does not work with server certificates that contain the "?" character in their names. (This is not a common occurrence.)
- You can choose any server certificate's issuing certificate authority (CA) from the drop-down list. It could be the direct issuer or any CA at higher level in the certificate chain, up to the root.

Connect Secure or Policy Secure (L3) Connection Type Options

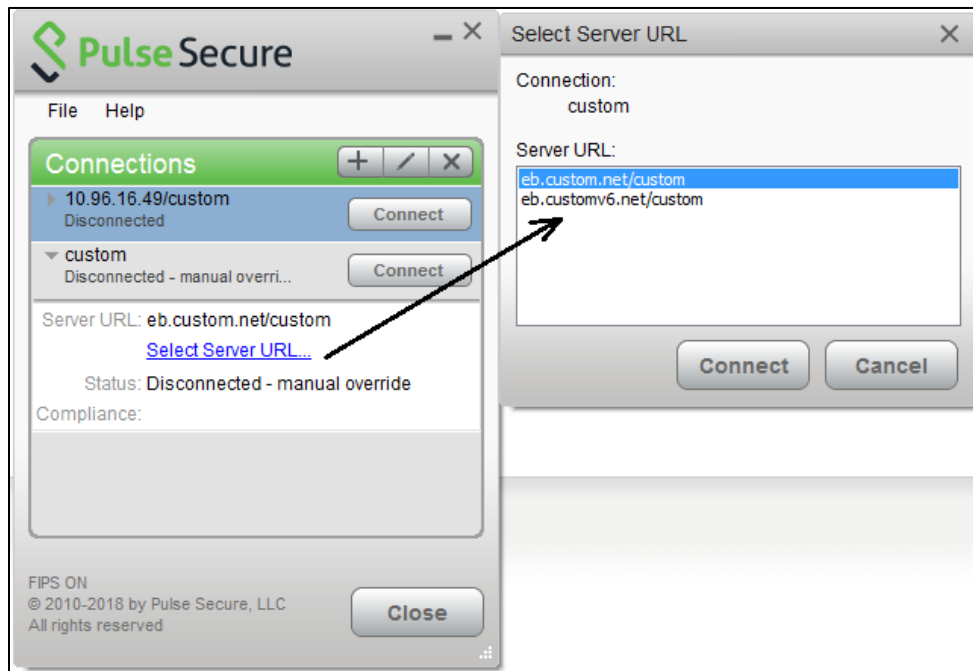
Use a Connect Secure or Policy Secure (L3) connection for a Layer 3 connection to Pulse Connect Secure or Pulse Policy Secure.

- **Allow user to override connection policy**—Allows a user to override the connection policy by manually connecting or disconnecting. Typically, you leave this option selected to make sure that a user can establish a connection under all conditions. If you disable this check box, the user cannot change the endpoint's connection status, suspend/resume a connection to Pulse Connect Secure or shut down Pulse.
- **Support Remote Access (SSL VPN) or LAN Access (UAC) on this connection**—This option must be selected if this connection is for Pulse Policy Secure. If the connection is for Pulse Connect Secure, you can disable this check box and use the connection for accessing Pulse Collaboration meetings only by also selecting Support Remote Access (SSL VPN) or LAN Access (UAC) on this connection.

- **Enable Pulse Collaboration integration on this connection**—This option must be disabled if this connection is for Pulse Policy Secure. If the connection is for Pulse Connect Secure, you can enable this check box and use the connection for accessing Pulse Collaboration meetings.
- **Connect to URL of this server only**—Specifies whether the endpoint connects to this Pulse server exclusively or if it can connect to any of the servers listed in the list of connection URLs. Disable this check box to enable the List of Connection URLs.
- **List of Connection URLs**—Allows you to specify a list of Pulse servers (Pulse Policy Secure or Pulse Connect Secure) for this connection. The Pulse client attempts to reach each server in the list, in the order listed, until it succeeds. You can specify up to 8 Pulse servers. If you enable the Randomize URL list order check box, Pulse ignores the listed order and chooses from the list randomly. If the Pulse connection is configured to use a list of Pulse servers, any preferred roles and realms you specify must be applicable to all of those servers. The default behavior is to start with the most recently connected URL first, then try from top of list. The most recently connected URL is saved across reboots. Connections that use machine authentication always use the ordered list of connection URLs. In the case of an interrupted connection, such as temporarily losing the WiFi link, Pulse always tries to reconnect to the most recently connected URL. During a credential provider connection attempt, Pulse chooses the URL automatically. It does not display a window to let the user choose a URL.

Figure 23 shows how the Pulse user can select a server from the list of connection URLs.

Figure 23: Pulse for Windows client with a List of Connection URLs



- **Attempt most recently connected URL first**—If you have specified a list of connection URLs, you can select this check box to have the Pulse client always attempt the most recent successful connection. If that connection is not successful, Pulse then starts at the top of the list.
- **Randomize URL list order**—If you have specified a list of connection URLs, select this check box to have the Pulse client ignore the order in which the servers are listed. You can select this option to spread the connection load across multiple Pulse servers. If you enabled Attempt most recently connected URL first, then Pulse attempts that connection first. If that connection attempt fails, Pulse chooses randomly from the list for the next connection attempt. During a credential provider connection attempt, Pulse chooses the URL automatically. It does not display a window to let the user choose a URL.



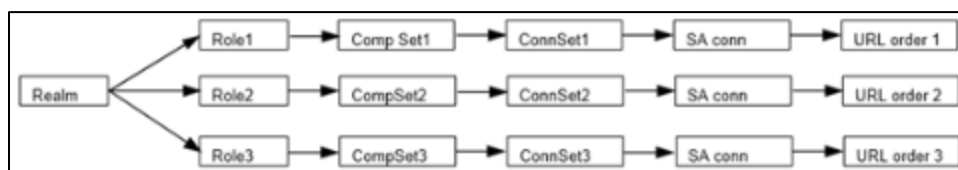
Note: IF-MAP federation must be configured to ensure that a suspended session can be resumed to a different URL.



Note: When this feature is used with Pulse Policy Secure, all of the Pulse servers in the list must be configured for failover, so that any one of them can provision the firewall enforcer.

The connection list enables you to support different URL ordering for different users. You can use custom expressions in a realm's role mapping rules to associate different users to different roles. For example, you could use a custom expression that is based on the OU (Organization Unit) when using an LDAP authentication server, (UserDN.OU = "Americas"). Each role is associated with a different Pulse connection set, and each Pulse connection within the connection set is configured with different URL lists. [Figure 24](#) shows the role mapping for providing different URL lists for different users.

Figure 24: Mapping URL Lists to Users



SRX (for Dynamic VPN) Connection Type Options

Use an SRX connection for a dynamic VPN connection to an SRX Series Services Gateway.

- **Address**—Specifies the IP address of the SRX Series device.
- **Allow user to override connection policy**—Allows users to override the connection policy by manually connecting or disconnecting. Typically, you leave this option selected to make sure that a user can establish a connection under all conditions. If you disable this check box, the user cannot change the endpoint's connection status or shut down Pulse Secure client.

Pulse Connection is Established Options

For all connection types, specify how the connection is established. The options vary according to the type of connection. Automatic connections include machine authentication and credential provider connections. Connections can be established using the following options.



Note: All connections that are configured to start automatically attempt to connect to their target networks at startup time. To avoid multiple connection attempts, be sure that only one connection is configured to start automatically, or configure location awareness rules.

- Modes:
 - **User**—Enables user authentication.
 - **Machine**—Enables machine authentication, which requires that Active Directory is used as the authentication server and that machine credentials are configured in Active Directory. A machine connection is, by default, an automatic connection.
 - **Machine or user**—Enables machine authentication for the initial connection. After user authentication, the machine authentication is dropped. When the user logs out, the machine authentication connection is restored.
- Options:

- **Connect automatically**—Connections are attempted when the conditions specified in the location awareness rules are true, and disconnected when the conditions are no longer true.
- **Enable pre-desktop login (Credential provider)**—Enables Pulse client interaction with the credential provider software on the endpoint. The user credentials are used to establish the authenticated Pulse connection to the network, login to the endpoint, and login to the domain server.
- **Reconnect at Session Timeout or Deletion**—If this option is enabled, user initiated sessions automatically attempt to reconnect upon a session timeout or deletion. If this option is disabled, then user initiated sessions remain disconnected upon a session timeout or deletion.

Pulse Connection is Established Examples

The following configurations show how to select the Connection is established options of a Pulse connection set for specific user login behavior:

Figure 25: Connect manually

♥ Connection is established:

Specify mode:

Options:

- ☐ Connect automatically
- ☒ Reconnect at Session Timeout or Deletion If checked, user-initiated sessions will automatically attempt to reconnect upon a session timeout or deletion.
- ☐ Enable pre-desktop login (Credential provider)

Figure 26: Connect automatically after user signs in to the desktop

♥ Connection is established:

Specify mode:

Options:

- ☒ Connect automatically
- ☐ Reconnect at Session Timeout or Deletion If checked, user-initiated sessions will automatically attempt to reconnect upon a session timeout or deletion.
- ☐ Enable pre-desktop login (Credential provider)

Figure 27: Connect automatically when the machine starts; machine credentials are used for authentication

♥ Connection is established:

Specify mode:

Options:

- ☒ Connect automatically
- ☒ Reconnect at Session Timeout or Deletion If checked, user-initiated sessions will automatically attempt to reconnect upon a session timeout or deletion.
- ☐ Enable pre-desktop login (Credential provider)



Note: When you use machine credentials for authentication and no user credentials, Pulse cannot perform user-based tasks. The following tasks can be run only when the user is logged in:

- Run session scripts
- Detect or modify proxy settings
- Run automatic Pulse client upgrade
- Install or upgrade Pulse components

Figure 28: Connect automatically when the machine starts; the connection is authenticated again when the user signs in to the desktop


♥ Connection is established:

Specify mode: Machine or User

Options:

- ☒ Connect automatically
- ☐ Reconnect at Session Timeout or Deletion If checked, user-initiated sessions will automatically attempt to reconnect upon a session timeout or deletion.
- ☐ Enable pre-desktop login (Credential provider)

The configuration in Figure 28 enables machine authentication for the initial connection. After the user connects with user credentials, the machine authentication is dropped. When the user logs out, the machine authentication connection is restored. In one typical use case, the machine credentials provide access to one VLAN, and the user credentials provide access to a different VLAN.



Note: If the machine and user have different roles, each role should map to the same connection set. Otherwise, after the user connects, the existing connection set might be replaced.

Figure 29: Connect automatically at user login

♥ Connection is established:

Specify mode: User

Options:

- ☒ Connect automatically
- ☒ Reconnect at Session Timeout or Deletion If checked, user-initiated sessions will automatically attempt to reconnect upon a session timeout or deletion.
- ☒ Enable pre-desktop login (Credential provider)

The configuration in Figure 29 enables Pulse client interaction with the credential provider software on the endpoint. The user credentials are used to establish the authenticated Pulse connection to the network, to log in to the endpoint, and to log in to the domain server.

Figure 30: Connect automatically when the machine starts; connection is authenticated again at user login

♥ Connection is established:

Specify mode: Machine or User

Options:

- ☒ Connect automatically
- ☒ Reconnect at Session Timeout or Deletion If checked, user-initiated sessions will automatically attempt to reconnect upon a session timeout or deletion.
- ☒ Enable pre-desktop login (Credential provider)

The configuration in Figure 30 enables Pulse client interaction with the credential provider software on the endpoint. Machine credentials are used to establish the authenticated Pulse connection to the network. When the user provides user credentials, the connection is authenticated again. In one typical use case, the machine credentials provide access to one VLAN, and the user credentials provide access to a different VLAN.

Location Awareness Rules

For Connect Secure or Policy Secure (L3) and SRX connections that are set to have the connection established automatically, you should define location awareness rules that enable an endpoint to connect conditionally. If you do not have location awareness rules defined, Pulse attempts to connect with each connection that is defined as an automatic connection until it connects successfully. Location awareness rules allow you to define an intelligent connection scheme. For

example, the endpoint connects to Pulse Policy Secure if it is connected to the company intranet, or it connects to Pulse Connect Secure if it is in a remote location.

A Pulse connection uses the IP address of a specified interface on the endpoint to determine its network location. Each location awareness rule includes the following settings:

- **Name**—A descriptive name, for example, “corporate-DNS.” A name can include letters, numbers, hyphens, and underscores.
- **Action**—The method the connection uses to discover the IP address. Choose one of the following values:
 - **DNS Server**—Allows the endpoint to connect if the endpoint’s DNS server on the specified interface is set to one of the specified values. Use the Condition box to specify IP addresses or address ranges.
 - **Resolve Address**—Allows the endpoint to connect if the hostname specified in the DNS Name box can be resolved by the DNS server for the specified interface. If one or more address ranges are specified in the Address Range box, the address must resolve to one of the ranges to satisfy the expression.
 - **Endpoint Address**—Allows the endpoint to connect if the IP address of the specified interface is within a range specified in the IP Address Range box.



Note: To create a negative location awareness rule, you first create the positive state and then use rule requirement logic to use the rule as a negative condition.

Machine Connection Preferences

The Machine Connection Preferences appear if you have selected one of the machine authentication options for how the connection is established. Normally Pulse presents a selection dialog box if more than one realm or role is available to the user. However, a connection that is established through machine authentication fails if a dialog box is presented during the connection process. To suppress the selection dialogs, either ensure that only one role and realm is available to users, or specify a preferred realm and role for this connection.

- **Preferred Machine Realm**—Specify the realm that this connection uses when establishing the machine connection. The connection ignores any other realm available for the specified login credentials
- **Preferred Machine Role Set**—Specify the role or the name of rule for the role set that this connection uses when establishing the machine connection. The role or rule name used must be a member of the preferred machine realm.

User Connection Preferences

The User Connection Preferences options enable you to specify a realm and role for automatic connections that would otherwise present a selection dialog box to the user. To suppress the selection dialogs, either ensure that only one role and realm is available to users, or specify a preferred realm and role for this connection.



Note: If the Pulse connection is configured to use a list of Pulse servers, the preferred roles and realms you specify must be applicable to all of those servers.

- **Preferred User Realm**—Specify the realm for this connection that is used when a user logs onto the endpoint. The connection ignores any other realm available for the user’s login credentials

If one of the credential provider connection options is enabled, the following options are available:

- **Preferred Smartcard Logon Realm**—Preferred realm to be used when user logs in with a smart card.
- **Preferred Password Logon Realm**—Preferred realm to be used when user logs in with a password.



Note: Be sure that the authentication realms you specify exist, and that they support the appropriate login credential option.

- **Preferred User Role Set**—Specify the preferred role or the name of the rule for the role set to be used for user authentication. The role or rule name used must be a member of the preferred user realm.
- **Select client certificate from machine certificate store**—Enables you to specify the location of the client certificate on a Windows endpoint as part of a Pulse connection that verifies the identity of both the machine and the user before establishing a connection. When this check box is selected, the Pulse connection looks at client certificates located in the Local Computer personal certificate store. When this check box is not selected, the connection accesses the user certificate store on a Windows endpoint. For more information, see [“Machine and User Authentication Through a Pulse Connection for Pulse Policy Secure”](#).

Related Documentation

- [Machine Authentication for Pulse Policy Secure Overview](#)
- [Configuring Location Awareness Rules for Pulse Secure Client](#)
- [Machine and User Authentication Through a Pulse Connection for Pulse Policy Secure](#)
- [Remote Desktop Protocol Compatibility with Pulse Secure 802.1X Machine Authentication Connection](#)
- [Creating a Client Connection Set for Pulse Policy Secure](#)

Creating a Client Connection Set for Pulse Policy Secure

A Pulse Secure client connection (also called a client configuration) set contains network options and allows you to configure specific connection policies for client access to any Pulse server that supports Pulse Secure client.

To create a Pulse Secure client configuration:

1. From the admin console, select **Users > Pulse Secure > Connections**.
2. Click **New**.
3. Enter a name and, optionally, a description for this connection set.



Note: You must enter a connection set name before you can create connections.

4. Click **Save Changes**.
5. From the main Pulse Secure Connections page, select the connection set.
6. Under Options, select or clear the following check boxes:
 - **Allow saving logon information**—Controls whether the Save Settings check box is available in login credential dialog boxes in the Pulse Secure client. If you clear this check box, the Pulse Secure client always requires users to provide credentials. If you select this check box, users have the option of saving their credentials.
 - **Allow user connections**—Controls whether connections can be added by the user through the Pulse client interface.
 - **Display splash screen**—Clear this check box to hide the Pulse splash screen that normally appears when the Pulse client starts.
 - **Dynamic certificate trust**—Determines whether users can opt to trust unknown certificates. If you select this check box, a user can ignore warnings about invalid certificates and connect to the target Pulse server.



Note: By default, **Dynamic certificate trust** check box will be unchecked.

- **Dynamic connections**—Allows new connections to be added automatically to a Pulse Secure client when the user connects through the Pulse server's Web portal and then starts Pulse through the Web portal interface.
- **FIPS mode enabled**—Enable FIPS mode communications for all Pulse connections in the connection set. The Federal Information Processing Standard (FIPS) defines secure communications for the U.S. government. When a Pulse connection is operating in FIPS mode, **FIPS On** appears in the lower corner of the Pulse client interface. If the Pulse server hardware does not support FIPS mode operations, FIPS mode configuration options are not present in the admin console interface. FIPS mode operations are supported on PSA-V Series Pulse Secure

Gateways and some SA series appliances. The device must be running Pulse Policy Secure R5.0 or later or Pulse Connect Secure R8.0 or later.



Note: Users cannot enable FIPS mode for connections that are created on the client. You must deploy connections with FIPS mode enabled using a pre-configured connection set with FIPS mode enabled or have users establish a browser session to the FIPS enabled Pulse server.

- **Wireless suppression**—Disables wireless access when a wired connection is available. Wireless suppression occurs only when the wired connection is connected and authorized.
7. Under Connections, click **New** to define a new connection.
 8. Enter a name and, optionally, a description for this connection.
 9. Select a type for the connection. Type can be any of the following:
 - **UAC 802.1X**
 - **Connect Secure or Policy Secure (L3)**
 - **SRX**
 10. If you select **UAC 802.1X** from the type list, enter a value or select or clear the following check boxes:
 - **Adapter type**—Select Wired or Wireless.
 - **Outer username**—Enter the outer username.
 - **Scan list**—If you selected wireless as the adapter type, the scan list box is available to specify the SSIDs, including non-broadcast SSIDs, to connect to in priority order. If you leave the list empty, the user can connect to any available wireless network.
 - **Support Non-broadcast SSID**—Allow users to connect to a non-broadcast wireless network from within the Pulse interface.

Wireless Security Algorithm—Specify the type of wireless security that the client uses to connect to the non-broadcast wireless network:

 - WPA
 - WPA2

Wireless Security Cipher—Specify the type of encryption that the client uses to communicate with the non-broadcast network:

 - TKIP
 - AES
 11. Click **Save Changes**.
 12. If you selected **Connect Secure or Policy Secure (L3)** for the type, configure the following:
 - **Allow user to override connection policy**



Note: If you disable this check box, the user cannot change the endpoint's connection status or shut down Pulse Secure client.

- **Enable Pulse Collaboration integration on this connection**
- **Support Remote Access (SSL VPN) or LAN Access (UAC) on this connection**
- **Connect to URL of this server only**—Specifies whether the endpoint connects to this Pulse server exclusively or if it can connect to any of the servers listed in the list of connection URLs. Disable this check box to enable the List of Connection URLs.
- **List of Connection URLs**—Allows you to specify a list of Pulse servers (Pulse Policy Secure or Pulse Connect Secure) for this connection. The Pulse client attempts to reach each server in the list, in the order listed, until it succeeds. You can specify up to 8 Pulse servers. If you enable the Randomize URL list order check box, Pulse ignores the listed order and chooses from the list randomly. If the Pulse connection is configured to use a list of Pulse servers, any preferred roles and realms you specify must be applicable to all of those servers.

- Start with most recently connected URL first, then try from top of list. The most recently connected URL is saved across reboots.
- Connections that use machine authentication ignore this option and always use the ordered list of connection URLs.
- In the case of an interrupted connection, such as temporarily losing the WiFi link, Pulse always tries to reconnect to the most recently connected URL.
- During a credential provider connection attempt, Pulse chooses the URL automatically. It does not display a window to let the user choose a URL.
- **Randomize URL list order**—If you have specified a list of connection URLs, select this check box to have the Pulse client ignore the order in which the servers are listed. You can select this option to spread the connection load across multiple Pulse servers. If you enabled Attempt most recently connected URL first, then Pulse attempts that connection first. If that connection attempt fails, Pulse chooses randomly from the list for the next connection attempt. During a credential provider connection attempt, Pulse chooses the URL automatically. It does not display a window to let the user choose a URL.



Note: IF-MAP federation must be configured to ensure that a suspended session can be resumed to a different URL.



Note: When this feature is used with Pulse Policy Secure, all of the Pulse servers in the list must be configured for failover, so that any one of them can provision the firewall enforcer.

- **Client Certificate Location: Client**—Enables you to specify the certificate store that the Pulse client accesses for certificate authentication on Windows endpoints. Typically, you would use the default setting, which retrieves the certificate from the user's personal certificate store, and then certificate authentication is controlled by the **Connection is established** option. If you disable this option, the Pulse connection uses a machine certificate from the Local Computer Personal certificate store, which enables you to perform machine authentication and user authentication for the Pulse connection. If you disable this option, you must also create a sign-in policy and configure authentications servers to perform the user authentication.
13. If you select **SRX**, enter the IP address of the SRX device in the Address box and specify whether you want the user to be able to override connection policy.
 14. Specify how the connection is established, manually or automatically. These options enable you to configure machine authentication and credential provider authentication.
 15. (Optional) You can enable location awareness on automatic connections by creating location awareness rules. Location awareness can force a connection to a particular interface.
 16. (Optional) You can set preferred role and realm options for a machine authentication connection.



Note: If the Pulse connection is configured to use a list of Pulse servers, the preferred roles and realms you specify must be applicable to all of those servers.

17. After you have created the client connection set, create a client component set and select this connection set.

Related Documentation

- [Configuring Location Awareness Rules for Pulse Secure Client](#)
- [Preferred Realm and Role for Pulse Secure Client Machine Authentication](#)
- [Remote Desktop Protocol Compatibility with Pulse Secure 802.1X Machine Authentication Connection](#)

Pulse Secure FIPS Mode Overview for Pulse Policy Secure

The United States Federal Information Processing Standard (FIPS) defines security and interoperability requirements for computer systems that are used by the U.S. government. Pulse Secure client for Windows, Mac, iOS (32-bit iOS devices only), and Android support FIPS mode operations when communicating with Pulse Connect Secure and Pulse Secure client for Windows and Mac support FIPS mode operations when communicating with Pulse Policy Secure. When it is operating in FIPS mode, **FIPS On** appears in the bottom corner of the Pulse for Windows and Mac clients.

You enable FIPS mode operations for the Pulse Windows client when you configure Pulse connections on the server. You enable FIPS mode operations for a connection set. That connection set can include any or all of the four types of Pulse connections:

- **UAC (802.1X)**—Pulse client uses FIPS mode cryptography for authentication but it uses default Microsoft cryptography for the WEP/WPA wireless encryption.
- **Connect Secure or Policy Secure (L3)**—FIPS mode cryptography is supported.
- **SRX**—FIPS mode cryptography is not supported.



Note: Users cannot enable FIPS mode for connections that are created on the client. You must deploy connections with FIPS mode enabled using a pre-configured connection set with FIPS mode enabled or have users establish a browser session to the FIPS enabled Pulse server.

Windows Endpoint Requirements

Pulse supports FIPS mode on Windows 7 and later Windows versions, and on Pulse for iOS and Android for communications with Pulse Connect Secure. FIPS is not supported by the Pulse client for Apple OS X.

To support client certificate private key operations on Windows, the security policy on the Windows endpoint must have FIPS enabled. To verify that FIPS is enabled, use the Microsoft Management Console (MMC). Make sure that the Group Policy Snap-in is installed, and then open the following item:

Local Computer Policy > Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options

Scroll through the Policy list and make sure that the following policy is enabled:

System cryptography: Use FIPS compliant algorithms for encryption, hashing and signing

Configuration Overview

The Pulse client includes all components required for FIPS mode communications. You enable the Pulse server for FIPS mode operations as part of the System SSL Options (System > Configuration > Security > SSL Options). To enable FIPS mode communications for Pulse for Windows clients, deploy one or more Pulse connections to the client that are FIPS enabled. [Figure 31](#) shows the check box in the Pulse connection set configuration screen that enables FIPS mode operations for all connections in the connection set.

Figure 31: Enabling FIPS Mode for Pulse Connections

Pulse Secure Client > Connections > New Connection Set

New Connection Set

Name:

Description:

Owner:
 Last Modified: 2018-05-09 05:48:23 UTC
 Server ID: 0312MVD4A0EM704VS

▼ Always-on vpn wizard

[Configure Always-on VPN using wizard](#)

▼ Options

Name	Value
Always-on Pulse Client Prevents end users from circumventing Pulse connections. This option will disable all configuration settings that allow the end user to disable or remove Pulse connections, services or software.	<input type="checkbox"/>
VPN only access When Pulse client connects to a PCB having lock down mode enabled, it will enter lock-down mode and won't let any traffic flow through unless a Locked-down VPN connection is in connected state. User is allowed to connect or disconnect any connection. User is allowed to add any new connection/server URI. User is allowed to delete a connection if the connection is not locked down.	<input type="checkbox"/>
Allow saving logon information Enables the Save settings checkbox in the certificate trust and password prompts.	<input checked="" type="checkbox"/>
Allow user connections Allows user to create connections via the Pulse UI.	<input checked="" type="checkbox"/>
Display Splash Screen Controls whether the splash screen is displayed when Pulse starts.	<input checked="" type="checkbox"/>
Dynamic certificate trust Controls whether users may accept to trust unknown certificates.	<input checked="" type="checkbox"/>
Dynamic connections Allows connections to be deployed automatically from devices.	<input checked="" type="checkbox"/>
EAP Fragment Size Maximum number of bytes in an EAPoL message from the client for 802.1x connections. Range: 450 - 3000 bytes	<input type="text" value="1400"/>
Enable captive portal detection Pulse will attempt to detect the presence of a captive portal hotspot. Only applies to Connect Secure and Policy Secure (L3) connections.	<input type="checkbox"/>
Enable embedded browser for captive portal Pulse will use an embedded web browser for captive portal pages. Only applies when captive portal detection is enabled.	<input checked="" type="checkbox"/>
Enable embedded browser for authentication Pulse will use embedded browser for saml, custom sign-in or token based authentication.	<input type="checkbox"/>
FIPS mode enabled Deploy client with Federal Information Processing Standard enabled.	<input checked="" type="checkbox"/>
Wireless suppression Disconnect all wireless interfaces when a wired interface gets connected to a network. Applies to all wireless connections (not just those managed by Pulse).	<input type="checkbox"/>
Prevent caching smart card PIN Enabling this will ensure the smart card PIN value is not cached by the client process.	<input type="checkbox"/>



Note: If the Pulse server hardware does not support FIPS mode operations, FIPS mode configuration options are not present in the admin console interface. FIPS mode operations are supported on PSA-V Series Pulse Secure Gateways and some SA series appliances. The device must be running Pulse Policy Secure R5.0 or later or Pulse Connect Secure R8.0 or later.

Related Documentation

- [Understanding Pulse Secure FIPS Level 1 Support](#)
- [Enabling FIPS Level 1 Support](#)
- [Creating a Client Connection Set for Pulse Policy Secure](#)
- [FIPS Supported Platforms](#)

Securing the Connection State on the Pulse Secure Client

To disable user interaction with Pulse connections on the endpoint, you can configure Pulse Secure Connections so that when they are deployed to the endpoint, users cannot shut down a connection, suspend or resume a connection, or shut down Pulse. Disabling user interaction with Pulse enables the Pulse administrator to control how endpoints connect to the network without allowing the user to override administrative control. For example, if you use machine authentication, the connection from endpoint to server is established automatically. By locking down the Pulse endpoint, users cannot change their connection.

To secure the Pulse endpoint:

1. Click **Users > Pulse Secure Connections**.
2. Edit or create a new connection.
3. Disable the check box labeled **Allow user to override connection policy**.

Related Documentation

- [Client Connection Set Options for Pulse Policy Secure](#)
- [Endpoint Security Monitoring and Management for Pulse Policy Secure](#)
- [Machine Authentication for Pulse Policy Secure Overview](#)

Machine Authentication for Pulse Policy Secure Overview

Machine authentication uses machine credentials (machine name and password or machine certificate) to authenticate the endpoint. You can enable machine authentication for Pulse Policy Secure as part of a Pulse Secure Connection and distribute the connection to endpoints through the normal Pulse distribution methods. You enable Pulse machine authentication support on a Pulse connection, either Layer 2 or Layer 3.

The following describes the requirements for a machine authentication environment:

- The authentication server used by the Pulse connection must be Active Directory/Windows NT for machine name/password authentication or a certificate server for machine certificate authentication.
- The endpoint must be a member of a Windows domain, and the machine credentials must be defined in Active Directory.
- The Pulse connection must be configured so that no prompts are presented during the login process. For example, prompts for realm or role selection or for a server certificate trust prompt cause the connection to fail. You can specify a preferred role and realm for the connection, which eliminates realm and role selection dialogs.



Note: If the Pulse connection is configured to use a list of Pulse servers, the preferred roles and realms you specify must be applicable to all of those servers.

- For machine certificate authentication, the domain workstation login certificate must be issued by the domain certificate authority. The root certificate must be in the Machine Trusted Certificate store instead of the certificate store for a particular user.

Pulse supports the following machine authentication types:

- **machine-only**—The connection is established using machine credentials when no user is logged in. The connection is maintained after user login.
- **user-after-desktop**—The connection is established using machine credentials when no user is logged in. After user login, the machine connection is disconnected. Once the user logs out, the user connection is disconnected and the machine connection is reestablished.

Related Documentation

- [Preferred Realm and Role for Pulse Secure Client Machine Authentication](#)
- [Configuring Machine-Only Machine Authentication for a Pulse Secure Connection](#)

- [Configuring User-After-Desktop Machine Authentication for a Pulse Secure Connection](#)
- [Machine and User Authentication Through a Pulse Connection for Pulse Policy Secure](#)
- [Remote Desktop Protocol Compatibility with Pulse Secure 802.1X Machine Authentication Connection](#)

Configuring Machine-Only Machine Authentication for a Pulse Secure Connection

When a Pulse connection is configured for machine-only machine authentication, the Pulse connection is established using machine credentials when no user is logged in. The connection is maintained after user login.

To enable a Pulse connection for machine-only machine authentication:

1. Click **Users > Pulse Secure > Connections** and create or select a connection set.
2. Create or edit a connection. For the connection type, you can select either UAC (802.1X) for a Layer 2 connection or Connect Secure or Policy Secure (L3) for a Layer 3 connection.
3. Under Connection is established, for the mode select **Machine**.

Machine credentials are used to connect to the Pulse server when the endpoint is started, before a user logs in. The connection is maintained when a user logs in, logs out, or switches to a different login.



Note: When you use machine credentials for authentication and no user credentials, Pulse cannot perform user-based tasks. The following tasks can be run only when the user is logged in:

- Run session scripts
- Detect or modify proxy settings
- Run automatic Pulse client upgrade
- Install or upgrade Pulse components

4. Select the **Connect automatically** check box.

Figure 32: Connect automatically when the machine starts; machine credentials are used for authentication

♥ Connection is established:

Specify mode: Machine

Options:

☒ Connect automatically

☒ Reconnect at Session Timeout or Deletion If checked, user-initiated sessions will automatically attempt to reconnect upon a session timeout or deletion.

☐ Enable pre-desktop login (Credential provider)

5. For a Layer 2 connection that uses machine certificate authentication, make sure that the connection has an entry in the Trusted Server List. To allow any server certificate, type **ANY** as the Server certificate DN. To allow only one server certificate, specify the server certificate's full DN, for example, C=US; ST=NH; L=Kingston; O=My Company; OU=Engineering; CN=c4k1.stnh.mycompany.net; E=ausername@mycompany.com.
6. Specify **Realm and Role Preferences** to suppress realm or role selection dialogs during the login process:
 - **Preferred Machine Realm**—Specify the realm for this connection. The connection ignores any other realm that is available for the specific login credentials.
 - **Preferred Machine Role Set**—Specify the preferred role or the name of the rule for the role set to be used for user authentication. The role or rule name must be a member of the preferred machine realm.



Note: If the Pulse connection is configured to use a list of Pulse servers, the preferred roles and realms you specify must be applicable to all of those servers.

Related Documentation

- [Machine Authentication for Pulse Policy Secure Overview](#)
- [Credential Provider Authentication for Pulse Policy Secure Overview](#)
- [Remote Desktop Protocol Compatibility with Pulse Secure 802.1X Machine Authentication Connection](#)

Configuring User-After-Desktop Machine Authentication for a Pulse Secure Connection

When a Pulse connection is configured for user-after-desktop machine authentication, the connection is established using machine credentials when no user is logged in. After user login, the machine connection is disconnected. Once the user logs out, user connection is disconnected and machine connection is reestablished.

To enable a Pulse connection for user-after-desktop machine authentication:

1. Click **Users > Pulse Secure > Connections**, and then create or select a connection set.
2. Create or edit a connection. For the connection type, you can select either UAC (802.1X) for a Layer 2 connection or Connect Secure or Policy Secure (L3) for a Layer 3 connection.
3. Under Connection is established, for mode, select **Machine or User**.

Machine credentials are used to connect to the Pulse server when the endpoint is started, before a user logs in. When a user logs in, the machine authentication connection is dropped, and the user login is used instead. When the user logs out, the machine connection is reestablished.

4. Select the **Connect automatically** check box.

Figure 33: Connect automatically when the machine starts; the connection is authenticated again when the user signs in to the desktop

▼ Connection is established:

Specify mode: Machine or User

Options:

- ☒ Connect automatically
- ☐ Reconnect at Session Timeout or Deletion If checked, user-initiated sessions will automatically attempt to reconnect upon a session timeout or deletion.
- ☐ Enable pre-desktop login (Credential provider)

5. For a Layer 2 connection that uses machine certificate authentication, make sure that the connection has an entry in the Trusted Server List. To allow any server certificate, type **ANY** as the Server certificate DN. To allow only one server certificate, specify the server certificate's full DN, for example, C=US; ST=NH; L=Kingston; O=My Company; OU=Engineering; CN=c4k1.stnh.mycompany.net; E=ausername@mycompany.com.
6. Specify **Realm and Role Preferences** to suppress realm or role selection dialogs during the login process for both machine and user logins:
 - **Preferred Machine Realm**—Specify the realm that this connection uses when establishing the machine connection. The connection ignores any other realm that is available for the specific login credentials.
 - **Preferred Machine Role Set**—Specify the role or the name of a rule for the role set that this connection uses when establishing the machine connection. The role or rule name used must be a member of the preferred machine realm.
 - **Preferred User Realm**—Specify the realm that for this connection that is used when a user logs into the endpoint. The connection ignores any other realm that is available for the user's login credentials.
 - **Preferred User Role Set**—Specify the preferred role or the name of the rule for the role set to be used for user authentication. The role or rule name used must be a member of the preferred user realm.



Note: If the Pulse connection is configured to use a list of Pulse servers, the preferred roles and realms you specify must be applicable to all of those servers.

Related Documentation

- [Machine Authentication for Pulse Policy Secure Overview](#)
- [Credential Provider Authentication for Pulse Policy Secure Overview](#)
- [Remote Desktop Protocol Compatibility with Pulse Secure 802.1X Machine Authentication Connection](#)

Preferred Realm and Role for Pulse Secure Client Machine Authentication

When a Pulse Secure Connection is configured to use machine authentication, any prompts that occur during the login process cause the connection to fail. For example, if the Pulse server authentication policy allows a user to select a realm or a role during the login process, Pulse presents a dialog box to the user and prompts for the realm or role selection. To avoid failed connections caused by prompts during machine authentication, you can specify a preferred role and realm for a Pulse connection.



Note: Realm and role prompts are not the only prompts that are possible during the login process. If the Pulse connection has the Dynamic Certificate Trust option enabled and there is an issue with the server certificate, Pulse asks the user if it is OK to proceed. That certificate prompt causes a machine connection to fail. Note that the Pulse prompt for upgrading Pulse software is presented after the user connection is established, and it will not affect a machine authentication connection.



Note: If the Pulse connection is configured to use a list of Pulse servers, any preferred roles and realms you specify must be applicable to all of those servers.

For a Pulse connection that is used for machine authentication, you *do not* need to specify the preferred role if either of the following conditions is true:

- Users are mapped to only one role.
- Users are mapped to more than one role, but the realm's role mapping properties are set to merge settings for all assigned roles.

For a Pulse connection that is used for machine authentication, you *must* specify the preferred realm if the authentication sign-in policy allows the user to select a realm. If that realm maps to only one role, you do not need to specify the role.

For a Pulse connection that is used for machine authentication, you *must* specify the preferred role if either of the following conditions is true:

- The realm that the user connects to maps to more than one role and the realm's role mapping properties are set to require that the user must select a role. The preferred role set must be the name of a role assigned in that realm.
- The realm that the user connects to maps to more than one role, and the realm's role mapping properties are defined by role mapping rules. You specify the preferred role by specifying the name of a rule that assigns the role set. [Figure 34](#) shows a role mapping rule with the rule name highlighted.

Figure 34: Pulse Role Mapping Rule

User Realms > custom > Role Mapping

Role Mapping

General Authentication Policy **Role Mapping**

Specify how to assign roles to users when they sign in. Users that are not assigned a role will not be able to sign in.

[New Rule...](#) [Duplicate](#) [Delete](#) [↶](#) [↷](#) [Save Changes](#)

	When users meet these conditions	assign these roles	Rule Name	Stop
<input type="checkbox"/>	1. username is "test"	→ custom	custom	

When more than one role is assigned to a user:

- ☒ Merge settings for all assigned roles
- ☐ User must select from among assigned roles
- ☐ User must select the sets of merged roles assigned by each rule

Note: Users that do not meet any of the above rules will not be able to sign into this realm.

To identify the connection as a machine authentication connection, you specify how the connection is established using one of the configurations shown in [Figure 35](#) and [Figure 36](#).

Figure 35: Connect automatically when the machine starts; machine credentials are used for authentication

♥ Connection is established:

Specify mode:

Options:

- ☒ Connect automatically
- ☒ Reconnect at Session Timeout or Deletion If checked, user-initiated sessions will automatically attempt to reconnect upon a session timeout or deletion.
- ☐ Enable pre-desktop login (Credential provider)

This option uses the machine credentials defined in Active Directory for the machine login process and uses the same credentials for user login. When you select this option, the Realm and Role Set Preferences settings enable you to specify the following options:

- **Preferred Machine Realm**—Type the realm name that maps to the role you want to assign.
- **Preferred Machine Role Set**—Type the name of the role. The role must be one that is identified in the realm's role mapping properties. Alternatively, you can specify the name of a role mapping rule that assigns the role set.

Note: When you use machine credentials for authentication and no user credentials, Pulse cannot perform user based tasks. The following tasks can be run only when the user is logged in:

- Run session scripts
- Detect or modify proxy settings
- Run automatic Pulse client upgrade
- Install or upgrade Pulse components

Figure 36: Connect automatically when the machine starts; the connection is authenticated again when the user signs in to the desktop

♥ Connection is established:

Specify mode:

Options:

- ☒ Connect automatically
- ☐ Reconnect at Session Timeout or Deletion If checked, user-initiated sessions will automatically attempt to reconnect upon a session timeout or deletion.
- ☐ Enable pre-desktop login (Credential provider)

This option uses the Active Directory machine credentials for the machine login process. When machine login is complete, Pulse drops that connection and then uses the user credentials for user login. When you select this option, the Realm and Role Set Preferences enable you to specify the following options:

- **Preferred Machine Realm**—Type the realm name that maps to the role you want to assign.
- **Preferred Machine Role Set**—Type the name of the role. The role must be one that is identified in the realm's role mapping properties. Alternatively, you can specify the name of a role mapping rule that assigns the role set.
- **Preferred User Realm**—Type the realm name that maps to the role you want to assign.
- **Preferred User Role Set**—Type the name of the role. The role must be one that is identified in the realm's role mapping properties. Alternatively, you can specify the name of a role mapping rule that assigns the role set.

Related Documentation

- [Machine Authentication for Pulse Policy Secure Overview](#)

Remote Desktop Protocol Compatibility with Pulse Secure 802.1X Machine Authentication Connection

If you want to use Remote Desktop Protocol (RDP) to access an endpoint over a Pulse 802.1X connection, machine authentication is required. Because of a Microsoft OS limitation, an RDP connection attempt over a user-only 802.1X authenticated connection will fail. To support RDP connectivity over an authenticated 802.1X connection, you must have a machine-only connection or a machine-then-user connection. In the case of a machine-then-user connection, when you use RDP to connect to a machine over an 802.1X connection that is connected as user, the connection transitions the 802.1X connection to a machine connection. If you subsequently log into the machine directly, it transitions back to a user connection.

To access the endpoint using RDP, you must define the connection to be established using one of the following Pulse configurations:

Figure 37: Connect automatically when the machine starts; machine credentials are used for authentication

Figure 37 shows a configuration window titled "Connection is established:". It contains a "Specify mode:" dropdown menu set to "Machine". Below this, under "Options:", there are three checkboxes: "Connect automatically" (checked), "Reconnect at Session Timeout or Deletion" (checked), and "Enable pre-desktop login (Credential provider)" (unchecked). A note next to the second checkbox states: "If checked, user-initiated sessions will automatically attempt to reconnect upon a session timeout or deletion."

Figure 38: Connect automatically when the machine starts; the connection is authenticated again when the user signs in to the desktop

Figure 38 shows a configuration window titled "Connection is established:". It contains a "Specify mode:" dropdown menu set to "Machine or User". Below this, under "Options:", there are three checkboxes: "Connect automatically" (checked), "Reconnect at Session Timeout or Deletion" (unchecked), and "Enable pre-desktop login (Credential provider)" (unchecked). A note next to the second checkbox states: "If checked, user-initiated sessions will automatically attempt to reconnect upon a session timeout or deletion."

Related Documentation

- [Machine Authentication for Pulse Policy Secure Overview](#)

Credential Provider Authentication for Pulse Policy Secure Overview

When Microsoft introduced Windows Vista, it moved away from a login integration interface based on Graphical Identification and Authentication (GINA) in favor of credential provider authentication. The Pulse Secure credential provider

integration enables connectivity to a network that is required for the user to log into the Windows domain. For example, the domain controller might reside behind a firewall and the endpoint uses credential provider login to connect to Pulse Policy Secure prior to domain login. Pulse Secure client integrates with Microsoft credential providers to enable password-based login and smart card login. Pulse connections also support an option that allows a user to use either a smartcard or a password to log in. Credential provider login is supported on Windows 7 and later Windows platforms.

You can use the Pulse support for credential provider authentication to provide single sign-on capabilities. Pulse establishes a connection to the network and then uses the same credentials to log in to the Windows domain.

You enable Pulse credential provider support on a Pulse connection. After the connection has been downloaded to the endpoint through the normal Pulse distribution methods, Pulse annotates the credential provider tile that appears on the user login screen by adding a Pulse icon in the lower right corner of the tile. When the user initiates the login process, Pulse establishes the connection.



Note: A connection attempt to a Pulse server fails if the connection uses Host Checker and Host Checker is installed in a non-default appdata folder. Host Checker is installed

Pulse supports the following credential provider types:

- **user-at-credprov**—The connection is established before the user login using credentials collected at the selected credential tile, which provides single-sign-on functionality. The connection is maintained as an active connection on the user's desktop. To enable user-at-credprov authentication, use the Pulse connection configuration shown in [Figure 39](#).

Figure 39: Connect automatically at user login

✓ Connection is established:

Specify mode: User

Options:

- ☒ Connect automatically
- ☒ Reconnect at Session Timeout or Deletion If checked, user-initiated sessions will automatically attempt to reconnect upon a session timeout or deletion.
- ☒ Enable pre-desktop login (Credential provider)

- **machine-then-user-at-credprov**—The connection is established using machine credentials when no user is logged in. When a user clicks a login tile and provides user credentials, the machine connection is disconnected and a new connection is established. When the user logs out, the user connection is disconnected and the machine connection is reestablished. In one typical machine-then-user-at-credprov implementation, the machine connection and the user connection are mapped to different VLANs. To enable machine-then-user-at-credprov authentication, use the Pulse connection configuration shown in [Figure 40](#).

Figure 40: Connect automatically when the machine starts; connection is authenticated again at user login

✓ Connection is established:

Specify mode: Machine or User

Options:

- ☒ Connect automatically
- ☒ Reconnect at Session Timeout or Deletion If checked, user-initiated sessions will automatically attempt to reconnect upon a session timeout or deletion.
- ☒ Enable pre-desktop login (Credential provider)

Pulse credential provider support usage notes:

- If the endpoint includes more than one Pulse Layer 2 connection, Windows determines which connection to use:
 1. If a network cable is attached to the endpoint, Layer 2 wired connections are attempted, and then wireless connections. If more than one wireless network is available, the order is determined by the scan list specified as a Pulse connection option.

2. After all Layer 2 options are attempted, Pulse runs location awareness rules to find one or more eligible Layer 3 connections that are configured for credential provider login. If more than one Layer 3 connection is found, Pulse prompts the user to select a connection. A user can cancel the network connection attempt by clicking the cancel button.
 3. After Pulse evaluates all configured connection options, Pulse returns control to Windows, which enables the user login operation.
- For connections that use user credentials, you can configure the Pulse connection so that prompts are presented during the login process, for example, prompts for realm or role selection or a server certificate trust prompt. For connections that use machine credentials, Pulse prompts cause the connection to fail because there is no interface to allow a response to the prompts. You can suppress any potential realm and role choice by specifying a preferred realm and role for the connection.



Note: If the Pulse connection is configured to use a list of Pulse servers, the preferred roles and realms you specify must be applicable to all of those servers.

- Pulse upgrade notifications and actions are disabled during credential provider login and postponed until the user connection is established. Host Checker remediation notifications are displayed.
- To allow users to log in using either a smart card or a password, you can create different authentication realms for each use case and then specify the Preferred Smartcard Logon Realm and Preferred Password Logon Realm as part of the connection properties.
- A credIf the client machine has non-default value for the %appdata% environment variable, then login using Host Checker are enabled and client machine has non default value for %appdata% then login using GINA fails

appdata is a user environment variable as well. Here user modifies the appdata of user. Credential provider runs in system user context and no user is logged in that time. User appdata details are stored in HKEY_CURRENT_USER registry. Since no user is logged in current HKEY_CURRENT_USER will be of system user. So credential provider uses a logic to form the default appdata path of user. This logic will work when default path is modified.

Related Documentation

- [Configuring Location Awareness Rules for Pulse Secure Client](#)

Configuring User-at-Credprov Credential Provider Authentication for a Pulse Secure Connection

With a user-at-credprov connection, the Pulse connection establishes the connection before user login using credentials collected at the selected credential tile, which provides single sign-on functionality. The connection is maintained as an active connection on the user's desktop.

To enable user-at-credprov credential provider support for a Pulse connection:

1. Create a Pulse connection set for the role (Users > Pulse Secure > Connections), and then create a new Pulse connection. You can select either a Layer 3 connection type, Connect Secure or Policy Secure (L3), or a Layer 3 connection type, UAC (802.1X).
2. In the Connection is established section, select **User** for the mode.
3. Under Options, select the **Connect automatically** and the **Enable pre-desktop login (Credential provider)** check boxes.

Figure 41: Connect automatically at user login

♥ Connection is established:

Specify mode: User ▼

Options:

- ☒ Connect automatically
- ☐ Enable pre-desktop login (Credential provider)

- For a Layer 2 connection that uses machine certificate authentication, make sure that the connection has an entry in the Trusted Server List. To allow any server certificate, type ANY as the Server certificate DN. To allow only one server certificate, specify the server certificate's full DN, for example, C=US; ST=NH; L=Kingston; O=My Company; OU=Engineering; CN=c4k1.stnh.mycompany.net; E=username@mycompany.com.
- Specify Realm and Role Preferences to suppress realm or role selection dialogs during the login process:
 - Preferred User Realm**—Specify the realm for this connection. The connection ignores any other realm that is available for the specific login credentials.

The following options enable you to allow the user to login using a smart card or a password:

- Preferred Smartcard Logon Realm**—Preferred realm to be used when user logs in with a smart card.
- Preferred Password Logon Realm**—Preferred realm to be used when user logs in with a password.



Note: Be sure that the authentication realms you specify exist, and that they support the appropriate login credential option.

- Preferred User Role Set**—Specify the preferred role or the name of the rule for the role set to be used for user authentication. The role or rule name must be a member of the preferred user realm.



Note: If the Pulse connection is conmd to use a list of Pulse servers, the preferred roles and realms you specify must be applicable to all of those servers.

Related Documentation

- [Credential Provider Authentication for Pulse Policy Secure Overview](#)
- [Configuring a Pulse Credential Provider Connection for Password or Smart Card Login](#)

Configuring Machine-Then-User-at-Credprov Credential Provider Authentication for a Pulse Secure Connection

With a machine-then-user-at-credprov connection, Pulse establishes the connection using machine credentials when no user is logged in. When a user clicks a login tile and provides user credentials, the machine connection is disconnected, and a new connection is established. When the user logs out, the user connection is disconnected, and the machine connection is reestablished. In one typical machine-then-user-at-credprov implementation, the machine connection and the user connection are mapped to different VLANs.

To enable machine-then-user-at-credprov credential provider support for a Pulse connection:

- Create a Pulse connection set for the role (Users > Pulse Secure > Connections), and then create a new Pulse connection. You can select either a Layer 3 connection type, Connect Secure or Policy Secure (L3), or a Layer 2 connection type, Policy Secure (802.1X).

2. In the Connection is established section, select User or **Machine** for the mode.
3. Under **Options**, select the **Connect automatically** check box.

Figure 42: Connect automatically when the machine starts. Connection is authenticated again at user login

▼ **Connection is established:**

Specify mode: Machine or User ▼

Options:

☒ **Connect automatically**

☐ **Enable pre-desktop login (Credential provider)**

4. In the Connection is established section, select one of the following options:
5. For a Layer 2 connection that uses machine certificate authentication, make sure that the connection has an entry in the Trusted Server List. To allow any server certificate, type ANY as the Server certificate DN. To allow only one server certificate, specify the server certificate's full DN, for example, C=US; ST=NH; L=Kingston; O=My Company; OU=Engineering; CN=c4k1.stnh.mycompany.net; E=ausername@mycompany.com.
6. Specify **Realm and Role** Preferences to suppress realm or role selection dialogs during the login process for both machine and user logins:
 - **Preferred Machine Realm**—Specify the realm that this connection uses when establishing the machine connection. The connection ignores any other realm that is available for the specific login credentials.
 - **Preferred Machine Role Set**—Specify the role or the name of the rule for the role set that this connection uses when establishing the machine connection. The role or rule name used must be a member of the preferred machine realm.
 - **Preferred User Realm**—Specify the realm that for this connection that is used when a user logs in to the endpoint. The connection ignores any other realm that is available for the user's login credentials.

The following options enable you to allow the user to log in using a smart card or a password:

- **Preferred Smartcard Logon Realm**—Preferred realm to be used when user logs in with a smart card.
- **Preferred Password Logon Realm**—Preferred realm to be used when user logs in with a password.



Note: Be sure that the authentication realms you specify exist, and that they support the appropriate login credential option.

- **Preferred User Role Set**—Specify the preferred role or the name of rule for the role set to be used for user authentication. The role or rule name used must be a member of the preferred user realm.



Note: If the Pulse connection is configured to use a list of Pulse servers, the preferred roles and realms you specify must be applicable to all of those servers.

7. Optionally, specify pre-login preferences:
 - **Pre-login maximum delay**—The time period (in seconds) that a Windows client waits for an 802.1X connection to succeed during the login attempt. The range is 1 to 120 seconds.
 - **Pre-login user based virtual LAN**—If you are using VLANs for the machine login, you can enable this check box to allow the system to make the VLAN change.
8. Click **Save Changes**, and then distribute the Pulse connection to Pulse client endpoints.

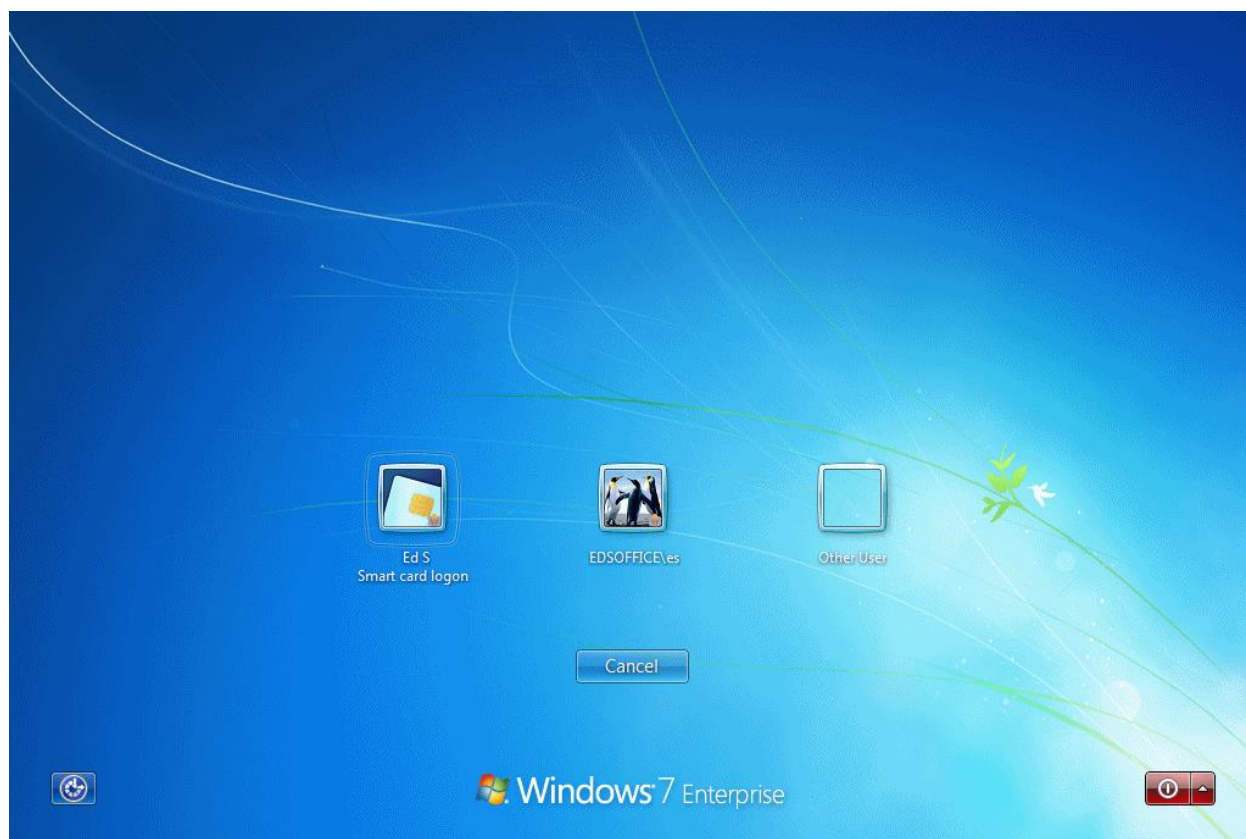
Related Documentation

- [Credential Provider Authentication for Pulse Policy Secure Overview](#)
- [Machine and User Authentication Through a Pulse Connection for Pulse Policy Secure](#)
- [Configuring a Pulse Credential Provider Connection for Password or Smart Card Login](#)

Configuring a Pulse Credential Provider Connection for Password or Smart Card Login

If you allow a user to log in with a smart card or with a username/password, then you can have the Pulse Credential provider automatically authenticate the user based on the login method. The Pulse user sees two different credential provider tiles for the Pulse connection, one for smart card authentication and one for username/password authentication. Credential provider tiles that launch a Pulse connection include a Pulse logo. See [Figure 43](#) The Pulse connection determines which realm to use through preferred realm settings that you specify as part of the Pulse connection preferences. If the connection succeeds, the login type is saved so that, if re-authentication is needed (for example, if the connection times out), the same login type is used.

Figure 43: Pulse Credential Provider Tiles



Before you begin:

- Before you deploy a connection that uses this feature, make sure that you have created all the authentication realms that are required. You need one realm for smart card authentication and a different one for user name/password authentication. Both realms can be mapped to the same role, or you can use different roles. In either case you include a remediation role for endpoints that do not pass Host Checker evaluation. If you use machine authentication for a connection (machine-then-user-at-credprov), you need an authentication realm for the machine.
- Make sure that all of the realms that are used in the Pulse connection are included in the sign-in policy.
- The authentication realms on the Pulse server must be configured so that the Preferred Pre-login Smartcard Realm uses certificate authentication and the Preferred Pre-login Password Realm uses username/password authentication.

The following procedure summarizes the steps to create a Pulse Secure Connection that uses credential provider authentication, and allows the user to choose either smart card login or username/password login. Table 8 describes the configuration options:

1. Click **Users > Pulse Secure > Connections** and create or select a connection set.
2. **Create or edit** a connection. For connection type, you can select either **UAC (802.1X)** for a **Layer 2 connection** or **Connect Secure or Policy Secure (L3)** for a Layer 3 connection. The **SRX connection** type does not support credential provider authentication.
3. For the Connection is established option, choose one of the credential configuration options shown in [Figure 44](#) and [Figure 45](#).

Figure 44: Connect automatically after user signs in to the desktop

▼ **Connection is established:**

Specify mode: User ▼

Options:

☒ Connect automatically

☐ Enable pre-desktop login (Credential provider)

The user credentials are used to establish the authenticated Pulse connection to the network, log in to the endpoint, and log in to the domain server.

Select **User** as the mode. Under options, select **Connect automatically**.

Figure 45: Connect automatically when the machine starts; the connection is authenticated again when the user signs in to the desktop

▼ **Connection is established:**

Specify mode: Machine or User ▼

Options:

☒ Connect automatically

☐ Enable pre-desktop login (Credential provider)

Machine credentials are used to establish the authenticated Pulse connection to the network using the specified Machine Connection Preferences or Pre-login Connection Preferences. When the user provides user credentials, the connection is authenticated again.

Select **Machine or User** as the mode. Under options, select **Connect automatically**.

4. For Connect Secure or Policy Secure (L3) connections that are set to have the connection established automatically, you can define location awareness rules that enable an endpoint to connect conditionally.
5. For a Layer 2 connection that uses machine certificate authentication, make sure that the connection has an entry in the Trusted Server List. To allow any server certificate, type **ANY** as the Server certificate DN. To allow only one server certificate, specify the server certificate's full DN, for example, C=US; ST=NH; L=Kingston; O=My Company; OU=Engineering; CN=c4k1.stnh.mycompany.net; E=ausername@mycompany.com.
6. For the desired connection behavior, set the connection preferences as described in Table 8.



Note: If the Pulse connection is configured to use a list of Pulse servers, the preferred roles and realms you specify must be applicable to all of those servers.

Table 8: Configuration Options for Credential Provider Login

Pulse Client Credential Provider Login Behavior	Connection is established	User Connection Preferences	Pre-Login Connection Preferences	Machine Connection Preferences
At user login, the user can choose from two credential provider tiles: smart card login or username/password login. The credentials are then used to connect to the network, login to the endpoint, and login to the domain server.	Automatically at user login	Preferred User Realm and Preferred User Role Set are not available if you specify values for Preferred Pre-login Password Realm Preferred Pre-login Smartcard Realm.	Enables Pulse credential provider tiles. The realm name appears on each tile. You must specify values for both of the following options: <ul style="list-style-type: none"> Preferred Pre-login Password Realm—The authentication realm that provides username/password authentication. Preferred Pre-login Smartcard Realm—The authentication realm that provides smartcard authentication. 	Not available.
At machine login and at user login, the user can choose from two credential provider tiles: smart card login or username/password login.	Automatically when machine starts. Connection is authenticated again at user login.		Enables Pulse credential provider tiles. The realm name appears on each tile. <ul style="list-style-type: none"> Preferred Pre-login Password Realm—The authentication realm that provides username/password authentication. Preferred Pre-login Smartcard Realm—The authentication realm that provides smartcard authentication. 	Preferred Machine Realm and Preferred Machine Role Set are not available if you specify values for Preferred Pre-login Password Realm Preferred Pre-login Smartcard Realm.

Related Documentation

- [Configuring Location Awareness Rules for Pulse Secure Client](#)
- [Machine and User Authentication Through a Pulse Connection for Pulse Policy Secure](#)

Machine and User Authentication Through a Pulse Connection for Pulse Policy Secure

Pulse Secure client supports certificate authentication for establishing Layer 2 and Layer 3 connections. On Windows endpoints, a Pulse client connection accesses client certificates located in the Local Computer personal certificate store to provide machine authentication, or user certificates located in a user's personal certificate store or on a smart card for user authentication. A Pulse connection can access certificates from only one location. For information on machine authentication, see "[Machine Authentication for Pulse Policy Secure Overview](#)".

You can create a Pulse connection that uses System Local, Active Directory, or RSA ACE server authentication to verify the user and a certificate to verify machine identity before establishing a connection. To do so, you must first enable an option for the Pulse connection that allows the connection to check the client certificates located in the Local Computer personal certificate store. The option, Select client certificate from machine certificate store, is part of the User Connection Preferences of a Pulse connection. User authentication is accomplished through realm authentication. Machine authentication is accomplished as part of a realm certificate restriction, because the Pulse connection uses the machine

certificate. If the certificate store holds more than one valid certificate for the connection, Pulse opens a dialog box that prompts the user to select a certificate.

The following list summarizes the steps to configure a Pulse connection on a Windows endpoint that authenticates both the user and the machine. For detailed procedures on how to perform each configuration task, see the related documentation links.

- Install a machine authentication certificate in the Local Computer personal certificate store of the Windows endpoint and configure the Pulse server certificate server.
- Create a Pulse connection for the target Pulse server. The connection type can be UAC (802.1X) or Connect Secure or Policy Secure (L3). The Connection is established option is typically set to Manually by the user or Automatically at user login.
- In the User Connection Preferences section of the connection properties, click the check box labeled **Select client certificate from machine certificate store**. This option enables the Pulse connection to perform the machine authentication as part of the Pulse connection attempt.
- Create a sign-in policy on the Pulse server that specifies a user realm. The realm authentication server can be a System Local, Active Directory, or RSA ACE server.
- Configure a certificate restriction on the realm to enable the Pulse server to request a client certificate. Be sure to enable the option labeled Only allow users with a client-side certificate signed by Trusted Client CAs to sign in. Because the Pulse connection is configured to use the machine certificate, the user authentication takes place by means of the realm certificate restriction.

Related Documentation

- [Using Device Certificates](#)
- [Using the Certificate Server](#)
- [About Sign-In Policies](#)

Configuring Location Awareness Rules for Pulse Secure Client

The location awareness feature enables a Pulse Secure client to recognize its location and then make the correct connection. For example, you can define rules so that a Pulse client that is started in a remote location automatically establishes a VPN connection to Pulse Connect Secure, and then that same client automatically connects to Pulse Policy Secure when it is started in the corporate office. If Pulse detects that it is connected to the corporate LAN and it already has a VPN connection (for example, the VPN connection was suspended when the computer was put into hibernation), it first discovers that the VPN location awareness rules are no longer true, disconnects that VPN connection, and then evaluates the location awareness rules for the other configured connections.

Location awareness relies on rules you define for each Pulse connection. If the conditions specified in the rules resolve to TRUE, Pulse attempts to make the connection. If the conditions specified in the rules do not resolve to TRUE, Pulse tries the next connection. To set up the location awareness rules that select among many connections, you must define location awareness rules for each connection. Each location awareness rule is based on the endpoint's ability to reach an IP address or resolve a DNS name over a specified network interface.

The following location awareness example includes two connections. Each connection is configured to connect to only one target server. The first connection is a Pulse Policy Secure connection that resolves to TRUE when the endpoint is connected to the corporate LAN. The second connection is a Pulse Connect Secure connection that resolves to TRUE when the endpoint is located in a remote location. If Pulse detects that it is connected to the corporate LAN and it already has a VPN connection, it disconnects that VPN connection.

Pulse Policy Secure connection

If the DNS server that is reachable on the endpoint's physical network interface is one of your organization's internal DNS servers, then establish the connection.

Pulse Connect Secure connection

If the DNS server that is reachable on the endpoint's physical network interface is not one of your organization's internal DNS servers, and the DNS name of your Pulse Connect Secure device resolves to the external facing IP address of the Pulse Connect Secure device, then establish the connection.



Note: Connections can be set to manual, automatic, or controlled by location awareness rules. When the user logs in, the Pulse client attempts every connection in its connections list that is set to automatic or controlled by location awareness rules.



Note: To create a negative location awareness rule, you first create the positive state and then use rule requirement logic to use the rule as a negative condition.

To configure location awareness rules:

1. If you have not already done so, create a connection or open an existing connection.

You can configure location awareness rules for SRX connections and Connect Secure or Policy Secure (L3) connections. Location awareness rules do not apply to UAC (802.1X) connections.

2. Click the Mode list, and then select one of the options, **User**, **Machine**, or **Machine or user**.
3. If you selected **User** as the Mode, Under Options, select **Connect automatically**. If you selected **Machine** or **User or Machine**, **Connect automatically** is enabled by default.
4. Under Location awareness rules, click **New**.

Alternatively, you can select the check box next to an existing rule, and then click **Duplicate** to create a new rule that is based on an existing rule.

5. Specify a name and description for the rule.
6. In the Action list, select one of the following:
 - **DNS server**—Connect if the DNS server associated with the endpoint's network properties is (or is not) set to a certain value or set of values. Specify the DNS server IP address in the IP address box. Also specify a network interface on which the condition must be satisfied:
 - **Physical**—The condition must be satisfied on the physical interfaces on the endpoint.
 - **Pulse Secure**—The condition must be satisfied on the virtual interface that Pulse Secure creates when it establishes a connection.
 - **Any**—Use any interface.
 - **Resolve address**—Connect if the configured hostname or set of hostnames is (or is not) resolvable by the endpoint to a particular IP address. Specify the hostname in the DNS name box and the IP address or addresses in the IP address box. Also specify a network interface on which the condition must be satisfied.



Note: The Pulse client software evaluates IP and DNS policies on network interface changes. DNS lookups occur on DNS configuration changes or when the time-to-live setting (10 minutes) expires for a particular host record. If Pulse cannot resolve the host for any reason, it polls the configured DNS server list every 30 seconds. If the host had been resolved successfully previously and the time-to-live timer has not expired, the polling continues until the timer expires. If the host had not been resolved successfully previously, the resolution attempt fails immediately.

- **Endpoint Address**—Connect if a network adapter on the endpoint has an IP address that falls within or outside of a range or a set of ranges. Specify the IP address or addresses in the IP address box. Also specify a network interface on which the condition must be satisfied.
7. Click **Save Changes**.

After you create the rule or rules, you must enable each rule you want to use for the connection. To enable a negative form of a rule, use a custom version of the rule. To enable location awareness rules:

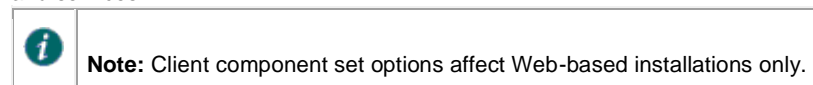
8. In the list of connection awareness rules for a connection, select the check box next to each rule you want to enable.
9. To specify how to enforce the selected location awareness rules, select one of the following options:
 - **All of the above rules**—The condition is TRUE and the connection is attempted only when all selected location awareness rules are satisfied.
 - **Any of the above rules**—The condition is TRUE and the connection is attempted when any select location awareness rule is satisfied.
 - **Custom**—The condition is TRUE and the connection is attempted only when all selected location awareness rules are satisfied according to the Boolean logic you specify in the Custom box. Use the Boolean condition to specify a negative location rule. For example, connect to Pulse Connect Secure when Rule-1 is false and Rule-2 is true. The Boolean logic in the custom box would be: **NOT Rule-1 AND Rule-2**. The accepted Boolean operators are AND, OR, NOT, and the use of ().
10. Click **Save Changes**.

Related Documentation

- [Understanding Session Migration](#)

Pulse Policy Secure Component Set Options

A Pulse Secure client component set includes specific software components that provide Pulse Secure client connectivity and services.



Component set options include the following choices:

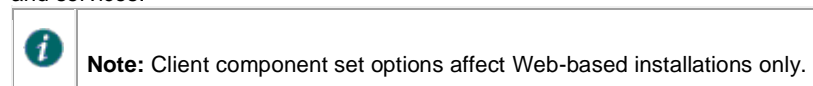
- **All components**—Supports all Pulse connection types. Use the **All components** option when you want client endpoints to be able to connect to all supported Pulse servers.
- **No components**—Updates existing Pulse client configurations, for example, to add a new connection. Do not use this option for a new installation.

Related Documentation

- [Pulse Secure Client Installation Overview](#)
- [Installing the Pulse Secure Client on Windows Endpoints Using a Preconfiguration File](#)
- [Installing the Pulse Secure Client on OS X Endpoints Using a Preconfiguration File](#)
- [Creating a Client Component Set for Pulse Policy Secure](#)

Creating a Client Component Set for Pulse Policy Secure

A Pulse Secure client component set includes specific software components that provide Pulse Secure client connectivity and services.



To create a client component set:

1. From the admin console, select **Users > Pulse Secure > Components**.
2. Click **New** to create a new component set.
3. If you have not yet created a client connection set, select **Users > Pulse Secure > Connections** and create a new connection set. Or you can use the default client configuration, which permits dynamic connections, supports the outer username anonymous, and allows the client to automatically connect to Pulse Policy Secure or Pulse Connect Secure.

4. Specify a name for the client component set.
5. (Optional) Enter a description for this client component set.
6. Select a connection set that you have created, or use the default connection set.
7. For Pulse Secure client components, select one of the following options:
 - **All components**—Supports all Pulse connection types.
 - **No components**—Updates existing Pulse client configurations, for example, to add a new connection. Do not use this setting for a new installation.
8. Click **Save Changes**.
9. After you create a component set, distribute the client to users through a role. When users access the role, the installer automatically downloads to the endpoint. The installer components and connections are applied to the endpoint client.

If client connections associated with the component set for a role are changed even though the list of components has not, the existing configuration on the endpoint is replaced immediately if the endpoint is currently connected, or the next time the endpoint connects.

If a user is assigned to multiple roles and the roles include different component sets, the first role in an endpoint's list of roles is the one that determines which client (component set) is deployed.

Related Documentation

- [Pulse Secure Client Installation Overview](#)
- [Installing the Pulse Secure Client on Windows Endpoints Using a Preconfiguration File](#)
- [Installing the Pulse Secure Client on OS X Endpoints Using a Preconfiguration File](#)
- [Endpoint Security Monitoring and Management for Pulse Policy Secure](#)

Endpoint Security Monitoring and Management for Pulse Policy Secure

You can configure Host Checker policies that verify the endpoint's operating system service pack, software version, or desktop application patch version compliance. Host Checker uses a list of the most current patch versions from the vendor for predefined rules in the Host Checker policy. Host Checker does not scan for non-security patches. Host Checker runs on Windows (Including Windows RT and Windows Phone) endpoints, Apple OS X and iOS endpoints, and on Google Android endpoints. The supported Host Checker features vary on each platform.



Note: Pulse Policy Secure Release 5.1 and later do not support custom patch assessment rules. The OPSWAT patch solution provides support for patch information monitoring and deployment. Host Checker downloads the OPSWAT SDK and uses it to detect the installed patch management software and the patch status (the list of missing patches as reported by the patch management software). To enable the patch management software to evaluate the patch status of the client machine, the administrator must configure a patch management policy to use for evaluating the patch status of endpoints.



Note: If a realm has a Host Checker policy enabled that is for desktop clients, and a mobile device user employs a browser on the mobile device to connect to the Web portal, the login is denied because the desktop Host Checker program is not compatible with the mobile client OS. If Pulse mobile users are mapped to multiple roles, the login operation assigns them to a role where Host Checker is not enabled if possible. If all the roles have Host Checker enabled, the mobile users will not be allowed to login from the browser. You can create and enable Host Checker policies that are specific to each mobile operating system and then Host Checker runs when the Pulse client connects to the server.

Pulse Policy Secure and Host Checker manage the flow of information between the corresponding pairs of TNC-based integrity measurement collectors (IMCs) and integrity measurement verifiers (IMVs). IMCs are software modules that run on the host and collect information such as antivirus, antispware, patch management, firewall, and other configuration and security information about the host. IMVs are software modules that run on Pulse Connect Secure and verify a particular

aspect of a host's integrity. Each IMV works with the corresponding IMC on the client endpoint to verify that the endpoint meets the Host Checker rules. IMCs scan the endpoint frequently for changes in security status. For example, if the user turns off virus checking, the IMC can detect this and then trigger a new check to make sure the modified system complies with the requirements of the Host Checker policy. You can configure Host Checker to monitor third-party IMCs installed on client computers by using third-party IMVs that are installed on a remote IMV server.

You obtain the most current patch version information from a Pulse Secure staging site. You can manually download and import the list into the Pulse Secure gateway, or you can automatically import the list from the Pulse Secure staging site or your own staging site at a specified interval.

Monitoring is based on one or more specified products or on specific patches, though not in the same policy. For example, you could check for Internet Explorer Version 7 with one policy, and Patch MSOO-039: SSL Certificate Validation Vulnerabilities with a second policy. Then, apply both policies to endpoints at the role or realm level to ensure that the user has the latest browser version with a specific patch. In addition, for Microsoft products, you can specify the severity level of patches that you want to ignore. For example, you could ignore low or moderate threats.

When you deploy Pulse Secure client, Host Checker is included with the installer. You can invoke Host Checker at the role level or the realm level to specify access requirements for endpoints seeking authentication. Host Checker policies that are implemented at the realm level occur before the user is authenticated. Host Checker policies at the role level are implemented after authentication but before the user is permitted to access protected resources. When an endpoint first connects to Pulse Connect Secure, the latest version of the IMC downloaded to the host computer. The initial check takes about 10-20 seconds to run. Outdated IMC files are automatically updated at subsequent checks.



Note: The first time an endpoint connects to a Pulse Connect Secure that has a patch assessment policy, if the connection is a Layer 2 connection, the IMC cannot download. In this case, you should configure a remediation role that displays instructions to direct the user to retry with a Layer 3 connection or to contact the administrator.

Remediation Options

Host Checker can identify issues on an endpoint. However, Host Checker and Pulse Connect Secure cannot resolve issues, that is, perform remediation tasks, on non-compliant endpoints. To repair those issues Pulse Connect Secure supports the following remediation options:

- Instructions to the user—The Pulse Connect Secure can send a message to the user describing the non-compliant patches or software and a link to where the user can obtain the required software. [Figure 46](#) shows a typical Pulse remediation message.

Figure 46: Pulse Remediation Instructions



- Initiate SMS/SCCM remediation—For remediation using Microsoft System Center Configuration Manager (ConfigMgr or SCCM), formerly Systems Management Server (SMS), a preinstalled SMS/SCCM client on the endpoint is triggered by Host Checker to get patches from a preconfigured SMS/SCCM server. This mechanism installs only those patches that are published on the SMS/SCCM server.

Related Documentation

- [Issuing a Remediation Message with Pulse Policy Secure](#)
- [Using SMS/SCCM Remediation with Pulse Policy Secure](#)
- [Patch Management Info Monitoring and Patch Deployment](#)

Issuing a Remediation Message with Pulse Policy Secure

If a Host Checker policy finds that an endpoint is not in compliance, Host Checker can display a message through the Pulse interface that includes custom instructions and reason strings on how to bring the endpoint into conformance. The user must perform the steps described in the message before the endpoint is allowed to access protected resources.

To enable a remediation message for a Host Checker policy:

1. In the admin console, select **Authentication > Endpoint Security > Host Checker**.
2. In the **Policies** section, click **New** to create a new Host Checker policy.

For detailed information about Host Checker Rule Settings, see the [Pulse Policy Secure documentation](#).

3. As part of the Host Checker Policy, select **Enable Custom Instructions**.

When you select this option, a text box appears. Enter the instructions to display to the user on the Host Checker remediation page. You can use the following HTML tags to format text and to add links to resources such as policy servers or web sites: `<i>`, ``, `
`, ``, and `<a href>`. For example:

You do not have the latest signature files.

`Click here to download the latest signature files.`

4. Optionally, select **Send reason strings**. Select this option to display a message to users (called a reason string) that is returned by Host Checker or IMV and that explains why the client machine does not meet the Host Checker policy requirements. Reason strings describe to users what the IMV is checking on the client machine. This option applies to predefined rules, to custom rules, and to third-party IMVs that use extensions in the Pulse Secure TNC SDK.
5. Click **Save Changes**.

Be sure to include the Host Checker policy in the realm or role you configure for Pulse users.

Related Documentation

- [Endpoint Security Monitoring and Management for Pulse Policy Secure](#)
- [Using SMS/SCCM Remediation with Pulse Policy Secure](#)

Using SMS/SCCM Remediation with Pulse Policy Secure

Pulse Secure client supports the SMS/SCCM download method for patch deployment. If the Pulse Policy Secure is configured for the SMS/SCCM method for patch deployment, the Pulse client endpoint must have the SMS/SCCM client already installed on the endpoint, otherwise remediation fails.

Endpoints configured with SMS/SCCM for software management typically poll the server for updates every fifteen minutes or longer. In a worst-case scenario, clients that are not in compliance with existing Host Checker software requirements might have to wait until the next update interval to login. Using the SMS/SCCM download method, you can force the client to initiate the software update immediately after the patch assessment check. If a user attempts to log in, and the endpoint does not have a required software version for compliance with a Host Checker patch assessment policy, Host Checker immediately notifies the client to poll the server for an immediate update. The client receives notification that an SMS/SCCM update has started.

To configure SMS/SCCM to update the client when notified, set the advertisement time on the SMS/SCCM to As soon as possible.

You assign clients to a particular group or collection on the SMS/SCCM server and then server can advertise patches for that collection. You can configure roles on the Pulse Policy Secure that correspond to collections and SMS/SCCM can send the appropriate patches for a particular role.

You must have the SMS/SCCM client installed and configured correctly on endpoints, and the SMS/SCCM server must be reachable. In a Layer 2 network, Host Checker is performed before the endpoint is connected to the network. Host Checker can obtain the IP address of the SMS/SCCM server configured for the client. If the endpoint is out of compliance and remediation is necessary, Host Checker pings the server IP address every 15 seconds until the server can be notified to update the client.

You should inform users of the expected behavior if this feature is enabled, as there is no notification to the user until the SMS/SCCM sends back the advertisement.

To enable SMS/SCCM assessment and remediation:

1. In the admin console, select **Authentication > Endpoint Security > Host Checker**.
2. In the **Policies** section, click **New** to create a new Host Checker policy.
3. Under Patch Remediation Options, select **SMS/SCCM Patch Deployment**.

4. Click **Save Changes**.

Be sure to include the Host Checker policy in the realm or role you configure for Pulse users.

Related Documentation

- [Endpoint Security Monitoring and Management for Pulse Policy Secure](#)
- [Issuing a Remediation Message with Pulse Policy Secure](#)

Patch Management Info Monitoring and Patch Deployment

Configuration and Migration Options for Deprecated Custom: Patch Assessment Rules

With Release 8.1/5.1, the OPSWAT patch solution provides support for patch information monitoring and deployment. Host Checker downloads the OPSWAT SDK and uses it to detect the installed patch management software and the patch status (the list of missing patches as reported by the patch management software). To enable the patch management software to evaluate the patch status of the client machine, the administrator must configure a patch management policy to use for evaluating the patch status of endpoints.

Custom patch assessment rules are not supported beginning in Release 8.1/5.1. The existing patch management rules will be converted to dummy rules during the migration. You can delete the existing rules or convert them to predefined: patch management rules.

To delete the custom patch assessment rules.

1. Select **Authentication > Endpoint Security > Host Checker**.
2. Select the check box to back up the configuration and the XML file that contains Host Checker, realms, and role details.

[Figure 47](#) shows the configuration page for Host Checker.

3. Under Delete deprecated Custom: Patch Assessment rules, select **Delete**.

Result

Displays a confirmation page with the list of deprecated Custom:Patch Assessment rules and the policies in which they are configured. It also lists the Rule Expression for the respective policies which will be changed and the list of policies that becomes empty because of deletion of above rules. You need to click on Confirm if you want to continue deletion of deprecated rules, otherwise click on Cancel.

To convert the existing Shavlik rules to Opswat rules:

1. Select **Authentication > Endpoint Security > Host Checker**.
2. Select the check box to back up the configuration and the XML file that contains Host Checker, realms, and role details.

[Figure 47](#) shows the configuration page for Host Checker.

Figure 47: Delete or Convert the Deprecated Patch Assessment Rules

'Custom: Patch Assessment' rules are deprecated. Please use below options to delete these rules or to convert these rules to 'Predefined: Patch Management' rules.

The screenshot shows two main sections in a web interface:

- Delete deprecated 'Custom: Patch Assessment' rules:**
 - A checkbox labeled "Backup 'User Configuration' and 'XML containing configured Host Checker, Realms and Roles details'" is checked.
 - A "Delete..." button is present.
 - A note states: "Note: This deletes deprecated 'Custom: Patch Assessment' rules and their usage in policies. If this results in empty policies (policies with no rules configured), these policies will be removed and accordingly host checker policy restrictions on Roles and Realms will be updated."
- Convert deprecated 'Custom: Patch Assessment' rules to 'Predefined: Patch Management' rules:**
 - The same backup checkbox is checked.
 - A label reads: "Select Patch Management software product name that needs to be detected:".
 - A dropdown menu is open, showing a list of products:
 - Select Product Name -
 - Select Product Name -
 - BigFix Enterprise Client (8.x)
 - Security and Patch Manager (8.x)
 - Security and Patch Manager (9.x)
 - Microsoft Windows AutomaticUpdate (7.x)
 - Microsoft Windows Update Agent (7.x)
 - System Center Configuration Manager (4.x)
 - System Center Configuration Manager (5.x)
 - Below the dropdown, there are buttons: "New...", "New 3rd Party Policy...", and "Delete...".
 - A note at the bottom says: "You may download a Host Checker installer from the [installers](#) page."

3. Select the patch management software that you will use to convert custom patch assessment rules to predefined patch management rules and then click on convert.

i

Note: Convert button appears only after selecting the Patch management Software. If you select convert you can see the confirmation page which lists the deprecated Custom:Patch Assessment rules and the policies in which they are configured. It also lists the Rule Expression for the respective policies which will be changed. Click Confirm to continue replacement of deprecated Custom:Patch Assessment rules with Predefined: Patch Management rules, otherwise click Cancel.

Using a System Management Server

You can use a System Management Server (SMS) to provide a method for automatic updates to non-compliant software. From Release 8.1/5.1, only SMS/SCCM patch remediation is supported. You can enable SMS/SCCM patch remediation in the Predefined patch management policy page. The client machine must have the SCCM client installed and must be communicating to the SCCM server.

Related Documentation

- [Configuring Patch Remediation Options](#)
- [Configuring a Predefined Host Checker Patch Management Rule](#)

Pushing Pulse Secure Client Configurations Between Pulse Servers of the Same Type

You can use the Push Configuration feature to centrally manage Pulse Secure Connections, components, and uploaded Pulse packages. The Push Configuration feature enables you to copy all configuration settings or selected configuration settings from one Pulse server to another Pulse server of the same type, for example, from one Pulse Connect Secure server to another Pulse Connect Secure server.

The following notes apply to pushing configurations:

- You can push to a single Pulse server or to multiple Pulse servers in one operation. You can push up to 8 targets per push operation. You can run up to 25 push operations simultaneously. The maximum number of targets is 200. If a push to a target Pulse server fails, the operation proceeds to the next target until all identified targets are updated. The results page displays the status and any problems encountered during the process.
- You can push to a Pulse server that is a member of a cluster as long as the target Pulse server is not a member of the same cluster as the source.
- Target Pulse servers can refuse pushed configuration settings. The default is to accept.
- After an update, the target Pulse server restarts its services. Brief interruptions might occur while the service restarts. We recommend that you push to targets when they are idle or when you can accommodate brief interruptions.
- Target Pulse servers do not display a warning message when they receive a pushed configuration.
- The target Pulse server automatically logs out administrators during the push process.
- The source and target Pulse servers must have the same build version and number.
- The administrator account on the source Pulse server must sign in to the target Pulse server without any human intervention. For example, you cannot have dynamic credentials or multiple roles that are not merged as these both require manual interaction.

Before you use Push Configuration, you must configure your system according to the following conditions:

- You must map to the Administrators role, thereby creating a “super administrator” with full administration privileges. Modify Authentication > Auth Servers > Administrator Server > Users settings to add yourself to the Administrators role.
- The target Pulse server administrator account must use static password authentication or two-factor tokens that do not use challenge/response type authentication. For example, certificates, Soft ID, and Defender Authentication are not supported. Modify Administrators > Admin Realms > [Administrator Realm] > General settings to select the proper authentication server for the administrator realm.
- Do not configure the administrator account in a way that requires the administrator to select a role to sign in to the target Pulse server. For example, do not map a single user to multiple roles, including the Push Configuration administrator role, and then fail to merge those roles. We recommend creating an account exclusively for Push Configuration administrators to guarantee that the administrator does not need to choose a role during the sign-in process and to clearly distinguish the actions of Push Configuration administrators in your log files. Use the Administrators > Admin Realms > [Administrator Realm] > Role Mapping settings to set the appropriate role-mapping rules.

To push Pulse configurations from one Pulse server to other Pulse servers of the same type:

1. If you have not already done so, define the targets by selecting **Maintenance > Push Config > Targets**.
2. From the admin console, select **Maintenance > Push Config > Push Configuration**.
3. In the What to push box, select **Selected configuration** to display the configuration categories.
4. Scroll down the list and expand the item labeled Pulse Secure.
5. Select the **Select All Configurations** check box to push all Pulse configurations on this Pulse server. Or chose none, all, or selected items from the following categories:
 - **Pulse Secure Connections**—Connection sets and connections.
 - **Pulse Secure Components**—Component sets.
 - **Pulse Secure Versions**—Pulse packages that were uploaded to the Pulse server.
6. Add the targets to the **Selected Targets** box.
7. Click **Push Configuration**.

Related Documentation

- [Enabling or Disabling Automatic Upgrades of the Pulse Secure Client](#)

Enabling or Disabling Automatic Upgrades of the Pulse Secure Client

After you deploy Pulse Secure client software to endpoints, software updates occur automatically. If you upgrade the Pulse client configuration on your Pulse server, updated software components are pushed to a client the next time it connects.



Note: If you configure Pulse Secure client to make 802.1X based connections, a reboot might be required on Windows endpoints when Pulse is upgraded.



Note: A bound endpoint receives connection set options and connections from its binding server, but it can have its Pulse client software upgraded from any Pulse server that has the automatic upgrade option enabled. During a client software upgrade the client loses connectivity temporarily.

Pulse client software upgrades are enabled by default. To change the behavior of Pulse client upgrades:

1. From the admin console, select **Maintenance > System > Options**.
2. Set or clear the **Enable automatic upgrade of Pulse Secure Clients** check box.
3. Click **Save Changes**.

Related Documentation

- [Upgrading Pulse Secure Client](#)

Upgrading Pulse Secure Client

The software image for each supported Pulse server includes a Pulse Secure client software package. When a newer version of Pulse is available, you can upload the new software to the Pulse server. You can have more than one version of Pulse on a Pulse server but only one Pulse client package can be active. If you activate a new version of Pulse, and if the Pulse server's automatic upgrade option is enabled, connected Pulse clients display an upgrade prompt to the user. The user can choose to install the upgrade or cancel the operation. If a user cancels, the upgrade prompt appears each time the client connects to the server. During a client software upgrade the Pulse client loses connectivity temporarily.

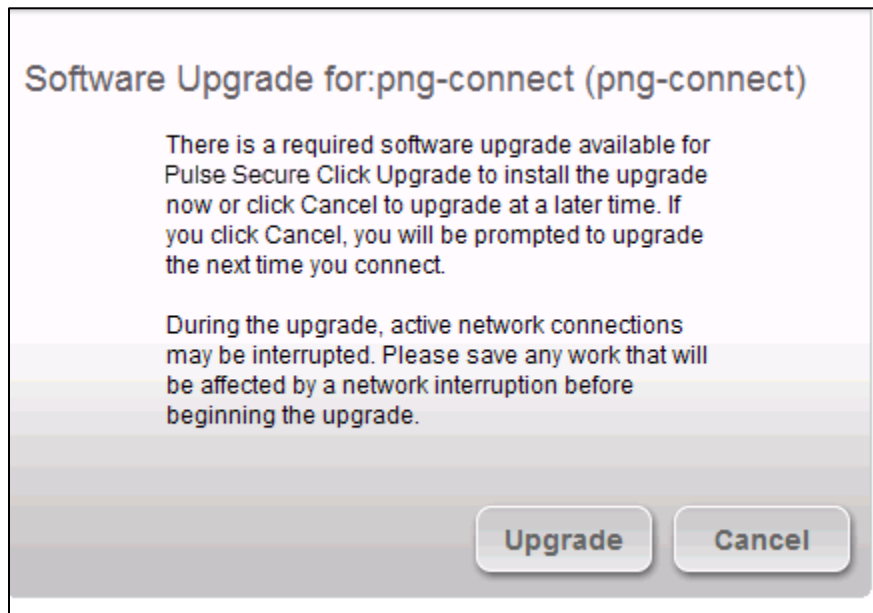


Note: The automatic update feature is supported on Pulse Connect Secure and Pulse Policy Secure servers only. SRX gateways do not support automatic Pulse software updates.



Note: If you configure Pulse to make 802.1X based connections, a reboot might be required on Windows endpoints when Pulse is upgraded.

Figure 48: Pulse Client Upgrade Message



After you have staged the new Pulse software package in a location accessible to the Pulse server, use the following procedure to upload the software to the Pulse server:

1. In the device admin console, select **Users > Pulse Secure > Components**.
2. In the section labeled Manage Pulse Secure Client Versions, click **Browse**, and then select the software package.
3. Click **Upload**.

Only one Pulse Secure client software package can be active at a time. After you upload a new package, you need to enable it.

To enable a Pulse package as the default:

1. In the admin console, select **Users > Pulse Secure > Components**.
2. In the section labeled Manage Pulse Secure Client Versions, select the radio button next to a version, and then click **Activate**.

Related Documentation

- [Enabling or Disabling Automatic Upgrades of the Pulse Secure Client](#)

Using Device Certificates

This topic describes how to use device certificates. It includes the following information:

- [Understanding Device Certificates](#)
- [Understanding Self-Signed Certificates](#)
- [Importing a Device Certificate and Private Key](#)
- [Creating a Certificate Signing Request](#)
- [Importing a Signed Certificate Created from a CSR](#)
- [Understanding Intermediate Certificates](#)
- [Importing Intermediate CA Certificates](#)
- [Importing a Renewed Certificate That Uses the Existing Private Key](#)

- [Downloading a Device Certificate](#)
- [Using Device Certificates with Virtual Ports](#)

Understanding Device Certificates

A device certificate helps to secure network traffic to and from the Pulse Secure client service using elements such as your company name, a copy of your company's public key, the digital signature of the Certificate Authority (CA) that issued the certificate, a serial number, and an expiration date. The system also uses device certificates for secure communications with the Infranet Enforcer.

When receiving encrypted data from the system, the client's browser first verifies whether the device certificate is valid and whether the user trusts the CA that issued the certificate. If the user has not already indicated that they trust the certificate issuer, the Web browser prompts the user to accept or install the certificate.

The system supports X.509 device certificates in DER and PEM encode formats (file extensions include .cer, .crt, .der, and .pem) as well as PKCS #12 (file extensions include .pfx and .p12). The system also supports the following features:

- Intermediate device CA certificates—Within a certificate hierarchy, one or more intermediate certificates are branched off a single root certificate.
- Multiple device certificates—When using multiple device certificates, each certificate handles validation for a separate hostname or fully qualified domain name (FQDN) and can be issued by a different CA.




Note: Beginning with Connect Secure system software release 7.2, you can assign device certificates to the Connect Secure VLAN interfaces.

Understanding Self-Signed Certificates

When you initialize the system with the serial console, the system creates a self-signed certificate that enables you to immediately begin setting up the system. Users are prompted with a security alert each time they sign in because the certificate is not issued by a trusted CA. [Figure 49](#) shows the security alert.

Figure 49: Security Alert When the Device Certificate Is Not Issued by a Trusted CA





There is a problem with this website's security certificate.


The security certificate presented by this website was issued for a different website's address.
The security certificate presented by this website was not issued by a trusted certificate authority.

Security certificate problems may indicate an attempt to fool you or intercept any data you send to the server.

We recommend that you close this webpage and do not continue to this website.

 [Click here to close this webpage.](#)

 [Continue to this website \(not recommended\).](#)

 [More information](#)

Before promoting the system to production use, we recommend you replace the self-signed certificate with a certificate issued by a trusted CA.



Note: In Policy Secure deployments with ScreenOS Enforcers, you must use a CA-signed device certificate. If you use a self-signed certificate, the ScreenOS Enforcer does not allow a connection. Import a CA-signed device certificate into the Policy Secure and then import the certificate of the CA that signed the device certificate into the ScreenOS Enforcer.

Importing a Device Certificate and Private Key

The system uses certificates to verify itself to other network devices. A digital certificate is an electronic means of verifying your identity through a trusted third party, known as a Certificate Authority (CA). Your company might use its own enterprise CA server, or it might use a reputable third-party CA.

To import an enterprise root server certificate and private key:

1. Select **System > Configuration > Certificates > Device Certificates**.
2. Click **Import Certificate & Key** to display the configuration page.
3. Use one of the following options to complete the import procedure:
 - **If certificate file includes private key**—When the certificate and key are contained in one file.
 - **If certificate and private key are separate files**—When the certificate and key are in separate files.
 - **Import via System Configuration file**—When the certificate and key are contained in a system configuration file. With this option, the system imports all of the certificates specified (including private keys and pending CSRs, but not the corresponding port mappings).

In the appropriate form, browse to the certificate and key files. If the file is encrypted, enter the password key.

4. Click **Import**.



Note: The Import Certificate and Key button is disabled on FIPS hardware platforms because importing private keys is not allowed. On a FIPS hardware platform, you must create a CSR and then import a signed certificate from the CA.

Creating a Certificate Signing Request

If your company does not own a digital certificate for its Web servers, you can create a certificate signing request (CSR) and then send the request to a CA for processing. When you create a CSR, a private key is created locally that corresponds to the CSR. If you delete the CSR at any point, this file is also deleted, prohibiting you from installing a signed certificate generated from the CSR.

To create a certificate signing request:

1. Select **System > Configuration > Certificates > Device Certificates**.
2. Click **New CSR** to display the configuration page.
3. Complete the required information and click **Create CSR**.
4. Follow the onscreen instructions, which explain what information to send to the CA and how to send it.

When you submit a CSR to a CA authority, you might be asked to specify either the type of Web server on which the certificate was created or the type of Web server the certificate is for. Select **apache** (if more than one option with apache is available, select **any**). If you are prompted for the certificate format to download, select the standard format.

Do not send more than one CSR to a CA at one time. Doing so can result in duplicate charges.



Note: To view details of any pending requests that you previously submitted, click the **Certificate Signing Request Details** link.

Importing a Signed Certificate Created from a CSR

When you receive the signed certificate from the CA, import it.

To import a signed device certificate created from a CSR:

1. Select **System > Configuration > Certificates > Device Certificates**.
2. Under Certificate Signing Requests, click the **Pending CSR** link that corresponds to the signed certificate.
3. Under Import signed certificate, browse and select the certificate file you received from the CA, and then click **Import**.

Understanding Intermediate Certificates

Within a certificate hierarchy, one or more intermediate certificates are branched off a single root certificate. The root certificate is issued by a root CA and is self-signed. Each intermediate certificate is issued by the certificate preceding it in the chain.

To use chained certificates in your deployment, you must ensure that the server (Policy Secure or Connect Secure) and client (Web browser) together contain the entire certificate chain. For example, you can secure traffic using a chain that stems from a VeriSign root certificate. If your users' browsers come preloaded with VeriSign root certificates, you need to install only the lower-level certificates in the chain. When your users sign in, the system presents any required certificates within the chain to the browser to secure the transaction. The system creates the proper links in the chain using the root certificate's IssuerDN. If the system and browser together do not contain the entire chain, the user's browser does not recognize or trust the device certificate because it is issued by another certificate instead of by a trusted CA.

You can upload one or more intermediate CAs in a PEM file. The entire chain must be sent to the client in descending order, starting with the root certificate.

Within a certificate hierarchy, one or more intermediate certificates are branched off a single root certificate. The root certificate is issued by a root CA and is self-signed. Each intermediate certificate is issued by the certificate preceding it in the chain.

To use chained certificates in your deployment, you must install the appropriate client-side certificates in each user's Web browser and then upload the corresponding CA certificates to Pulse Secure client Service Intermediate CA store. Use one of the following methods to upload the certificate chain:

- Import the entire certificate chain in one file. The file must contain the root certificate and any sub certificates whose parents are in the file or already imported. You can include certificates in any order in the import file.
- Import the certificates one at a time in descending order. You must install the root certificate first, and then install the remaining chained certificates in descending order.

If you follow one of these methods, the system automatically chains the certificates together in the correct order and displays them hierarchically in the admin console.



Note: If you install multiple certificates in a user's Web browser, the browser prompts the user to choose which certificate to use when signing in.

Importing Intermediate CA Certificates

To import an intermediate CA certificate:

1. Select **System > Configuration > Certificates > Device Certificates**.
2. Click the **Intermediate Device CAs** link to display the management page.
3. Click **Import CA certificate**.
4. Browse to the certificate file, select it, and click **Import Certificate** to complete the import operation.

Importing a Renewed Certificate That Uses the Existing Private Key

You can renew a device certificate in two ways:

- Submit a new CSR to a CA—This process is more secure because the CA generates a new certificate and private key and retires the older private key. To use this renewal method, you must first create a CSR through the admin console.
- Request renewal based on the CSR previously submitted to the CA—This process is less secure, because the CA generates a certificate that uses the existing private key.

When you order a renewed certificate, you must either resubmit your original CSR or ensure that the CA has a record of the CSR that you submitted for your current certificate.

To import a renewed device certificate that uses the existing private key:

1. Follow your CA's instructions for renewing a certificate that you previously purchased through them. Be sure to specify the same information you used in the original CSR. Your CA uses this information to create a new certificate that corresponds to the existing key.



Note: Even though you specify the same information used in the original CSR, your root CA might have different serial numbers and keys from the original. You might need to support both new client and old client certificates during the transition period, which also requires that you maintain two root CA certificates (your existing certificate and the renewed certificate), at least temporarily.

2. Select **System > Configuration > Certificates > Device Certificates**.
3. Click the link that corresponds to the certificate you want to renew.
4. Click **Renew Certificate** to display the page.
5. In the Renew the Certificate form, browse to the renewed certificate file, enter the password for the certificate key, and click **Import**.

Downloading a Device Certificate

You download the device certificate to your local host so that you can import it into other network devices as needed.

To download a device certificate:

1. Select **System > Configuration > Certificates > Device Certificates**.
2. Click the link of the device certificate you want to download to display the configuration page.
3. Click the **Download** link.
4. Save the file to the desired location.

Using Device Certificates with Virtual Ports

Virtual ports can be used to create multiple fully qualified domain names for user sign-in.

When a user tries to sign in using the IP address defined in a virtual port, the system uses the certificate associated with the virtual port to initiate the SSL transaction and for NetScreen Address Change Notification (NACN) communications with the Infranet Enforcer.

You must associate the signed certificate with the port that is connected to the Infranet Enforcer. You can use the same port and certificate for OAC or Pulse. Or, you can import other signed certificates and associate them with ports connected to OAC.

You can implement digital certificate security with virtual ports in either of the following ways:

- Associate all hostnames with a single certificate—With this method, you use a single wildcard certificate to validate the identity of all system hostnames, regardless of which hostname is used to sign into. A wildcard certificate includes a variable element in the domain name, making it possible for users who sign in from multiple hosts to map to the “same” domain. For example, if you create a wildcard certificate for *.yourcompany.com, the system uses the same certificate to validate its identity to users who sign in to employees.yourcompany.com as it does to users who sign into partners.yourcompany.com.
- Associate each hostname with its own certificate—With this method, you associate different hostnames with different certificates. Create a virtual port for each hostname. A virtual port activates an IP alias on a physical port. For example, you can create two virtual ports on a single appliance, mapping the first virtual port to the IP address 10.10.10.1 (sales.yourcompany.com) and the second virtual port to the IP address 10.10.10.2 (partners.yourcompany.com). Then you can associate each of these virtual ports with its own certificate, ensuring that users authenticate through different certificates.

To associate certificates with virtual ports:

1. Create the virtual ports.
2. Import the device certificates.
3. Associate the device certificates with the virtual ports:
 1. Select **System > Configuration > Certificates > Device Certificates**.
 2. Click the link of the device certificate you want to configure to display the configuration page.
 3. Use the controls in the “Present certificate on these ports” section to associate ports with the certificate.



Note: You can assign only one device certificate to the Management Port. If you assign a certificate other than the default device certificate to the Management Port, the default device certificate is automatically deselected as the default. If you do not select a device certificate for the Management Port, the system uses the default device certificate that is presented on the Internal port. You cannot assign certificates to Management Port VIPs.

CHAPTER 3 Configuring Pulse Connect Secure

- [Before You Begin](#)
- [Pulse Connect Secure Overview](#)
- [About Sign-In Notifications](#)
- [Configuring and Implementing Sign-in Notifications](#)
- [Pulse Connect Secure Split Tunneling Overview](#)
- [Configuring a Role for Pulse Connect Secure](#)
- [Machine Authentication for Pulse Connect Secure Overview](#)
- [Credential Provider Authentication for Pulse Connect Secure Overview](#)
- [Configuring User-at-Credprov Credential Provider Authentication for a Pulse Secure Connection](#)
- [Configuring Machine-Then-User-at-Credprov Credential Provider Authentication for a Pulse Secure Connection](#)
- [Machine and User Authentication Through a Pulse Connection for Pulse Connect Secure](#)
- [Stealth Mode](#)
- [Configuring Pulse Secure Client for Secure Application Manager](#)
- [Pulse Connection Set Options for Pulse Connect Secure](#)
- [Securing the Connection State on the Pulse Secure Client](#)
- [Creating a Client Connection Set for Pulse Connect Secure](#)
- [Pulse Secure Client FIPS Mode for Pulse Connect Secure Overview](#)
- [Configuring Location Awareness Rules for Pulse Secure Client](#)
- [Component Set Options for Pulse Connect Secure](#)
- [Creating a Client Component Set for Pulse Connect Secure](#)
- [Endpoint Security Monitoring and Management for Pulse Connect Secure](#)
- [Issuing a Remediation Message with Pulse Connect Secure](#)
- [Using SMS/SCCM Remediation with Pulse Connect Secure](#)
- [Pushing Pulse Configurations Between Pulse Servers of the Same Type](#)
- [Enabling or Disabling Automatic Upgrades of the Pulse Secure Client](#)
- [Upgrading Pulse Secure Client](#)
- [Pulse Collaboration Suite Overview](#)

Before You Begin

Before you begin configuring Pulse Secure client, be sure that you have already configured Pulse Connect Secure server network settings. Also, be sure that you have defined the Authentication settings, including the authentication servers and sign-in settings. The Authentication and Host Checker settings can directly affect a Pulse installation because you can define the conditions that an endpoint must meet to be allowed access to protected resources.

Pulse Connect Secure Overview

To enable Pulse Connect Secure, you configure the service so that when users request authentication, they are assigned a role based on the role mappings and optional security profile that you create. Access to specific resources is permitted only for users and devices that provide the proper credentials for the realm, that are associated with the appropriate roles, and whose endpoints meet security restrictions. If a user attempts to connect to the network from an endpoint that does not comply with the security restrictions you have defined, the user cannot access the realm or role.

As you plan your Pulse configuration, be sure you know how you want to deploy the Pulse client software. You can use one or more of the following Pulse deployment options:

- Use the defaults or make changes to the Pulse Secure default component set and default connection set, and then download and distribute Pulse by having users log in to the Pulse server's user Web portal and be assigned to a role. After the installation is complete, users have all the connections they need to access network resources.
- Create the connections that an endpoint needs for connectivity and services, download the settings file (.pulsepreconfig), and download default Pulse installation program. For Windows endpoints you run the Pulse installation program by using an msixec command with the settings file as an option. For OS X endpoints, you run the default installer and then import the .pulsepreconfig file using a separate command.
- Distribute Pulse Secure client with no preconfiguration. You can download the default Pulse Secure installation file (.msi format for Windows; .dmg format for Mac) from a Pulse server, and then distribute the file to endpoints using your organization's standard software distribution methods. Because the installer does not contain preconfigured connections, users must define network connections manually. Or you can create dynamic connections on each Pulse server. These connections are automatically downloaded to the installed Pulse client when users provide their login credentials to the Pulse server's user Web portal, and then starts Pulse through the Web portal interface. Dynamic connections are created as manual rather than automatic connections, which means that they are run only when the user initiates the connection or the user browses to a Pulse Server and launches Pulse from the server's Web interface.

Note: For a Windows installation (.msi) that uses an automated distribution mechanism and where the users do not have administrator privileges, you should ensure that the installation is run in the proper context, typically the USER context. To install in USER context, first advertise the .msi while in the SYSTEM context. For example, to advertise the 64-bit Windows installation to all users, use the following msixec command:



msixec /jm \PulseSecure.x64.msi

The advertisement allows the installation to be run in USER context even if the user is a restricted (non-admin) user. The location where the advertisement is run and where the actual installation is run must be the same. If the installation is an upgrade, you must advertise the upgrade version before running it. (Note that it is much easier to upgrade the Pulse client by not disabling the automatic upgrade feature on the Pulse server.) After the installation is run by the user, the Pulse client will use the correct user certificate and context.

The following tasks summarize how to configure Pulse Connect Secure:

- Create and assign user roles to control who can access different resources and applications on the network. If you are converting your access environment from agentless or a Pulse Secure Client environment, you should create new roles that are specific for Pulse Secure client.
- Define security restrictions for endpoints with Host Checker policies.
- Define user realms to establish authentication domains. If you are converting your access environment from agentless or a Pulse Secure Client environment, typically you can use your existing realms.
- Associate the roles with appropriate realms to define your access control hierarchy using role mapping.
- Define Pulse Secure client component sets, connection sets, and connections.
- Deploy Pulse Secure client to endpoints.

Pulse Secure Client and IVS

Pulse Connect Secure and Pulse Secure Client do not support Instant Virtual System (IVS) feature anymore.

Pulse Secure Client and Traffic Enforcement

The Traffic Enforcement feature (supported on Windows and macOS) enables the user to prevent the leakage of any packet out of the tunnel as per PCS tunnel configuration. This is accomplished by applying firewall rules on the Pulse Desktop Client. These rules are created based on the PCS tunnel configuration.

For more information on PCS tunnel configuration policies, refer to the section “Defining Split Tunneling Network Policies” and “Defining the Route Precedence Options” of chapter “VPN Tunneling” of Pulse Connect Secure Administration Guide.

A local program might bypass the routing tables and bind traffic to the physical interface instead of allowing it to go through the Pulse virtual interface. If you enable Traffic Enforcement, you ensure that all traffic is bound by the PCS tunnel configuration. Traffic Enforcement feature is more useful in macOS because of Apple routing behavior.

For example, If SSH session is created using physical adapter before VPN tunnel, the session will continue to use physical adapter even after the tunnel is established because of macOS scoped routing (Apple functionality). If Traffic Enforcement is enabled, the same SSH session gets terminated because firewall rule finds packet leaking out of tunnel from SSH session and it will deny that traffic.

Related Documentation

- [Creating a Client Connection Set for Pulse Connect Secure](#)
- [Configuring a Role for Pulse Connect Secure](#)

Advanced Client Configuration Feature

This topic describes the XML advanced client configuration that can be used by the PCS administrator to configure the custom settings, which are meant to solve a specific customer scenario without changing the PCS admin console. Admin can set these custom settings in the form of XML input through the Advanced Client Configuration UI feature. Pulse clients supporting these custom settings will consume them when connecting to this PCS, and the same would be applied on the client machines. From 9.0R3 release onwards, this feature will minimize the number of changes going into the PCS admin console to fulfill the requirement of a specific customer.

In the earlier Pulse client releases, i.e. prior to v5.2R2, the virtual adapter MTU was calculated based on the physical adapter MTU (of the host machine) and the MTU sent by the PCS.

Basically, the formula used to calculate the virtual adapter MTU is:

MIN (Physical Adapter MTU, MTU from PCS, TCP MSS value + 40)

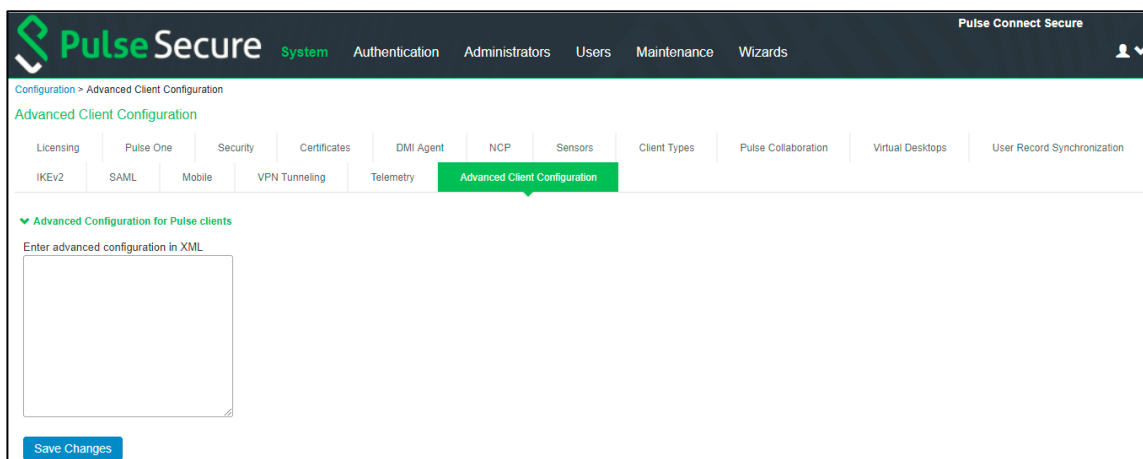
Following is a scenario where Firewall on the data path is stripping the TCP MSS options being advertised by the SA/PCS to the Pulse client. In this scenario, the TCP MSS value on the Pulse client will default to a minimum value of 536, and as a result the client side MTU calculation will result in a minimum MTU value of 576. Here, customer wants to ignore the TCP MSS options while calculating the Virtual Adapter MTU calculation.

If the administrator configures the Pulse Connect Secure sever with the following XML input in "Advanced Client Configuration for Pulse Client" option, it will ignore TCP MSS options while calculating the virtual adapter MTU on client side.

1. Select System > Configuration > Advanced Client Configuration to display the configuration page.

The following figure shows the configuration page for Pulse Connect Secure.

Figure 50: Advanced Client Configuration



2. Enter the following XML input in "Advanced Client Configuration for Pulse Client".

```
<advanced-config>
  <version>9.0.3</version>

  <desktop-client-config>
    <layer3-connection-config>
      <adapter-config>
        <ignore-tcp-mss>TRUE</ignore-tcp-mss>
      </adapter-config>
    </layer3-connection-config>
  </desktop-client-config>
</advanced-config>
```

3. Click Save Changes.

The advanced configuration setting "ignore-tcp-mss" is Layer3 Adapter configuration setting and this will be consumed by the Pulse client as part of the IpsecConfig.



Note: This "ignore-tcp-mss" setting is applicable for the virtual adapter MTU calculation only for IPv4. By default the setting is always false, and therefore the TCP MSS options are always considered for MTU by default. Admin has to explicitly set the ignore-tcp-mss setting to TRUE (case-insensitive), to ignore the TCP MSS.


About Sign-In Notifications

With sign-in notifications, you can create and configure detailed notification messages that appear for Pulse clients and for agentless access endpoints when the user attempts to sign in. For example, you could configure a notification message that explains terms of use, company-specific policies, a welcome page, an end user license agreement (EULA), or a message of the day (MOTD).

For a browser-based (agentless) login, the notification message appears in a separate page either before (pre-auth) or after (post-auth) user authentication during the sign-in process. For a Pulse client login, the notification messages appear in a Pulse message box. The user is expected to read the content of the sign-in notification message and acknowledge by clicking a Proceed button. The user may indicate disagreement by clicking a Decline button, which ends the login attempt.

You can configure a sign-in policy to use a sign-in notification either as pre-auth or post-auth (or both). In the case of post-auth configuration, you can either use a common message for all roles or use separate messages for each role.

You can create a multi-language sign-in notification package that relies on the language setting of the endpoint. You can customize the sign-in notification page appearance for browser-based logins by modifying the related fields in a sign-in page in the Admin UI or by using a custom sign-in page.

	<p>Note:</p> <ul style="list-style-type: none"> • Sign-in notifications are supported on Windows, Mac, and for browser-based access on mobile devices. However, sign-in notifications might not work well with all mobile devices due to device limitations. • Sign-in notifications (including uploaded packages) are included in XML exports. • If a Pulse session is resumed or extended, the pre-auth notification message is not shown again. However, if the user switches roles when resuming a session, and that role change results in a new notification, Pulse displays the message. You can configure the post-auth message to be skipped if it has already been seen. If the post-auth message is not marked to be skipped, then it always appears.
---	--

Related Documentation

- [Configuring and Implementing Sign-in Notifications](#)

Configuring and Implementing Sign-in Notifications

Sign-in notifications appear for Pulse client and for browser-based logins when the user attempts to sign in.

To configure and implement sign-in notifications:

1. In the admin console, select **Authentication > Signing In > Sign-in Notifications**.
2. Click **New Notification**.
3. Specify a Name for the notification. This name appears in the sign-in policies page, and in the UI Options page for a selected role.
4. Select **Text** or **Package** in the Type box.
 - If you select **Text**, type the desired sign-in notification message, or copy and paste the relevant text into the Text field.
 - If you select **Package**, click the **Browse** button and navigate to a previously prepared .zip file. A package is typically used to provide different language versions of the notification message.
 - The zip file should include a default.txt file and one or more <language>.txt files (Example: en.txt).

- Language-abbreviations should be strings that can appear in Accept-Language header of an HTTP request.
- The character encoding supported is UTF-8.



Note: When you create a zip file, do not add the folder containing the files, but add the files directly.

5. Click **Save Changes**.

To enable sign-in notifications:

1. In the admin console, click **Authentication > Signing In > Sign-in Policies**.
2. Select an existing URL or create a new URL.
3. Under Configure Sign-in Notifications, select the check box for **Pre-Auth Sign-in Notification**, **Post-Auth Sign-in Notification**, or both.
 - After Pre-Auth Sign-in Notification, select a previously configured sign-in notification from the drop-down menu.
 - After Post-Auth Sign-in Notification, select the option for **Use a common Sign-in Notification for all roles** or **Use the Sign-in Notification associated to the assigned role**.
 - If you select **Use a common Sign-in Notification for all roles**, select a previously configured sign-in notification from the drop-down menu.
 - If you select **Use the Sign-in Notification associated to the assigned role**, the sign-in notification configured for the assigned role will be used.
 - Prevent the Post-Auth sign-in notification from being displayed to users who have seen it before, by selecting the **Skip if already shown** check box. (This is only a hint to the system and might not be honored in all environments.)
4. Click **Save Changes**.
5. You can customize the appearance of the sign-in notification message by selecting **Authentication > Signing In > Sign-in Pages** and creating a sign-in page or using an existing page.
6. Under Sign-in Notification appearance, customize UI options for Pre-Auth Notifications and Post-Auth Notifications by changing the following items:
 - For **Notification Title** enter the text that appears at the top of the sign-in notification page.
 - In the **Proceed Button** box, enter the text for the button that the user clicks to proceed with the sign-in.

This text applies to browser-based logins only. A Pulse client login always displays Proceed.
 - Optionally, clear the check box for **Display “Decline” Button**. If this box is not checked, the user does not have the option to decline.
 - In the **Decline Button** box, enter the text for the button that the user clicks to decline.

This text applies to browser-based logins only. A Pulse client login always displays Decline.
 - In the **Message on Decline** box, enter the text that you would like to appear when a user clicks the Decline button.
7. Click **Save Changes**.



Note: If you enabled Use the Sign-in Notification associated to the assigned role you must complete the implementation by selecting the sign-in notification on the Users > User Roles > Role Name > General > UI Options page or Administrators > Admin Roles > Role Name > General > UI Options page, as applicable.

If more than one role is available to a user, the sign-in notification associated with the first role assigned is displayed.

8. Add the sign-in page in which you have customized the sign-in notification appearance to the sign-in policy.

Related Documentation

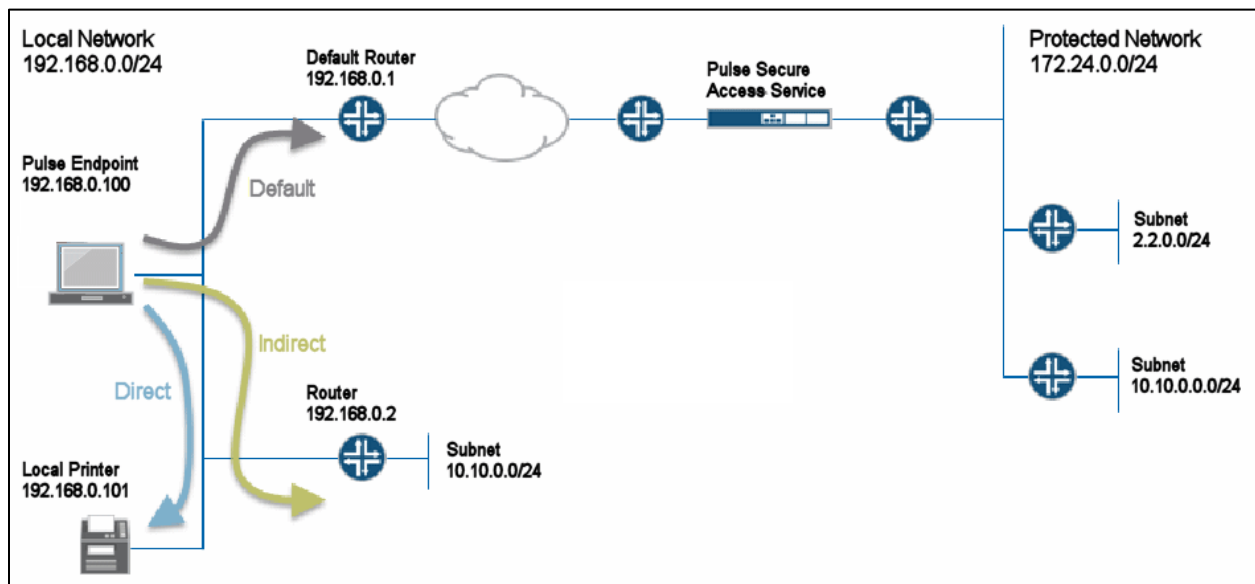
- [About Sign-In Notifications](#)

Pulse Connect Secure Split Tunneling Overview

The Pulse Secure clients for Windows, Apple OS X, Google Android, and Apple iOS all support split tunneling. Split tunneling is configured as part of the role that is assigned to a user after authentication. When the client and Pulse Connect Secure establish a VPN tunnel, the Pulse server takes control of the routing environment on the endpoint to ensure that only permitted network traffic is allowed access through the VPN tunnel. Split tunneling settings enable you to further define the VPN tunnel environment by permitting some traffic from the endpoint to reach the local network or another connected subnet. When split tunneling is enabled, split tunneling resource policies enable you to define the specific IP network resources that are excluded from access or accessible through the VPN tunnel.

Figure 51 shows a simple network configuration with three possible routes: through the default router, to the local subnet, or to a router connection to an indirectly connected subnet.

Figure 51: Pulse Split Tunneling



The network configuration in Figure 51 shows that the local network and the protected network at the other end of the VPN tunnel both have a subnet with the same private IP address, 10.10.0.0/24. In this case, the endpoint needs more information to determine where to send traffic addressed to that IP address range. You use the route precedence setting in the split tunneling settings to define which routing table takes precedence, either the tunnel routes (the routing table associated with the VPN tunnel) or the endpoint route (the routing table associated with the physical interface). If you select tunnel routes for route precedence, traffic addressed to network 10.10.0.0/24 in Figure 51 goes through the VPN tunnel and the 10.10.0.0/24 network available on the local indirect network is not reachable. If you select endpoint routes for route precedence, traffic addressed to network 10.10.0.0/24 goes through the physical adapter and the 10.10.0.0/24 network available through the VPN tunnel is not reachable. Pulse restores the original routes when the VPN tunnel is disconnected. However, no matter which way you define route precedence, the endpoint loses connectivity to one of the other of the networks if there are duplicate IP address networks.

Split Tunneling Disabled

When the endpoint has an active VPN tunnel connection, and split tunneling is disabled, the default route is modified to send all network traffic from the endpoint through the VPN tunnel where it is bound by the VPN access control and resource policies. If you set route precedence to endpoint routes, all network traffic goes through the VPN tunnel *except* traffic that is destined for directly-connected (local) subnets and indirectly connected (routed) subnets. The Pulse Secure clients for Windows and OS X also support the following option to permit limited access to the local network:

- If the Pulse connection set is configured to allow the user to override the connection policy, the user can manually suspend the active Pulse connection to enable access to the local network. In the network in [Figure 51](#) the user could suspend the Pulse connection to access the local printer, which resides on the same subnet as the Pulse endpoint. Suspending the Pulse connection is a manual method. The user must suspend the connection to access to local subnet and then resume the connection to restore connectivity through the tunnel. While the connection is suspended, no traffic goes through the tunnel.
- You can configure the split tunneling properties to allow access to the local subnet. With split tunneling disabled and local subnet access allowed, network traffic goes through the tunnel except for addresses that are on the local subnet. In the network in [Figure 51](#), the user could print to the local printer but other traffic would go through the default route to the tunnel. No traffic would go through the subnet router 192.168.0.2.

Split Tunneling Enabled

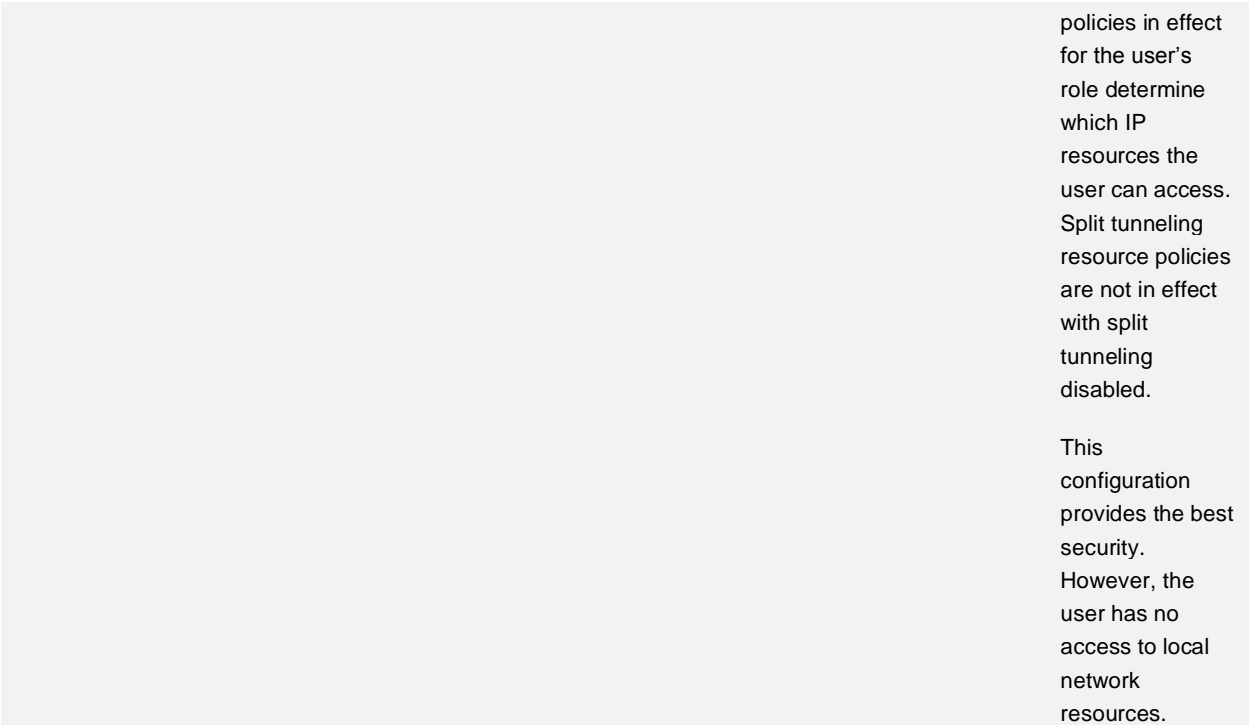
When the endpoint has an active VPN connection, and split tunneling is enabled for the role, the Pulse server adds or modifies routes on the endpoint so that traffic meant for specific subnets uses the VPN tunnel, and all other traffic goes through the local physical adapter. You specify the subnets that are excluded from access or accessible through the VPN tunnel by defining split tunneling resource policies. In a case where you have identically numbered subnets on both the local network and the tunnel network, (that is, the specified split-tunnel subnet conflicts with an existing endpoint route), the route precedence setting determines the traffic path. The route monitoring option, when enabled, enhances network security by terminating the VPN tunnel if another process on the endpoint makes a change to the routing table.

Pulse Split Tunneling Summary

Table 9 summarizes the traffic flows that are possible with each split tunnel configuration. Split tunneling options enable you to control the network traffic on the endpoint so that you can allow the needed connectivity to users while maintaining network security.

Table 9: Split Tunneling Options Summary

Split Tunnel Settings								Network Traffic Path
General	Web	Files	SAM	Telet/SSH	Terminal Services	Virtual Desktops	HTML5 Access	
<div>Options</div> <div>Pulse Secure Client Settings</div> <div>Split Tunneling</div> <div>Choose the split-tunneling mode</div> <div> <input checked="" type="radio"/> Enable <input type="radio"/> Disable </div> <div>▼ VPN client options</div> <div>Route Precedence</div> <div>Should the precedence be given to existing endpoint routes or tunnel routes in case of an overlap?</div> <div> <input checked="" type="radio"/> Tunnel routes (Applicable on Windows, MAC OSX, Linux) <input type="radio"/> Tunnel routes with local subnet access (Applicable with Pulse on Windows, MAC OSX) <input type="radio"/> Endpoint routes </div>								<p>All network traffic from the endpoint goes through the VPN tunnel. Local networks are not available. Pulse client users may choose to suspend the Pulse connection to allow local access if the Pulse connection set has the property Allow users to override connection policy enabled. VPN tunneling access control resource</p>



policies in effect for the user's role determine which IP resources the user can access. Split tunneling resource policies are not in effect with split tunneling disabled.

This configuration provides the best security. However, the user has no access to local network resources.

All network traffic goes through the VPN tunnel except traffic that is destined for directly-connected (local) subnets. VPN tunneling access control resource policies in effect for the user's role determine which IP resources the user can access. Split tunneling resource policies are not in effect with split tunneling disabled.

GeneralWebFilesSAMTelnet/SSHTerminal ServicesVirtual DesktopsHTML5 AccessMeetingsVPN Tunneling

OptionsPulse Secure Client Settings

Split Tunneling

Choose the split-tunneling mode

☒ Enable

☐ Disable

VPN client options

Route Precedence

Should the precedence be given to existing endpoint routes or tunnel routes in case of an overlap?

☐ Tunnel routes (Applicable on Windows, MAC OSX, Linux)

☒ Tunnel routes with local subnet access (Applicable with Pulse on Windows, MAC OSX)

☐ Endpoint routes

Network traffic addressed to the networks defined in the split tunnel resource policies goes through the VPN tunnel.

Network traffic that is addressed to the directly-connected (local) subnet goes to the local subnet. The default route is set to the local subnet so all other network traffic is subject to the original endpoint routing table.

GeneralWebFilesSAMTelnet/SSHTerminal ServicesVirtual DesktopsHTML5 AccessMeetingsVPN Tunneling

OptionsPulse Secure Client Settings

Split Tunneling

Choose the split-tunneling mode

☒ Enable

☐ Disable

VPN client options

Route Precedence

Should the precedence be given to existing endpoint routes or tunnel routes in case of an overlap?

☐ Tunnel routes (Applicable on Windows, MAC OSX, Linux)

☐ Tunnel routes with local subnet access (Applicable with Pulse on Windows, MAC OSX)

☒ Endpoint routes

Network traffic addressed to the networks defined in the split tunnel resource policies goes through the VPN tunnel.

The default route is set

to the local subnet so all other network traffic is subject to the original endpoint routing table.

This configuration provides the greatest flexibility for the user.

IPv6/IPv4 Split Tunneling

Pulse VPN now allows accessing both IPv4, IPv6 corporate resources from IPv4 and IPv6 endpoints and FQDN resources. It enables client to access both corporate network and local network at the same time. The network traffic designated are directed to tunnel interface for corporate network by configuring route policies, whereas other traffics are sent to direct interface.

Figure 52: IPv6 Split Tunneling

* Name:

Required: Label to reference this

Description:

▼ Resources

Specify the resources for which this policy applies, one per line.

IPv4 Resources:

Examples:

10.10.0.0/255.255.0.0

10.10.10.0/255.255.255.0

10.2.12.0/24

IPv6 Resources:

Examples:

[2001:db8:a0b:12f0::1]

[2001:DB8::6:0/112]

[2001:DB8::7:50]

FQDN Resources:

Examples:

www.example.com

*.example.com

Note: FQDN resources will be resolved to IPv4 addresses only.



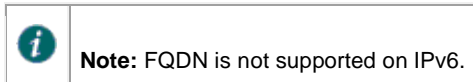
Note: All configurations to IPv6 are similar to IPv4.

Table 10: Split Tunneling Deployment Scenarios

Split Tunnel	IPv4 Tunnel Address	IPv6 Tunnel Address	IPv4 Include Policy	IPv4 Exclude Policy	IPv6 Include Policy	IPv6 Exclude Policy	Expected Client Behavior
Disabled	Yes	No	NA	NA	NA	NA	All IPv4 traffic should go through tunnel.
Disabled	Yes	Yes	NA	NA	NA	NA	Both IPv4 and IPv6 traffics should go through tunnel.
Enabled	Yes	No	IPv4 include subnet Eg: 10.0.2.0/24	IPv4 exclude subnet if any otherwise Empty.	Empty	Empty	All IPv4 include traffic should go through tunnel. All IPv4 exclude traffic should go directly through physical interface.
Enabled	Yes	Yes	IPv4 include subnet Eg: 10.0.2.0/24	IPv4 exclude subnet if any otherwise Empty.	Empty	Empty	All IPv4 include traffic should go through tunnel. All IPv4 exclude traffic should go directly through physical interface. All IPv6 traffic should go through the tunnel.
Enabled	Yes	Yes	Empty	Empty	IPv6 include subnet Eg: [fc00:0:0:2::/64]	IPv6 exclude subnet if any otherwise Empty	All IPv4 traffic should go through the tunnel. All IPv6 traffic included traffic should go through tunnel. All IPv6 exclude traffic should directly go through physical interface.
Enabled	Yes	Yes	IPv4 include subnet Eg: 10.0.2.0/24	IPv4 exclude subnet if any otherwise empty	IPv6 include subnet [fc00:0:0:2::/64]	IPv6 exclude subnet if any otherwise Empty	All IPv4 include traffic should go through tunnel. All IPv4 exclude traffic should go directly through physical interface. All IPv6 include traffic should go through tunnel. All IPv6 exclude traffic should go directly through physical interface.

Split Tunnel	IPv4 Tunnel Address	IPv6 Tunnel Address	IPv4 Include Policy	IPv4 Exclude Policy	IPv6 Include Policy	IPv6 Exclude Policy	Expected Client Behavior
Enabled	Yes	Yes	1 st policy: 10.0.2.0/24 2 nd policy: empty	Empty Empty	1 st policy: Empty 2 nd policy: [fc00:0:0:2::/64]	Empty Empty	All IPv4 traffic to 10.0.2.0/24 should go through tunnel. Other IPv4 traffic should directly go through physical adapter. All IPv6 traffic to [fc00:0:0:2::/64] should go through tunnel. Other IPv6 traffic should go to physical adapter.

For FQDN deployment scenarios, refer to FQDN based Split Tunneling Deployment Guide at <https://www.pulse.secure.net/>.



Split Tunneling Notes

- The Pulse server tries to resolve all DNS requests through the endpoint's physical adapter first, and then routes those that fail to the VPN tunneling adapter.

Configuring a Role for Pulse Connect Secure

A user role defines session settings and options, personalization settings (user interface customization and bookmarks), and access features (Web, file, application, Telnet/SSH, Terminal Services, network, meeting, and e-mail access). A user role does not specify resource access control or other resource-based options for an individual request. For example, a user role can define whether a user can perform Web browsing when the user is connected through the Pulse Connect Secure server Web portal. However, the individual Web resources that a user can access are defined by the Web resource policies that you configure separately.

The following procedure describes the role configuration options.

To create a role for Pulse Secure endpoints:

1. Select **Users > User Roles > New User Role** in the admin console.
2. Enter a name for the role and, optionally, a description. This name appears in the list of roles on the Roles page.
3. Under Client Options, select **Pulse Secure**.

When this option is enabled, the Pulse Secure button appears on the Connect Secure Web portal. When a user clicks it, Pulse is downloaded and installed on the user's endpoint.

Enabling this option alone does not enable Pulse as the client for the role. This option works in conjunction with the settings you enable in the Access Features section and then configure on the respective role tabs. The combination of settings determines whether you enable Pulse Secure client or Pulse Secure Application Manager (SAM). The following procedures describe how to enable each client option.

To enable Pulse Secure:

1. In the role's **General > Overview > Options** section select **Pulse Secure**.
2. This setting applies to both Windows and Apple OS X versions of Pulse Secure client.
3. In the Access Features section select **VPN Tunneling**.

The VPN Tunneling tab enables you to specify split tunnel behavior, specify the Pulse component set, and enable 3rd-party software integrations.

To enable Pulse Secure for SAM:

1. In the Options section select **Pulse Secure**.
2. In the Access Features section select **Secure Application Manager** and then select **Windows version**.

The SAM tab enables you to specify applications and servers secured by SAM.

4. Click Save Changes. Role configuration tabs appear.



Note: When the Pulse Secure option is enabled and no other access method (VPN Tunneling, WSAM) is enabled, then no client will be delivered.

Configuring General Role Options for Pulse Connect Secure

The General tab includes options for detailed control of how the client interacts with the server and the network. The following describes the options that apply to Pulse Secure.

General > Restrictions

Source IP—Control from which IP addresses users can access the Web portal sign-in page, be mapped to a role, or access a resource.

Browser—Allow or deny access to the role based on the browser's user agent string.

Certificate—Allow all users or only users with a signed client-side certificate.

Hot Checker—Select configured Host Checker policies to enforce with this role.

General > VLAN/Source IP

VLAN and Select Source IP—To direct traffic to specific sites based on the role, you can define a source IP alias for each role and then use the alias to configure virtual ports you define for the internal interface source IP address. A back-end device can then direct end user traffic based on the alias. This capability enables you to direct various end users to defined sites based on their roles, even though all of the end user traffic has the same internal interface source IP address.

General > Session Options

Idle Timeout—The maximum time a session can remain idle (no traffic) before the server ends the session.

Max. Session Length—The maximum time for a session before the server ends the session.

Reminder Time—When the Enable Session Extension feature is enabled, the Reminder Time specifies the number of minutes prior to a session end when the server sends a notice through Pulse and notifies the user that the session will end soon.

Enable Session Extension—Allows the user to extend the session. The user can choose to extend the session at any time by selecting a menu option in the Pulse client interface. If the Session Timeout Warning is selected, a notice message appears when the Reminder Time is reached and the user can choose to extend the session from within that notice message.

Enable Session Timeout Warning—Enables or disables the session timeout warning, which notifies the user when their Pulse VPN session is close to expiring. The Reminder Time value specifies the point at which the reminder appears.

Roaming Session—Select one of the following options to specify the client's roaming behavior:

- **Enabled**—A roaming session allows a user to retain connectivity when moving a device, such as a laptop with a dynamic IP address, from one subnet to another. Disable this feature to prevent users from accessing a previously established session from a new source IP address. Disabling roaming can help protect against an attack that spoofs a user's session.
- **Limit to Subnet**—Limit the roaming session to the local subnet specified in the endpoint's IP configuration. Users can sign in from one IP address and continue using their sessions with another IP address as long as the new IP address is within the same subnet.
- **Disabled**—Disable roaming user sessions for users mapped to this role.

Browser Session Cookie—Select **Enabled** to remove the Connect Secure session cookie and log users out of their Connect Secure web session after the Pulse client is launched. Removing the browser session cookie enhances Pulse session security.

General > UI Options

UI Options—The settings on this page define the Pulse Connect Secure Web portal page.

SAM > Applications

Add Application—We recommend that you use resource profiles to specify the applications available to users, but you can use role and resource policy settings instead.

SAM > Options

Auto-uninstall Secure Application Manager—This feature is not applicable to the Windows Phone client. Users must download and install Pulse for Windows Phone before the Windows Phone device can connect to the Pulse Connect Secure.

Prompt for username and password for intranet sites—If you enable this option, the Pulse Connect Secure requires users to enter sign-in credentials before connecting to sites on your internal network. This option changes Internet Explorer's intranet zone setting so that Internet Explorer always prompts the user for network sign-in credentials for an intranet site.

Auto-upgrade Secure Application Manager—This feature is not applicable to the Pulse for Windows Phone app.

Resolve only hostnames with domain suffixes in the device DNS domains—If you enable this option, users can only browse to Web sites that are part of their login domain.

Session start script and Session end script—This feature is not applicable to the Pulse for Windows Phone app.

Configuring Role Options for Host Checker for Pulse Connect Secure

Host Checker options allow you to enable configured Host Checker policies, to choose one or more policies for the role, and to specify whether the endpoint must meet all or just one of the selected Host Checker policies. Before you can assign Host Checker policies for a role, you must have already defined the policies.

To configure Host Checker for a selected role:

1. For a selected role, select **General > Restrictions > Host Checker**.
2. Select the check box **Allow users whose workstations meet the requirements specified by these Host Checker policies**.
3. Click Add to move **Host Checker policies** from the Available Policies list to the **Selected Policies list**.
4. Select the check box Allow access to the role... to grant access if the endpoint passes any of the selected **Host Checker policies**.
5. Click **Save Changes**.

Related Documentation

- [Configuring Pulse Secure Client for Secure Application Manager](#)
- [Machine Authentication for Pulse Connect Secure Overview](#)

Machine Authentication for Pulse Connect Secure Overview

Machine authentication uses machine credentials (machine name and password or machine certificate) to authenticate the endpoint. You can enable machine authentication for Pulse Connect Secure as part of a Pulse Secure Connection and distribute the connection to endpoints through the normal Pulse distribution methods. You enable Pulse machine authentication support on a Pulse connection, either Layer 2 or Layer 3.

The following describes the requirements for a machine authentication environment:

- The authentication server used by the Pulse connection must be Active Directory/Windows NT for machine name/password authentication or a certificate server for machine certificate authentication.
- The endpoint must be a member of a Windows domain, and the machine credentials must be defined in Active Directory.
- The Pulse connection must be configured so that no prompts are presented during the login process. For example, prompts for realm or role selection or for a server certificate trust prompt cause the connection to fail. You can specify a preferred role and realm for the connection, which eliminates realm and role selection dialogs.



Note: If the Pulse connection is configured to use a list of Pulse servers, the preferred roles and realms you specify must be applicable to all of those servers.

- For machine certificate authentication, the domain workstation login certificate must be issued by the domain certificate authority. The root certificate must be in the Machine Trusted Certificate store instead of the certificate store for a particular user.

Pulse supports the following machine authentication types:

- **machine-only**—The connection is established using machine credentials when no user is logged in. The connection is maintained after user login.
- **user-after-desktop**—The connection is established using machine credentials when no user is logged in. After user login, the machine connection is disconnected. Once the user logs out, the user connection is disconnected and the machine connection is reestablished.

Related Documentation

- [Configuring Machine-Only Machine Authentication for a Pulse Secure Connection](#)
- [Configuring User-After-Desktop Machine Authentication for a Pulse Secure Connection](#)
- [Credential Provider Authentication for Pulse Connect Secure Overview](#)
- [Machine and User Authentication Through a Pulse Connection for Pulse Policy Secure](#)
- [Creating a Client Connection Set for Pulse Connect Secure](#)
- [Configuring a Role for Pulse Connect Secure](#)

Credential Provider Authentication for Pulse Connect Secure Overview

When Microsoft introduced Windows Vista, it moved away from a login integration interface based on GINA (Graphical Identification and Authentication) in favor of credential provider authentication. The Pulse Secure client credential provider integration enables connectivity to a network that is required for the user to login to the Windows domain. For example, the domain controller might reside behind a firewall and the endpoint uses credential provider login to connect to the Pulse server prior to domain login. Pulse Secure client integrates with Microsoft credential providers to enable password-based login and smart card login. Credential provider login is supported on Windows 7 and later Windows platforms. You can use the Pulse support for credential provider to provide single sign-on capabilities. Pulse establishes a connection to the network and then uses the same credentials to login the Windows domain.

You enable Pulse credential provider support on a Pulse connection, (connection type UAC 802.1X (UAC) or SSL VPN (L3)). After the connection has been downloaded to the endpoint through the normal Pulse distribution methods, Pulse annotates the credential provider tile that appears on the user login screen by adding a Pulse icon in the lower right corner of the tile.

Pulse supports the following credential provider types:

- **user-at-credprov**—The connection is established before the user login using credentials collected at the selected credential tile, which provides single-sign-on functionality. The connection is maintained as an active connection on the user's desktop.
- **machine-then-user-at-credprov**—The connection is established using machine credentials when no user is logged in. When a user clicks a login tile and provides user credentials, the machine connection is disconnected and a new connection is established. When the user logs out, the user connection is disconnected and the machine connection is reestablished. In one typical machine-then-user-at-cred prov implementation, the machine connection and the user connection are mapped to different VLANs.

Pulse credential provider support usage notes:

- If the endpoint includes more than one Pulse Layer 2 connection, Windows determines which connection to use:
 1. If a network cable is attached to the endpoint, Layer 2 wired connections are attempted, and then wireless connections. If there are more than one wireless network available, the order is determined by the scan list specified as a Pulse connection option.
 2. After all Layer 2 options are attempted, Pulse runs location awareness rules to find one or more eligible Layer 3 connections that are configured for credential provider login. If more than one Layer 3 connection is found, Pulse prompts the user to select a connection. A user can cancel the network connection attempt by clicking the cancel button.
 3. After Pulse evaluates all configured connection options, Pulse returns control to Windows, which enables the user login operation.
- For connections that use user credentials, the Pulse connection can be configured so that prompts are presented during the login process, for example, prompts for realm or role selection or a server certificate trust prompt. For connections that use machine credentials, Pulse prompts cause the connection to fail because there is no interface to allow a response to the prompts. You can suppress any potential realm and role choice by specifying a preferred realm and role for the connection.
- Pulse upgrade notifications and actions are disabled during credential provider login and postponed until the user connection is established. Host Checker remediation notifications are displayed.

Related Documentation

- [Configuring User-at-Credprov Credential Provider Authentication for a Pulse Secure Connection](#)
- [Configuring Machine-Then-User-at-Credprov Credential Provider Authentication for a Pulse Secure Connection](#)

Configuring Role Options for Pulse Connect Secure

All of the options for role configuration tabs are described in *User Access Management Framework Feature Guide*.

To configure role options for Pulse Secure endpoints:

1. From the admin console, select **Users > User Roles**.
2. Click the role you want to configure and then click the VPN Tunneling tab.
3. Under Split Tunneling Options, select your options:

General VPN Options apply to all Layer 3 VPN clients, Pulse Secure client (Windows, OS X, iOS, and Android), , and third-party IKEv2 clients:

- **Split Tunneling**—Split tunneling options let you define how network traffic flows on the client.

Enable— Pulse modifies routes on the client so that traffic meant for the corporate intranet uses the virtual adapter created by Pulse (the Pulse tunnel) and all other traffic goes through the local physical adapter.

Disable—When the Pulse session is established, predefined local subnet and host-to-host routes that might cause split-tunneling behavior are removed, and all network traffic from the client goes through the Pulse tunnel. With split tunneling disabled, users cannot access local LAN resources during an active VPN session. However, when you create a Pulse connection for users, if you allow users to override you can enable an option that allows the user to suspend their VPN connection. While a Pulse connection is in the suspended state, all traffic goes through the local physical adapter.

Pulse Secure client Options apply to Pulse Secure client:

- **Route Precedence**—You can define which routing table takes precedence:

Tunnel Routes—The route table associated with the Pulse virtual adapter take precedence. Pulse overwrites the physical interface routes if there is conflict between the Pulse virtual adapter and the physical adapters. Pulse restores the original routes when the connection is ended.

Tunnel Routes with local subnet access (Pulse on Windows and macOS only)—Network traffic addressed to the networks defined in the split tunnel resource policies goes through the VPN tunnel. Network traffic that is addressed to the directly-connected (local) subnet goes to the local subnet. The default route is set to the local subnet so all other network traffic is subject to the original endpoint routing table.

Endpoint Routes—The route table associated with the endpoint's physical adapter take precedence.

- **Route Monitor**—Pulse can monitor the route tables and take appropriate action.

Yes – VPN tunneling ends the connection only if the route change affects the VPN tunnel traffic. For example, if the route metric is changed higher, it should not disconnect VPN tunneling.

No – Route tables are allowed to change on the client endpoint.

- **Traffic Enforcement**—When Traffic Enforcement is enabled, Pulse creates rules on the endpoint's firewall (macOS and Windows) that ensure that all traffic conforms to the PCS tunnel configuration. A local program might bypass the endpoint's routing tables and bind traffic to the physical interface instead of allowing it to go through the Pulse virtual interface. If you enable traffic enforcement, you ensure that all traffic is bound by the PCS tunnel configuration.

IPv4 – When this check box is selected all IPv4 traffic is bound by the PCS tunnel configuration.

IPv6 – When this check box is selected all IPv6 traffic is bound by the PCS tunnel configuration.

Typically, if you wanted to enable traffic enforcement, you would enable both options. Enabling only one option is useful for nested tunnel (tunnel-in-tunnel) configurations. See the [Pulse Secure Supported Platforms Guide](#) for supported nested tunnel configurations.

- **Enable TOS Bits Copy**—Enables you to control the client behavior in networks that employ Quality of Service (QoS) protocols. When you enable this check box, the Pulse Secure client copies IP Type of Service (TOS) bits from the inner IP header to outer the IP Header. Note that enabling this option might require a reboot of the client

endpoint when the client software is installed for the first time on Windows endpoints. Pulse Secure clients support TOS bit copy only for IPSec transport and not for SSL transport.

- **Multicast**—Enables the multicast feature on the client when this option is selected.
- **Auto-launch**—Activates the Pulse Secure client software automatically when the endpoint is started when this option is selected.

Figure 53: VPN Client Options

VPN client options

Route Precedence
Should the precedence be given to existing endpoint routes or tunnel routes in case of an overlap?

- ☐ Tunnel routes (Applicable on Windows, MAC OSX, Linux)
- ☐ Tunnel routes with local subnet access (Applicable with Pulse on Windows, MAC OSX)
- ☒ Endpoint routes

Route Monitor
Should VPN client disconnect when route changes that affect tunneled traffic are made?

- ☐ Yes
- ☒ No

Traffic Enforcement (Applicable with Pulse on Windows, MAC OSX)
Should traffic policies be enforced on the client?

- ☐ IPv4
- ☐ IPv6

☐ **Enable TOS Bits Copy**
VPN client will copy IP TOS bits from inner IP header to outer IP Header. This option is useful in situations where network between the client and IVE has QoS capabilities. Note that enabling this option may require reboot when client is installed for the first time on Windows platform. VPN client supports TOS bits copy only when it is using IPSec transport and not for SSL transport. This option is not supported on Pulse mobile.

☐ **Multicast**
VPN client will enable Multicast feature

☐ **Auto-launch**
Use auto-launch to automatically start the client application when users sign in

Options for Pulse Secure client on Windows apply to Pulse Secure client on Windows endpoints:

- **Launch client during Windows Interactive User Logon**—When this option is enabled, the Pulse Secure client starts when the user logs into Windows. Note that this setting is not the same as the Pulse connection settings that control machine authentication and credential provider authentication. Choose one of the following options:

Require client to start when logging into Windows

Allow user to decide whether to start client when logging into Windows

- **Windows: Session start script**—Specify a script (.bat, .cmd, or .exe) to run for users assigned to the role after Pulse Secure client connects with Pulse Connect Secure. For example, you can specify a script that maps network drives on an endpoint to shares on protected resources.
- **Windows: Session end script**—Specify a script (.bat, .cmd, or .exe) to run for users assigned to the role after Pulse Secure client disconnects from Pulse Connect Secure. For example, you can specify a script that disconnects mapped network drives. If there is no start script defined, or the start script has not been run, the end script does not run.
- **Skip if Windows Interactive User Logon Enabled**—Select this option to bypass the specified Windows session start script.
- If the client signs in to their Windows Domain via the Credential Provider automatic sign-in function, a script is executed by the Windows client. In this case, the sign-in script might be identical to the specified VPN Tunneling start script. You can use this option, therefore, as a way to avoid executing the same script twice.

Options for Pulse Secure client on Mac apply to Pulse Secure client on Apple OS X endpoints:

- **Mac: Session start script**—Specify a script (.bat, .cmd, or .exe) to run for users assigned to the role after Pulse Secure client connects with Pulse Connect Secure. For example, you can specify a script that maps network drives on an endpoint to shares on protected resources.
 - **Mac: Session end script**—Specify a script (.bat, .cmd, or .exe) to run for users assigned to the role after Pulse Secure client disconnects from Pulse Connect Secure. For example, you can specify a script that disconnects mapped network drives. If there is no start script defined, or the start script has not been run, the end script does not run.
4. In the **Session scripts** area, optionally specify a location for the following:
 - **Windows: Session start script**—Specify a script (.bat, .cmd, or .exe) to run for users assigned to the role after Pulse connects with Pulse Policy Secure. For example, you can specify a script that maps network drives on an endpoint to shares on protected resources. The script must be in a location (either local or on the network) that is accessible by the user.
 - **Windows: Session end script**—Specify a script (.bat, .cmd, or .exe) to run for users assigned to the role after Pulse Secure client disconnects from Pulse Policy Secure. For example, you can specify a script that disconnects mapped network drives. If there is no start script defined, or the start script has not been run, the end script does not run. The script must be in a location (either local or on the network) that is accessible by the user.
 5. Click **Save Changes**.

Configuring User-at-Credprov Credential Provider Authentication for a Pulse Secure Connection

With a user-at-credprov connection, the Pulse connection establishes the connection before user login using credentials collected at the selected credential tile, which provides single sign-on functionality. The connection is maintained as an active connection on the user's desktop.

To enable user-at-credprov credential provider support for a Pulse connection:

1. Create a Pulse connection set for the role (Users > Pulse Secure > Connections), and then create a new Pulse connection. You can select either a Layer 3 connection type, Connect Secure or Policy Secure (L3), or a Layer 3 connection type, UAC (802.1X).
2. In the Connection is established section, select User for the mode.
3. Under Options, select the Connect automatically and the Enable pre-desktop login (Credential provider) check boxes.

Figure 54: Connect automatically at user login

▼ Connection is established:

Specify mode: User ▼

Options:

☒ Connect automatically

☒ Enable pre-desktop login (Credential provider)

4. For a Layer 2 connection that uses machine certificate authentication, make sure that the connection has an entry in the Trusted Server List. To allow any server certificate, type **ANY** as the Server certificate DN. To allow only one server certificate, specify the server certificate's full DN, for example, C=US; ST=NH; L=Kingston; O=My Company; OU=Engineering; CN=c4k1.stnh.mycompany.net; E=ausername@mycompany.com.
5. Specify **Realm and Role Preferences** to suppress realm or role selection dialogs during the login process:
 - **Preferred User Realm**—Specify the realm for this connection. The connection ignores any other realm that is available for the specific login credentials.

The following options enable you to allow the user to login using a smart card or a password:

- **Preferred Smartcard Logon Realm**—Preferred realm to be used when user logs in with a smart card.
- **Preferred Password Logon Realm**—Preferred realm to be used when user logs in with a password.



Note: Be sure that the authentication realms you specify exist, and that they support the appropriate login credential option.

- **Preferred User Role Set**—Specify the preferred role or the name of the rule for the role set to be used for user authentication. The role or rule name must be a member of the preferred user realm.



Note: If the Pulse connection is configured to use a list of Pulse servers, the preferred roles and realms you specify must be applicable to all of those servers.

Related Documentation

- [Credential Provider Authentication for Pulse Policy Secure Overview](#)
- [Configuring a Pulse Credential Provider Connection for Password or Smart Card Login](#)

Configuring Machine-Then-User-at-Credprov Credential Provider Authentication for a Pulse Secure Connection

With a machine-then-user-at-credprov connection, Pulse establishes the connection using machine credentials when no user is logged in. When a user clicks a login tile and provides user credentials, the machine connection is disconnected, and a new connection is established. When the user logs out, the user connection is disconnected, and the machine connection is reestablished. In one typical machine-then-user-at-credprov implementation, the machine connection and the user connection are mapped to different VLANs.

To enable machine-then-user-at-credprov credential provider support for a Pulse connection:

1. Create a Pulse connection set for the role (Users > Pulse Secure > Connections), and then create a new Pulse connection. You can select either a Layer 3 connection type, Connect Secure or Policy Secure (L3), or a Layer 2 connection type, Policy Secure (802.1X).
2. In the Connection is established section, select **User or Machine** for the mode.
3. Under Options, select the **Connect automatically** check box.

Figure 55: Connect automatically when the machine starts. Connection is authenticated again at user login

✔ **Connection is established:**

Specify mode: Machine or User ▼

Options:

☒ **Connect automatically**

☐ **Enable pre-desktop login (Credential provider)**

4. In the Connection is established section, select one of the following options:
5. For a Layer 2 connection that uses machine certificate authentication, make sure that the connection has an entry in the Trusted Server List. To allow any server certificate, type **ANY** as the Server certificate DN. To allow only one server certificate, specify the server certificate's full DN, for example, C=US; ST=NH; L=Kingston; O=My Company; OU=Engineering; CN=c4k1.stnh.mycompany.net; E=username@mycompany.com.

6. Specify **Realm and Role Preferences** to suppress realm or role selection dialogs during the login process for both machine and user logins:
- **Preferred Machine Realm**—Specify the realm that this connection uses when establishing the machine connection. The connection ignores any other realm that is available for the specific login credentials.
 - **Preferred Machine Role Set**—Specify the role or the name of the rule for the role set that this connection uses when establishing the machine connection. The role or rule name used must be a member of the preferred machine realm.
 - **Preferred User Realm**—Specify the realm that for this connection that is used when a user logs in to the endpoint. The connection ignores any other realm that is available for the user's login credentials.

The following options enable you to allow the user to log in using a smart card or a password:

- **Preferred Smartcard Logon Realm**—Preferred realm to be used when user logs in with a smart card.
- **Preferred Password Logon Realm**—Preferred realm to be used when user logs in with a password.



Note: Be sure that the authentication realms you specify exist, and that they support the appropriate login credential option.

- **Preferred User Role Set**—Specify the preferred role or the name of rule for the role set to be used for user authentication. The role or rule name used must be a member of the preferred user realm.



Note: If the Pulse connection is configured to use a list of Pulse servers, the preferred roles and realms you specify must be applicable to all of those servers.

7. Optionally, specify pre-login preferences:
- **Pre-login maximum delay**—The time period (in seconds) that a Windows client waits for an 802.1X connection to succeed during the login attempt. The range is 1 to 120 seconds.
 - **Pre-login user based virtual LAN**—If you are using VLANs for the machine login, you can enable this check box to allow the system to make the VLAN change.
8. Click **Save Changes**, and then distribute the Pulse connection to Pulse client endpoints.

Related Documentation

- [Credential Provider Authentication for Pulse Policy Secure Overview](#)
- [Machine and User Authentication Through a Pulse Connection for Pulse Policy Secure](#)
- [Configuring a Pulse Credential Provider Connection for Password or Smart Card Login](#)

Machine and User Authentication through a Pulse Connection for Pulse Connect Secure

Pulse Secure client supports certificate authentication for establishing Layer 2 and Layer 3 connections. On Windows endpoints, the Pulse client connection accesses client certificates located in the Local Computer personal certificate store to provide machine authentication or user certificates located in a user's personal certificate store or a smart card for user authentication. A Pulse connection can access certificates from only one location. For information on machine authentication, see "[Machine Authentication for Pulse Connect Secure Overview](#)".

You can create a Pulse connection that verifies the identity of both the machine and the user before establishing a connection. There are two options for configuring this dual authentication connection. Both options employ user authentication against a Local System, Active Directory, or ACE server for user authentication and certificate authentication to verify the machine. Both options also use a Pulse connection option. The option, Select client certificate from machine certificate store, is part of the User Connection Preferences of a Pulse connection.

Option 1: Use an additional authentication server for a realm:

- Create a Pulse connection for the target Pulse server. The connection type can be Policy Secure (802.1X) or Connect Secure or Policy Secure (L3). The Connection is established option is typically set to manually by the user or automatically at user login.
- In the User Connection Preferences section of the connection properties, click the check box labeled Select client certificate from machine certificate store. This option enables the Pulse connection to perform the machine authentication as part of the Pulse connection attempt.
- Create a realm sign in policy that authenticates to a certificate server. When Pulse provides the certificate to the server, it uses the certificate from the Local Computer certificate store, which authenticates the machine. If the certificate store holds more than one valid certificate for the connection, Pulse opens a dialog box that prompts the user to select a certificate.
- Create a secondary authentication server for the realm. The secondary server can be a Local System, Active Directory, or RSA ACE server. When the machine authentication is successful, the user is prompted to provide authentication credentials for the secondary authentication server.

Option 2 — Use realm authentication to authenticate the user and a certificate restriction on the realm to authenticate the machine.

- Create a Pulse connection for the target Pulse server. The connection type can be Policy Secure (802.1X) or Connect Secure or Policy Secure (L3). The Connection is established option is typically set to manually by the user or automatically at user login.
- In the User Connection Preferences section of the connection properties, click the check box labeled Select client certificate from machine certificate store.
- Create a sign-in policy on the Pulse server that specifies a user realm. The realm authentication server can be a System Local, Active Directory, or RSA ACE server.
- Configure a certificate restriction on the realm to enable the Pulse server to request a client certificate. Be sure to enable the option labeled only allow users with a client-side certificate signed by Trusted Client CAs to sign in.

Related Documentation

- [Using Device Certificates](#)

Stealth Mode

Stealth mode is the robust solution to provide a seamless authentication to the user without any user interaction when transitioning from one connection to another. This feature supports only on Windows.

Stealth mode is the robust solution to provide a seamless Step-up, Step-down experience to the end-user when transitioning from one connection to another.

Now, while [Configuring Pulse Connect Secure](#) settings, the following two new checkboxes are added under connection set.

- **Enable stealth mode on this connection**
- **Show stealth connection to user**

When **Enable stealth mode on this connection** is enabled, user will not be able to see and control the established connection through the Pulse Client UI. User or machine authentication will happen seamlessly without any user interaction.

When **Show stealth connection to user** is enabled, user will be able to see the Stealth mode connection in Pulse UI. User will be able to see only the connection status in Pulse Tray icon and an option to view Advanced Connection details. User will not be able to control any actions.


Admin can enable the checkbox **Show stealth connection to user** only when **Enable Stealth mode on this connection** checkbox is checked.

For example, admin wants to configure two connections one by stealth and another connection as non-stealth.

One is stealth enabled connection named **9.0R3_Feature** (by enabling **Enable Stealth mode on this connection**), consider it as Step-down connection.

Second connection is non-stealth configured connection named **Step-Up**. Refer the following figure:

Figure 56: Connections

	Name	Type	Description
<input type="checkbox"/>	1. 9.0R3_Feature	Connect Secure or Policy Secure (L3)	
<input type="checkbox"/>	2. Step-Up	Connect Secure or Policy Secure (L3)	

Following are the two scenarios to understand the stealth mode behavior.

Scenario - I


Enable Stealth mode on this connection: Enabled

Show stealth connection to user: Disabled

User will not be able to see configured Step-down (Stealth mode connection - 9.0R3_Feature) on Pulse UI. Refer [Figure 57](#) and [Figure 58](#).

Figure 57: Stealth Mode Enabled

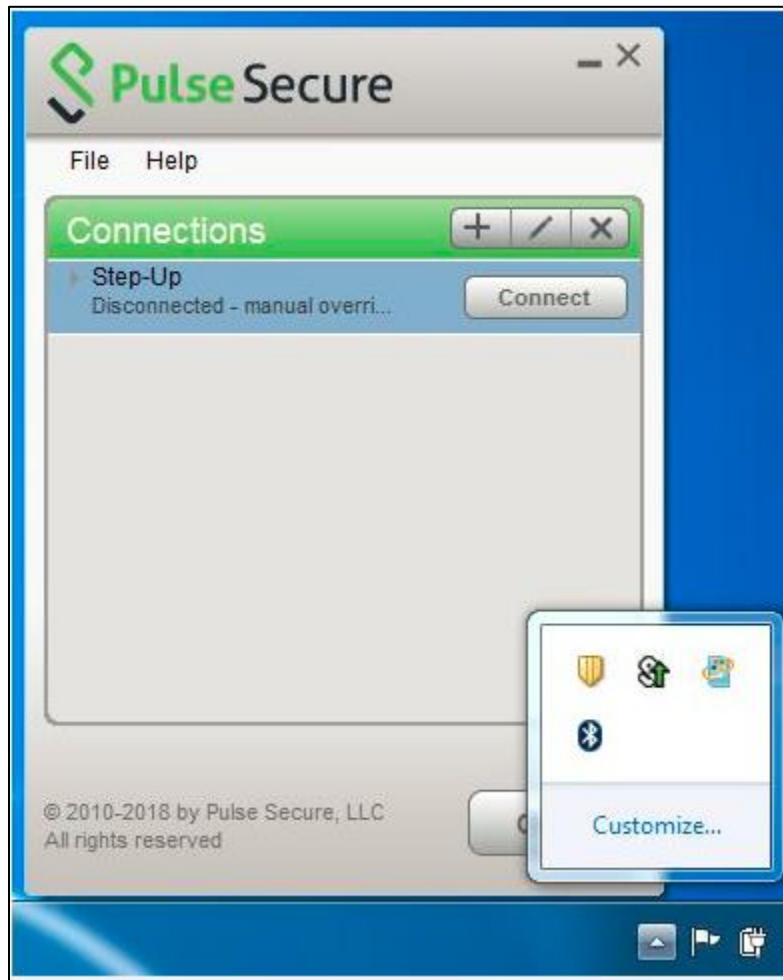
Type: Connect Secure or Policy Secure (L3)

 Options:

Name	Value
Allow user to override connection policy <small>Allows user to modify connection state.</small>	<input type="checkbox"/>
Lock down this connection <small>Network access is limited until this connection is established. This option is available only when the Always-on Pulse Client option or VPN only access option on the connection set is checked.</small>	<input type="checkbox"/>
Enable stealth mode on this connection <small>User will not be able to see and control the established connection through the Pulse client window. Under stealth mode, user or machine authentication happens seamlessly without any user interaction.</small>	<input checked="" type="checkbox"/>
Show stealth connection to users <small>When enabled, the end user can see the stealth mode connection in the Pulse client window. End user will only see the connection status and an option to view the Advanced connection details and will not have any other actionable controls. This can be used for troubleshooting purposes.</small>	<input type="checkbox"/>
Support Remote Access (Connect Secure) or LAN Access (Policy Secure) on this connection <small>Uncheck only if the connection is not used for Connect Secure or Policy Secure services (e.g Server is used for Pulse Collaboration only).</small>	<input checked="" type="checkbox"/>

Now, Step-down (Stealth enabled connection - 9.0R3_Feature) is set, but not visible to the user on Pulse UI. User can only see the connection status in Pulse tray icon. Refer the following figure:

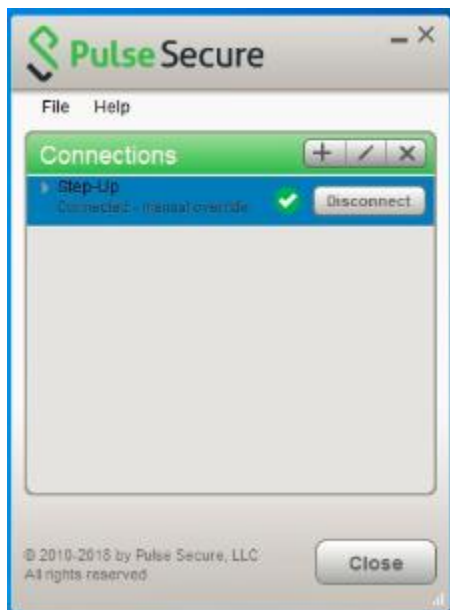
Figure 58: Show Stealth Mode Disabled



When user clicks **Connect** button of Step-up connection, Step-down (Stealth – 9.0R3 feature) gets disconnected and when user clicks **Disconnect** button to disconnect Step-up connection, step-down automatically gets connected.

Refer the following figure:

Figure 59: Step-up connection - Connected



Step-Up connections can get terminated in many scenarios for example:

- When user disconnects Step-Up connection
- Session Timeout (if user does not enter credentials once the timeout happens)
- Location Awareness becomes False

Scenario - II

Enable Stealth mode on this connection: Enabled

Show stealth connection to user: Enabled

User will be able to see Step-down (Stealth enabled connection- 9.0R3_Feature) on Pulse UI.

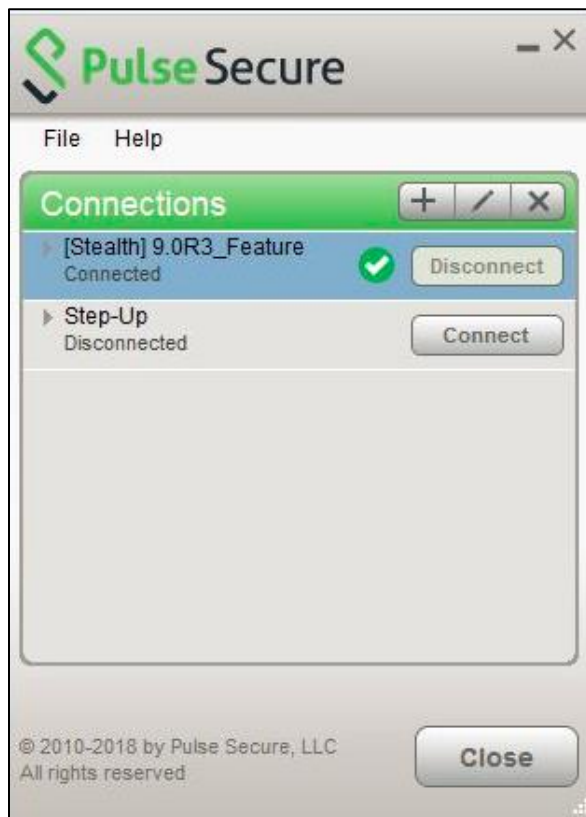
Refer the following figure:

Figure 60: Show stealth connection to users Enabled

Name	Value
Allow user to override connection policy Allows user to modify connection state.	<input type="checkbox"/>
Lock down this connection Network access is limited until this connection is established. This option is available only when the Always-on Pulse Client option or VPN only access option on the connection set is checked.	<input type="checkbox"/>
Enable stealth mode on this connection User will not be able to see and control the established connection through the Pulse client window. Under stealth mode, user or machine authentication happens seamlessly without any user interaction.	<input checked="" type="checkbox"/>
Show stealth connection to users When enabled, the end user can see the stealth mode connection in the Pulse client window. End user will only see the connection status and an option to view the Advanced connection details and will not have any other actionable controls. This can be used for troubleshooting purposes.	<input checked="" type="checkbox"/>
Support Remote Access (Connect Secure) or LAN Access (Policy Secure) on this connection Uncheck only if the connection is not used for Connect Secure or Policy Secure services (e.g Server is used for Pulse Collaboration only).	<input checked="" type="checkbox"/>
Enable Pulse Collaboration integration on this connection Applicable for Connect Secure type connections only. Leave this unchecked for Policy Secure type connections.	<input type="checkbox"/>
Connect to URL of this server only Connection is only made to the server which supplied configuration.	<input type="checkbox"/>

Now, Step-down (Stealth enabled connection - 9.0R3_Feature) connection is set and will be visible to the user on Pulse UI. Refer the following figure:

Figure 61: Show Stealth Mode Enabled



When user clicks **Connect** button of Step-up connection, Step-down (Stealth – 9.0R3 feature) gets disconnected and when

user clicks **Disconnect** button to disconnect Step-up connection, step-down connection automatically gets connected. Refer following figures [Figure 62](#) and [Figure 63](#)).

Figure 62: Step-Up connection - Connected

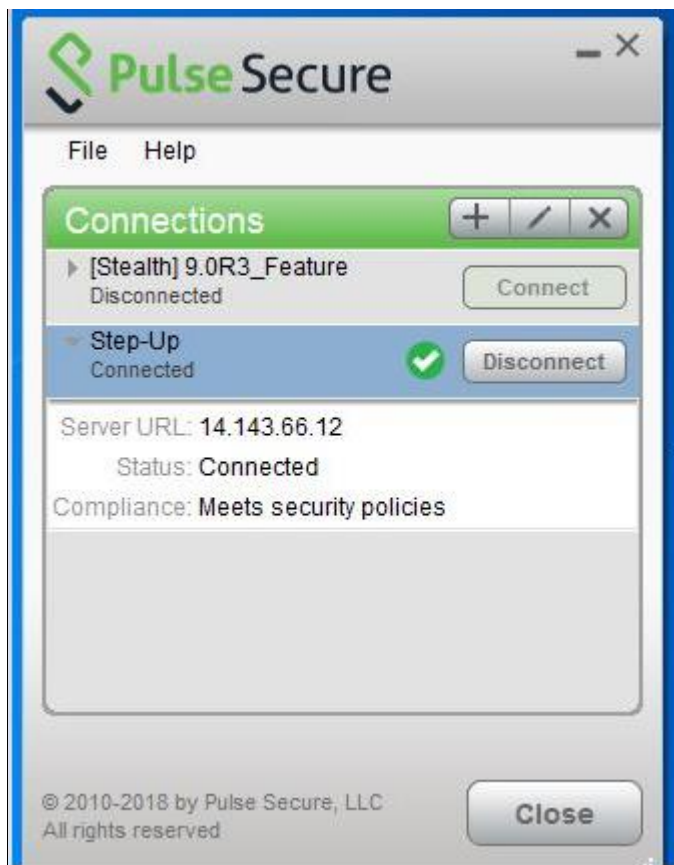
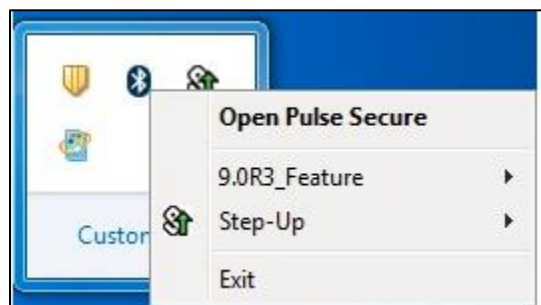


Figure 63: Step-Up connection – Pulse Tray icon

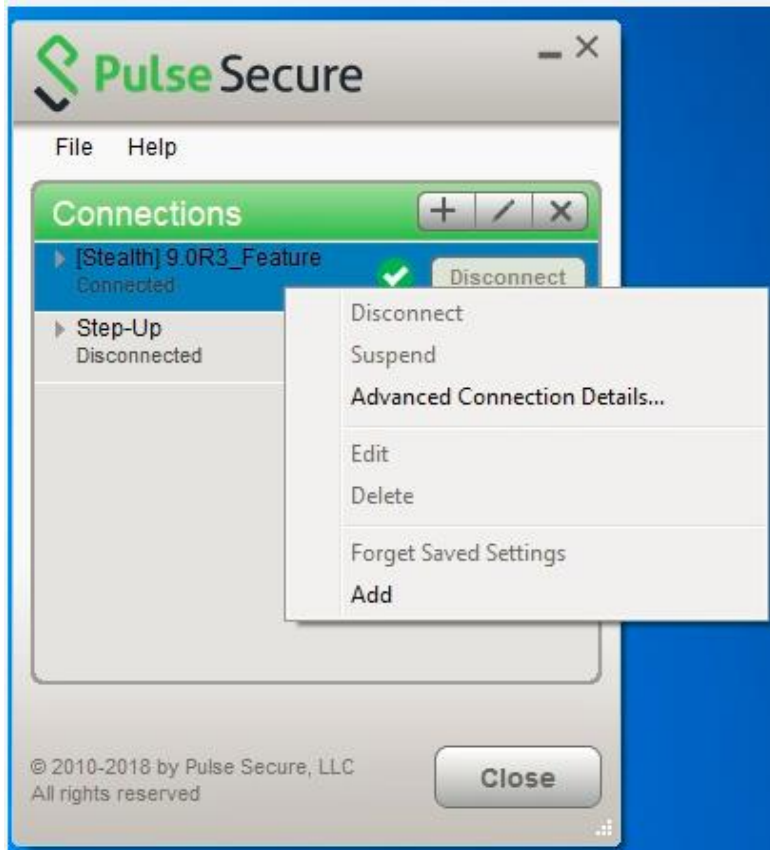


User will not be able to perform the actions like Disconnect, Suspend, Cancel, Edit, Delete, Forget Saved Settings. User will be able to see Advanced Connection Details and Add another connection. Refer [Figure: Stealth Mode Connection – Actions](#)



Note: When user clicks on Add to add another connection, it will not be in Stealth Mode.

Figure 64: Stealth Mode Connection - Actions



Stealth mode can be enabled for the following types of connections:

- User connection
- Machine connection
- User or Machine Connection

User would not know that a tunnel is established. Authentication could be done through AD username/password, Cert-based, Smart-card.

The following connection settings would be non-editable by the user when the Stealth mode is enabled on a connection.

- Allow user to override connection policy: Disabled

Name	Value
Allow user to override connection policy <small>Allows user to modify connection state.</small>	<input type="checkbox"/>

- Use Desktop Credentials: Enabled

Use Desktop Credentials <small>If checked, then the system login credentials will be cached and used for this connection. If credential provider is enabled, then the cached credentials will come from credential provider; otherwise, the credentials will come from the previous authentication on any connection that has this property checked.</small>	<input checked="" type="checkbox"/>
--	-------------------------------------

- Connect automatically: Enabled
- Reconnect at Session Timeout or Deletion: Enabled

✔ Connection is established:

Specify mode: ▼

Options:

☒ Connect automatically

☒ Reconnect at Session Timeout or Deletion

☐ Enable pre-desktop login (Credential provider)

Note: [L3 and Pulse SAM coexistence feature](#) will not work, if L3 is configured in Stealth Mode connection.

Configuring Pulse Secure Client for Secure Application Manager

Pulse Secure client supports Secure Application Manager (SAM). SAM provides remote access using application names and destinations. SAM does not require a virtual adapter or virtual IP address on the endpoint. SAM provides secure access to client/server applications and thin client solutions without provisioning a VPN tunnel.

With Pulse R3.0 and later, SAM connectivity is provided through SSL VPN (Pulse connection type Connect Secure or Policy Secure (L3)). Prior to Pulse R3.0, SAM connectivity was provided through a separate client. Table 11 describes the progression of Pulse/SAM client software.

Table 11: Pulse/SAM Client Version Summary

Pulse/SAM Version	Supported Platforms	Description	Notes
Pulse R1.0	Windows Mobile	SAM client that is installed from the Pulse server.	Supports Host Checker.
Included with SSL/VPN software R7.0 and R7.1	Windows XP		
	Windows Vista		
	Windows 7		

Pulse/SAM Version	Supported Platforms	Description	Notes
Pulse R2.0	Windows Mobile (6.0, 6.1, and 6.5) Pulse Secure R2.0 is supported on touch-based Windows Mobile devices only.	Pulse for Windows Mobile smart phone app; available for download from http://www.pulsesecure.net/support .	<p>If you install the Pulse 2.0 mobile client on a Windows Mobile device that already has Pulse R1.0, the installation detects the presence of the old client and removes it prior to installing the new client. It also detects and removes Host Checker. Host Checker is not supported.</p> <p>If Pulse R2.0 for Windows Mobile is installed on a Windows Mobile device, the user should not use a browser to sign into a realm that has Pulse R1.0 enabled. Pulse R1.0 cannot detect if Pulse R2.0 for Windows Mobile is already installed, and so it prompts the user to install Pulse R1.0.</p> <p>NOTE: If Pulse R2.0 is installed on a Windows Mobile device, and the user connects to a role that has Host Checker enabled, the user is prompted to install Host Checker. However, if the user allows the installation, nothing happens. To avoid this scenario, you should create a separate role for Pulse R2.0 for Windows Mobile devices.</p>
Pulse R3.0 and 4.0	Windows XP Windows Vista Included with Pulse Secure Access Service software R7.2 and later.	Pulse incorporates SAM functionality as a native Pulse connection method.	Supports Host Checker.
Pulse R5.0	Windows XP Windows Vista Windows 7 Windows 8		
Pulse R5.1 and later	Windows 7 Windows 8		

This section describes how to configure Pulse Connect Secure to support Windows endpoints. Pulse Connect Secure also supports a Java-based SAM client (JSAM). The JSAM client can be deployed from a Pulse Connect Secure server to any endpoint that supports Java.

To enable SAM for Windows endpoints and configure a role:

1. Log in to the Pulse Connect Secure admin console.
2. Select **User Roles > New User Role**.
3. On the New Role page, specify a name for the role and, optionally, a description. Make note of the name because later in this procedure, you create a realm and map realm users to this role.
4. In the Options section, select **Pulse Secure**.



Note: If you leave the Pulse Secure check box cleared, and then enable Secure Application Manager, Windows version in the Access Features section, you enable the Pulse/SAM for the Pulse for Windows Mobile smart phone app. The Pulse Secure check box must be selected to enable the role for Pulse for Windows endpoints.

5. In the Access Features section of the New Role page, select the **Secure Application Manager** check box and then select **Windows version**.
6. Click **Save Changes** to create the role and to display the role configuration tabs.

The General tab options (Restrictions (which includes Host Checker), VLAN/Source IP, Session Options, and UI Options) are all valid settings for a SAM role.

We recommend that you use resource profiles to specify the applications available to users, but you can use role settings instead.

To specify applications for SAM to secure as part of a role:

1. Open the role you created for Pulse/SAM.
2. Click the SAM tab.
3. In the Applications section, click **Add Application** or select an existing application in the list and then click **Add Duplicate**.
4. In the Details section, select a type from the Type list, and then specify a name and description.

If you select **Custom** to specify an application that is not included in the list, the Application Parameters section appears. Specify the following:

- **Filename**—Specify the name of the file's executable file
- **Path**—Specify the file's path
- **MD5 Hash**—Optionally specify the MD5 hash of the executable file. If you enter an MD5 hash value, Pulse verifies that the checksum value of the executable matches this value. If the values do not match, Pulse notifies the user that the identity of the application could not be verified and does not allow access.

If you select **Pick a Resource Profile**, and at least one application or destination has been configured as a Resource Profile SAM client application, a selection list appears and you can click a Resource Profile. Then, when you click **Save Application** or **Save + New**, the role is added to the profile's list of roles, and the profile's resource policies are updated. If there are no Resource Profile SAM client applications or destinations configured, this option is not available.

5. Click **Save Application** or **Save + New**.

To specify servers for SAM to secure as part of a role:

1. Open the role you created for Pulse/SAM.
2. Click the SAM tab.
3. In the Applications section, click **Add Server** or select an existing server in the list and then click **Add Duplicate**.

If you select **Standard**, specify a name and a description, and then identify the server by name or IP address.

If you select **Pick a Resource Profile**, a selection list appears and you can click a Resource Profile. Then, when you click **Save Application** or **Save + New**, the role is added to the profile's list of roles, and the profile's resource policies are updated.

4. Click **Save Application** or **Save + New**.

To specify options for the SAM role:

1. Open the role you created for Pulse/SAM.

2. Click the SAM tab.
3. Click **Options**.
4. Make sure **Windows SAM** is enabled, and then choose from the following:
 - Secure Application Manager options:
 - Auto-launch Secure Application Manager—If you enable this option, Pulse Connect Secure automatically launches Secure Application Manager services when a user signs in through the Connect Secure Web portal. If you do not select this option, users must manually start the Secure Application Manager from the Client Applications Sessions section of the Web portal.
 - Auto-allow application servers—If you enable this option, Pulse Connect Secure automatically creates a SAM resource policy that allows access to the servers specified for the role in the SAM tab application and server lists.
 - Windows SAM Options:
 - Auto-uninstall Secure Application Manager—This setting is not applicable to Pulse R3.0 or later. It applies to the previous WSAM client software only. If you enable it, it is ignored for connections that use Pulse R3.0 or later.
 - Prompt for username and password for intranet sites—If you enable this option, the Pulse Connect Secure requires users to enter sign-in credentials before connecting to sites on your internal network. This option changes Internet Explorer's intranet zone setting so that Internet Explorer always prompts the user for network sign-in credentials for an intranet site.
 - Auto-upgrade Secure Application Manager—This setting is not applicable to Pulse R3.0 or later. It applies to the previous WSAM client software only. If you enable it, it is ignored for connections that use Pulse R3.0 or later.
 - Resolve only hostnames with domain suffixes in the device DNS domains—If you enable this option, users can only browse to Web sites that are part of their login domain.
 - Session start script and Session end scripts—You can specify a script (.bat, .cmd, or .exe) to run on the user's endpoint after Pulse connects and disconnects. For example, you can specify a script that maps network drives on an endpoint to shares on protected resources when the user connects. The script must be in a location (either local or on the network) that is accessible by the user.
5. Click **Save Changes**.

To use resource profiles to specify the applications available to Pulse Secure client users:

1. Create resource profiles that enable access to client applications and destinations and configure the appropriate settings. Select **Users > Resource Profiles > SAM > Client Applications**.
2. Click **New Profile**.
3. From the Type list, select **WSAM**.
4. From the Application list, select one of the following options:
 - Custom—When you select this option, you must manually enter your custom application's executable filename (such as telnet.exe). Additionally, you can specify this file's path and MD5 hash of the executable file (although it is not required that you specify the exact path to the executable). If you enter an MD5 hash value, SAM verifies that the checksum value of the executable matches this value. If the values do not match, SAM notifies the user that the identity of the application could not be verified and does not forward connections from the application to the server.
 - Lotus Notes—Select this option to have SAM intermediate traffic from the Lotus Notes fat client application.
 - Microsoft Outlook—Select this option to have SAM intermediate traffic from the Microsoft Outlook application.
 - NetBIOS file browsing—Select this option to have SAM intercept NetBIOS name lookups in the TDI drivers on ports 137 and 139.



Note: NetBIOS file browsing is not supported for IPv6.

- Citrix—Select this option to have SAM intermediate traffic from Citrix applications.
- Domain Authentication—Select this option to allow integrated Windows applications, such as file sharing, Outlook, and so forth to authenticate to the domain controller when the client machine is part of a domain. Before using this option, you must:
 - Specify domain controllers that are reachable through the Pulse Connect Secure in the WSAM Destination list so that LDAP and Kerberos traffic can be proxied and sent to the Pulse server.
 - Configure a WSAM Access Control Policy to allow access to all domain controllers.



Note: You can configure access to a standard application once per user role. For example, you can enable one configuration of Microsoft Outlook and one configuration of Lotus Notes for the “Users” role.

5. Enter a unique name and optionally a description for the resource profile.
6. In the Autopolicy: SAM Access Control section create supporting auto policies and assign the policies to the role:
 1. If it is not already enabled, select the **Autopolicy: SAM Access Control** check box.
 2. In the **Resource** field, specify the application server to which this policy applies. You can specify the server as a hostname or an IP/netmask pair. You can also include a port.

If you select Domain Authentication from the Application list, enter your domain controller server addresses into the Resource field. You can add multiple domain controller servers if more than one is available.
 3. From the Action list, select Allow to enable access to the specified server or Deny to block access to the specified server.
 4. Click **Add**.
7. Click Save and Continue.
 1. In the Roles tab, select the roles to which the resource profile applies and click **Add**.

The selected roles inherit the autopolicy created by the resource profile. If it is not already enabled, the server also automatically enables the SAM option in the roles General > Overview page for all of the roles you select.
 2. Click **Save Changes**.
 3. Select **Users > User Realms > New User Realm**.
 4. Specify a name and, optionally, a description and then click **Save Changes** to create the realm and to display the realm option tabs.
 5. On the Role Mapping tab for the realm, create a new rule that maps all users to the role you created earlier in this procedure.

You can also use resource profiles to configure destination servers, network subnets and hosts and then add the resource profile to a role.

To use resource profiles to specify the network endpoints available to Pulse Secure client users:

1. In the admin console, choose Users > Resource Profiles > SAM > WSAM Destinations.
2. Click New Profile.
3. Enter a unique name and optionally a description for the resource profile.
4. In the WSAM Destinations section, specify which servers you want to secure using WSAM and click Add. You can specify the servers as hostname or IP/netmask pairs. You can also include a port.
5. Select the Create an access control policy allowing SAM access to this server check box (enabled by default) to enable access to the server specified in the previous step.
6. Click Save and Continue.
7. In the Roles tab, select the roles to which the resource profile applies and click Add.

The selected roles inherit the autopolicy created by the resource profile.

Pulse Connection Set Options for Pulse Connect Secure

A Pulse Secure client connection set contains network options and allows you to configure specific connection policies for client access to any Pulse server that supports Pulse Secure client. The following sections describe each of the configuration options for a Pulse connection set.

Pulse Secure Connection Set Options

The following items apply to all connections in a connection set.

- **Allow saving logon information**—Controls whether the Save Settings check box is available in login dialog boxes in the Pulse client. If you clear this check box, the Pulse client always requires users to provide credentials. If you select this check box, users have the option of saving their credentials.

The Pulse Secure client can retain *learned user settings*. These settings are retained securely on the endpoint, evolving as the user connects through different Pulse servers. The Pulse Secure client can save the following settings:

- Certificate acceptance
- Certificate selection
- Realm
- Username and password
- Proxy username and password
- Secondary username and password
- Role

When a user opts to save settings, that information is used for each subsequent connection without prompting. If a setting changes (for example, if a user changes a password), the saved setting is invalid and connection attempts fail. In this case, the user must use the client's Forget Saved Settings feature, which clears all user-saved settings.



Note: If the authentication server is an ACE server or a RADIUS server and authentication is set to Users authenticate using tokens or one-time passwords, Pulse ignores the Allow saving logon information option. If the user sees a username and token prompt and the Save settings check box is disabled. Pulse supports soft token, hard token, and smart card authentication.



Note: In 5.2R5 Pulse Secure desktop client introduced two new features to improve the end-user experience during certificate authentication. The administrative console option to configure this feature is now available in 5.3R2. This feature enables the following:

- Improved automatic certificate-selection algorithm
- Ability to prefer smart-card certificates over other certificates

- **Allow user connections**—Controls whether connections can be added by the user.
- **Always-on Pulse Client**—Prevent end users from circumventing Pulse connections. This option disables all configuration settings that allow the end user to disable or remove Pulse connections, service or software. For more details refer to Always-on VPN.

Note: Checking the “Always-on Pulse Client” option does not prevent end users with administrative privileges from stopping the Pulse Secure service on the endpoint device. Create a group policy object (GPO) to prevent users from disabling the Pulse Secure service. For more details on how to create GPOs refer to the article found in Microsoft’s Website.

Figure 65: Pulse Secure Connection Set Options

Name	Value
Allow saving logon information Enables the Save settings checkbox in the certificate trust and password prompts.	<input checked="" type="checkbox"/>
Allow user connections Allows user to create connections via the Pulse UI.	<input checked="" type="checkbox"/>
Always-on Pulse Client Prevents end users from circumventing Pulse connections. This option will disable all configuration settings that allow the end user to disable or remove Pulse connections, services or software.	<input type="checkbox"/>
Display Splash Screen Controls whether the splash screen is displayed when Pulse starts.	<input checked="" type="checkbox"/>
Dynamic certificate trust Controls whether users may accept to trust unknown certificates.	<input type="checkbox"/>
Dynamic connections Allows connections to be deployed automatically from devices.	<input checked="" type="checkbox"/>
EAP Fragment Size Maximum number of bytes in an EAPOL message from the client for 802.1x connections. Range: 450 - 3000 bytes	<input type="text" value="1400"/>
Enable captive portal detection Pulse will attempt to detect the presence of a captive portal hotspot. Only applies to Connect Secure and Policy Secure (L3) connections.	<input type="checkbox"/>
Enable embedded browser for authentication Pulse will use embedded browser for saml, custom sign-in or token based authentication.	<input type="checkbox"/>
Enable embedded browser for captive portal Pulse will use an embedded web browser for captive portal pages. Only applies when captive portal detection is enabled.	<input type="checkbox"/>
FIPS mode enabled Deploy client with Federal Information Processing Standard enabled.	<input type="checkbox"/>

- **VPN only access**— When the Pulse client connects to Pulse Connect Secure having lock down mode enabled, it will enable lock-down mode and block network if VPN is not in connected state.
 - When **VPN only access** option is enabled, the Enable captive portal detection and Enable embedded browser for captive portal will be automatically checked and cannot be edited.
- **Display splash screen**—Clear this check box to hide the Pulse splash screen that normally appears when the Pulse client starts.
- **Dynamic certificate trust**—Determines whether users can opt to trust unknown certificates. If you select this check box, a user can ignore warnings about invalid certificates and connect to the target Pulse server.

Note: By default, **Dynamic certificate trust** check box will be unchecked.

- **Dynamic connections**—Allows connections within this connection set to be automatically updated or added to a Pulse Secure client when the user connects to the Pulse server through the user Web portal, and then starts Pulse through the Web portal interface. Dynamic connections are created as manual rather than automatic connections, which means that they are run only when the user initiates the connection or the user browses to a Pulse server and launches Pulse from the server’s Web interface.

If dynamic connections are disabled, and the user logs in through the Web portal of a Pulse server that is not already included in the Pulse client’s connection set, then starting Pulse from the Web portal does not add a new Pulse

connection for that Pulse server. If you choose to disable dynamic connections, you can still allow users to manually create connections by enabling Allow User Connections.

- **Enable captive portal detection**—To detect the presence of a captive portal hotspot enable this option. It can be applied only to Pulse Connect Secure and Pulse Policy Secure (L3) connections.
- **Enable embedded browser for captive portal**—When enabled, pulse uses an embedded web browser that the end user can use to traverse captive portal pages and to gain network connectivity for establishing a VPN connection. This applies only when captive portal detection is enabled.
- **Enable embedded browser for authentication**—When enabled, pulse uses an embedded browser for web authentication, rather than external browser.
- **FIPS mode enabled**—Enable FIPS mode communications for all Pulse connections in the connection set. The Federal Information Processing Standard (FIPS) defines secure communications for the U.S. government. When a Pulse connection is operating in FIPS mode, **FIPS On** appears in the lower corner of the Pulse client interface. If the Pulse server hardware does not support FIPS mode operations, FIPS mode configuration options are not present in the admin console interface. FIPS mode operations are supported on PSA-V Series Pulse Secure Gateways and some SA series appliances. The device must be running Pulse Policy Secure R5.0 or later or Pulse Connect Secure R8.0 or later.



Note: Users cannot enable FIPS mode from within the Pulse client. You must create FIPS-enabled connections on the server and deploy them.

- **Prevent caching smart card PIN**—Enabling this field will allow system administrators to prevent smart card PIN values from being cached. This feature is applicable only to Windows.
- **Wireless suppression**—Disables wireless access when a wired connection is available. If the wired connection is removed, Pulse enables the wireless connections with the following properties:
 - Connect even if the network is not broadcasting.
 - Authenticate as computer when computer information is available.
 - Connect when this network is in range.



Note: Wireless suppression occurs only when the wired connection is connected and authorized. If you enable wireless suppression, be sure to also configure a connection that enables the client to connect through a wired connection.

Configuring Client Certificate Selection Option

From 9.1R3 release onwards, while [Configuring Pulse Connect Secure](#) settings, the following are the new checkboxes added under **Client Certificate Selection Option**.

- **Accept certificates with smartcard logon ECU**
- **Accept certificate with Custom ECU text**
- **Accept certificate with Custom ECU OID**

The valid certificates get filtered based on the specifications provided by administrator in the above fields.

Only the filtered certificates get displayed in certificate selection prompt.

Enhanced Key Usage (EKU) field, abbreviated as ECU has following two components to it.

- **EKUText** - It is the text which is in human readable format.
- **ECUOID** - It is the OID number which is unique for a given purpose.

The following table defines **EKUText** and **ECUOID** variables:

Table 12: Variables Description

Variable	Description	Examples
EKUText	<p>Format to be given is: EKUText = string or <comma separated string> or string with regular expression.</p> <p>Custom regular expressions need to be given with the following format: certAttr.EKUText = string or <comma separated string> or string with regular expression.</p>	certAttr.EKUText = "TLS Web Server Authentication", "E-mail Protection", "TLS Web Client Authentication".
EKUOID	<p>Format to be given is: EKUOID = to a.b.c.d.e.f.g.h.i or <comma separated list of EKUOIDs> or OID with regular expressions. This works in both certificate rule as well as custom expressions.</p> <p>Custom regular expressions need to be given with the following format: certAttr.EKUOID = a.b.c.d.e.f.g.h.i or <comma separated list of EKUOIDs> or OID with regular expressions.</p>	<p>Customer can create certificates with Custom OIDs. Example: certAttr.EKUOID=1.3.6.1.5.5.7.3.1,1.3.6.1.5.5.7.3.4,1.3.6.1.5.5.7.3.2</p>

To configure **Certificate Matching** and **Certificate Ranking** under **Client Certificate Selection Option**, the administrator needs to follow the below steps.

1. Log in to admin console.
2. Go to **Users > Pulse Secure Client > Connections**.
3. Select the connection from the **Connections** or click **New** to display the New Connection set configuration page.
4. Complete the configuration, as required.
5. Save the configuration.

Figure 66: Client Certificate Selection Option

▼ Client Certificate Selection Option

Certificate Matching

☒ Accept certificates with smartcard logon Enhanced Key Usage

☐ Accept certificates with custom Enhanced Key Usage OID

☐ Accept certificates with custom Enhanced Key Usage text [Examples: TLS Web server.*, TLS Web client.*, Code signing.*, E-mail protection.*]

Certificate Ranking

☐ Enable Automatic Client Certificate Selection

This option uses a proprietary certificate ranking algorithm to choose the most suitable client certificate.



Note: Prefer smart card certificate option can be checked only if **Enable Automatic Client Certificate Selection** option is checked.

If **Prefer smart card certificate** option is checked, then certificates with client auth ECU set get displayed on the top of the list of certificates and preferred over other certificates.

Accept certificates with smartcard logon Enhanced key Usage option is enabled by default.



Note: **Accept Certificates with smartcard logon Enhanced key Usage** option should be checked to use Yubikey as PIV smart card for VPN authentication.

Based on the ECU configuration settings, Pulse Desktop Client will pick-up the available certificate and make successful connection.

If end-user has more than one certificates, then Pulse Desktop Client will prompt the end-user to select the certificates to make successful connection.

Always-on VPN

By default, Always-on option is disabled. There are many possible configuration options within the Always-on feature. Although some of these options are new for the 5.2r5 Pulse Secure desktop client (e.g., lock-down mode, embedded browser for captive-portal remediation), some Always-on options existed in previous versions of the Pulse client. To make management of all these options easier, the 8.2r5 PCS gateway's administrative console provides a simplified way of configuring all the possible Always-on VPN options.

Note:

- From 9.0 R1 release onwards, Pulse Desktop Client for macOS will support Always-on VPN except **Lock-Down Exception** function.
- Pulse Desktop Client for Linux will not support for Always-on VPN.

Configuring Always-on Options

To configure the Connection set:

1. Login to Pulse Connect Secure admin console
2. Select **Users > Pulse Secure Client > Connections**
3. Click **New** to display the **New Connection set** configuration page, refer [Figure 65](#).
4. Complete the configuration as described in Table 13.
5. Save the configuration.

Table 13: Always-on options Settings

Settings	Description
Allow user connection	Controls whether connections can be added by the user.
Always-on Pulse Client	When checked it prevents end users from circumventing Pulse connections. This option will disable all configuration settings that allow the end user to disable or remove Pulse connections, service or software.
Enable captive portal detection	Controls whether the Pulse desktop client will notify the end user that a VPN connection cannot be established until the requirements of a captive portal are fulfilled.
Enable embedded browser for captive portal	When checked, the Pulse client will use an embedded web browser for captive portal pages.



Note: When Always-on Pulse Client is enabled "VPN Only Access", "Enable captive portal detection" and "Enable embedded browser for captive portal" will be automatically checked and cannot be edited.

Checking this option will modify several checkboxes in both the Connection Set and the Connections within the Connection Set with the effect of:

- Impeding the end user's ability to disconnect or disable VPN connections (Windows and Mac)
- Ensuring that captive portals can still be traversed even when connectivity is locked down (Windows and Mac)



Note: "Always-on" checkbox does not prevent end users (with admin privileges) from stopping endpoint services (the Pulse Secure Service and the Base Filtering Engine (BFE)) which are required for VPN connections to be established. If you wish to have the level of protection that comes with prohibiting end users from stopping these services, then it is best to use Group Policy Objects (GPOs).

Configuring Always-on VPN Options using Wizards

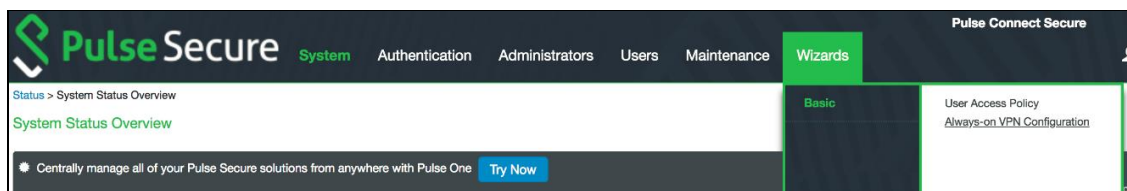
To configure the Always-on VPN Options using wizards.

1. Login to Pulse Connect Secure admin console.
2. Admin can configure Always-on VPN options using Wizards in following two ways:
 - Using Global Wizards
 - Using Connection Set

Using Global Wizards

1. Click **Wizards**. Following screen appears:

Figure 67: Wizards Screen



2. Select **Wizard -> Basic -> Always-on VPN Configuration**. Following screen appears:

Figure 68: Always-on VPN connection set details

Always-on VPN Wizard:connection set details

Detected below connection sets on the server. Please select the connection set to configure always-on VPN and dependent options.

- AoPVN
 ▼

Do you want to continue configuration through Always-on VPN wizard?

No
Yes

3. Choose the connection set from the drop-down list of existing connection sets. [Figure 70](#) appears.
Go to [Step-5](#) to continue with configuration of Always-on VPN.

Using Connection Set

1. Select **Users > Pulse Secure Client > Connections**. Go to [Step-3](#) to continue with configuration of Always-on VPN.
3. Click **New** to display the New Connection set configuration page. The following screen appears:

Figure 69: New Connection Set

Pulse Secure Client > Connections > New Connection Set

New Connection Set

Name:

Description:

Owner:
Last Modified: 2018-02-15 22:46:04 UTC
Server ID: 0311M68Q502570IQS

▼ Always-on vpn wizard

[Configure Always-on VPN using wizard](#)

▼ Options

Name	Value
Always-on Pulse Client Prevents end users from circumventing Pulse connections. This option will disable all configuration settings that allow the end user to disable or remove Pulse connections, services or software.	<input type="checkbox"/>
VPN only access When Pulse client connects to a PCS having lock down mode enabled, it will enter lock-down mode and won't let any traffic flow through unless a Locked-down VPN connection is in connected state. User is allowed to connect or disconnect any connection. User is allowed to add any new connection/server URI. User is allowed to delete a connection if the connection is not locked down.	<input type="checkbox"/>
Allow saving logon information Enables the Save settings checkbox in the certificate trust and password prompts.	<input checked="" type="checkbox"/>

Create new connection set with default values. For more information on Connection set, refer to [Creating a Client Connection Set for Pulse Connect Secure](#).

Note: Admin can edit the existing connection set also.

4. Click **Configure Always-on VPN using wizard**. The following screens appears:

Figure 70: Configure Always-on VPN using wizard

Always-on VPN

Introduction

The functionality of Always-on VPN feature with lock down mode enabled is to deny all network traffic until connected via VPN. Exemption rules can be configured to exempt certain types of traffic

Click 'Next' to configure Always-on VPN.

Cancel < Previous Next >

5. if admin has already configured Always-on VPN, the following screen appears:

Figure 71: Always-on VPN

Always-on VPN

Introduction

The functionality of Always-on VPN feature with lock down mode enabled is to deny all network traffic until connected via VPN. Exemption rules can be configured to exempt certain types of traffic

Always-on VPN is turned ON already, Please click OFF to turnoff this feature and exit the wizard. else, click 'Next' to configure this feature and dependent options.

TURN-OFF ALWAYS-ON VPN

Cancel < Previous Next >

Click **TURN-OFF ALWAYS-ON VPN**, if you want to disable Always-on VPN option. Also, VPN only access will get disabled.

6. Click **Next** to continue with configuration of Always-on VPN. The following screen appears:

Figure 72: Can users connect or disconnect VPN

The screenshot shows the 'Always-on VPN' configuration window. On the left is a sidebar with four tabs: 'Introduction' (blue), 'Connection Options' (green), 'Exception Configuration' (grey), and 'Summary' (grey). The main area is titled 'Connection Options' and contains the following settings:

- Can users connect or disconnect VPN**: ☐
- Connection Name: Test**
- Lockdown this connection**: ☐
- Allow user to override connection policy**: ☐
- Connection Mode: User**
- Reconnect at Session Timeout or Deletion**: ☒
- Connect automatically**: ☒

At the bottom of the window are three buttons: 'Cancel' (grey), '< Previous' (blue), and 'Next >' (blue).

Check the **Lockdown this connection** checkbox. Following screen appears:

Scenario 1 – Always-on VPN

Figure 73: Can users connect or disconnect VPN (Lockdown mode)

Always-on VPN

Introduction
Connection Options
Exception Configuration
Summary

Connection Options

Can users connect or disconnect VPN

Connection Name: Test

Lockdown this connection.

Cancel < Previous Next >

Scenario – 2 VPN Only Access

Figure 74: Can users connect or disconnect VPN

The screenshot shows the 'Always-on VPN' configuration window. On the left is a sidebar with four tabs: 'Introduction' (blue), 'Connection Options' (green), 'Exception Configuration' (grey), and 'Summary' (grey). The main area is titled 'Connection Options' and contains the following settings:

- Can users connect or disconnect VPN** ☒
- Allow user connections** ☒
- Connection Name:** Test
- Lockdown this connection.** ☒
- Allow user to override connection policy.** ☒
- Connection Mode:** User
 - Reconnect at Session Timeout or Deletion.** ☒
 - Connect automatically.** ☒

At the bottom of the window are three buttons: 'Cancel' (grey), '< Previous' (blue), and 'Next >' (blue).

7. Check **Can users connect or disconnect VPN** checkbox to continue with Always-on VPN configuration.
8. Click Next. The following screen appears:

Scenario – 1: Always-on VPN

Figure 75: Lock down mode exception configuration

Always-on VPN

[Introduction](#)
[Connection Options](#)
[Exception Configuration](#)
[Summary](#)

Exception configuration

Lock down mode exception configuration

When Always-on VPN Feature with Lockdown mode enabled, Admin can add more exceptions to the Core Access Rules using exception rules. Exceptions already configured in client are called core Access Rules. DHCP, DNS, Kerberos, LDAP, SMP and Portmapper are already configured core access rules in client. Exemption rules can be configured to exempt certain types of traffic. Program based and Port based exceptions can be added using this wizard, custom rule can be added from connection-set

[Create new exception](#)

Name	Direction	Program	Protocol
Port ⓘ	inbound	ANY	TCP

[Cancel](#)
[< Previous](#)
[Next >](#)

Configure the exception rules. Admin can add more exception rules to the Core Access Rules.

9. Click **Create new exception**. The following screen appears:

Figure 76: Lock down mode exception configuration

The screenshot shows a web-based configuration interface for 'Always-on VPN'. On the left is a sidebar with four tabs: 'Introduction', 'Connection Options', 'Exception Configuration' (which is selected and highlighted in blue), and 'Summary'. The main content area is titled 'Always-on VPN' in green. Below this title is a 'New exception' dialog box. The dialog has the title 'Lock down mode exception configuration'. It contains the following fields and options:

- Name:** A text input field.
- Description:** A larger text input field.
- Direction:** Two radio button options: 'Inbound' and 'Outbound'.
- Exception type:** Two radio button options: 'Program' and 'Port'.
- At the bottom of the dialog are two blue buttons: 'Skip' and 'Submit'.

Below the dialog box, at the bottom left of the main content area, is a 'Cancel' button.

10. Click **Submit**, if exception rules are set.

11. Click **Skip**, if admin wants to change the exception rules.

Scenario – 1: Always-on VPN

Figure 77: Summary Screen

Always-on VPN

Introduction

Connection Options

Exception Configuration

Summary

Summary

Introduction	
Always-on pulse client	ON
VPN-only access	OFF
Allow user connection	OFF
Connection configuration:Test	Details in tab-2
Exception configuration	Details in tab-3

Cancel

< Previous

Finish >

Scenario – 2 VPN Only Access

Figure 78: Summary Screen

Summary	
Introduction	
Always-on pulse client	OFF
VPN-only access	ON
Allow user connection	ON
Connection configuration:Test	Details in tab-2
Exception configuration	Details in tab-3

12. Click **Finish**. 'Always-on VPN update successful.' message appears on the screen.

Note: 'Always-on VPN update successful' message appears along with connection set page (see [Figure 69](#)), if Admin is configuring Always-on VPN using Wizards through **Wizard** tab in PCS console.

Requirement to set up the appropriate GPOs

Setting up GPOs is not required to leverage the benefits of the Pulse Secure desktop client's Always-on VPN feature. Setting up GPOs would only be necessary if you need the extra security of restricting users with admin privileges from stopping Pulse services. As such, setting up GPOs should be considered optional.

The Pulse Secure Desktop Client should be installed on a box running a server version Microsoft Windows (e.g., Windows server 2008 R2, 2012 R2, 2016).

- The startup type for "Pulse Secure Service" should be set to "Automatic", and permission to start and stop the service should be removed from "Administrators".
- Ensure that "SYSTEM" still retains permission to start and stop the service.
- A "Pulse Secure Admins" should be created on the domain. Permission to start and stop the service should be assigned to this "Pulse Secure Admins". The "Domain Admins" and any other group which should be allowed to start and stop Pulse Secure can be made members of the "Pulse Secure Admins" group.

- Disabling the ability to stop the Base Filtering Engine (BFE) should be done in a manner similar to what is described above for the Pulse Secure Service.

Installing Pulse Desktop Client in Windows Server

To install Pulse Desktop Client in Windows Server, follow the below steps:

1. The Pulse Secure Desktop Client MSI file should be used for installation.
2. Install Wireless LAN service on Windows 2008 R2, 2012 R2 and 2016 servers before installing the Pulse Client to avoid unsuccessful registration during the Pulse Client Installation.
3. To enable the Wireless LAN service, follow the below steps:
 - a. Open **Server Manager**.
 - b. Navigate to **Feature > Add Feature**.
 - c. Select the **Wireless LAN Service**.
 - d. Click on **install** Wireless LAN Service for installing.
 - e. Click on **Close > Done** once the Wireless LAN Service is installed.



Note: On Windows 2016 servers, even after Wireless LAN Service is installed, an error message will appear during the Pulse Client Installation and the error can be accepted and the Pulse Installation will be completed.



Note: There is no mechanism within the Pulse desktop client itself to prevent end users with administrative privileges from uninstalling the Pulse desktop client. If this functionality is needed, then it would be best to enforce those restrictions outside of Pulse.

Always-on with Lock-down Mode

“Lock-down” mode is a new aspect of Always-on functionality that was added to the 5.2r5 Pulse desktop client for Windows and 9.0R1 for Mac. Lock-down mode prohibits network communication outside the VPN tunnel when a VPN tunnel is in the process of being created. Lock-down option can be enabled only in conjunction with the “Always-on” and “VPN only access” option. To ensure that end users can easily traverse captive portals in lock-down mode, the “Captive portal remediation with embedded mini-browser” is automatically enabled when lock-down mode is enabled. Lock-down mode is intended for use with Location Awareness rules; this feature can ensure that the user is either:

- Physically on the corporate network,
- Connected to the corporate network through a VPN connection or on the process of creating a VPN connection and cannot access the Internet/local subnet in the meantime.

Location Awareness rules should be set up to automatically initiate a VPN connection when the user is not on the corporate network, and to disconnect the VPN connection when the user is physically on the corporate network. If Location Awareness rules are not configured in this method, then Lock-down mode has very little value, because Lock-down mode prohibits connectivity only when the Pulse client is in the process of creating a network connection. If the Pulse client is not configured to automatically create a Pulse connection when off the corporate network, then Lock-down mode will not be automatically invoked when the user leaves the corporate network. For information on configuring Location Awareness rules, refer [Location Awareness Rules](#).

The lock-down option blocks nearly all network traffic, but there are exceptions for the minimum amount of traffic required to initialize network adapter such that a tunnel can be created. As such, traffic used to get IP addresses, hostnames, etc. (DHCP, DNS, etc.) are permitted even when the machine is locked down.



Note: Lock-down mode is supported only for IPv4 endpoints.



Note: For known third party software issues with Lock-down Mode, refer to KB article [KB43679](#).

To enable the “Lock down this connection” option, follow the below steps:

1. On the **Pulse Connect Secure admin console**.
2. Select the connection from **Connection Options**.
3. Use a Connect Secure L3 connection for a Layer 3 connection to Pulse Connect Secure.
4. Check the Lock – down this connection to disable network access when VPN is enabled until connected, see [Figure 79](#).
5. Click on **Save Changes**.

Figure 79: Lock down this connection

Pulse Secure Client > Connections > employeeconn > employeeconn

employeeconn

Name:

Description:

Type: Connect Secure or Policy Secure (L3)

▼ Options:

Name	Value
Allow user to override connection policy Allows user to modify connection state.	<input type="checkbox"/>
Lock down this connection Network access is limited until this connection is established. This option is available only when the Always-on Pulse Client option or VPN only access option on the connection set is checked.	<input checked="" type="checkbox"/>

Lock-down Exception

When Pulse Desktop Client is in Lock-down mode, all network traffic except those defined in Lock-down exception rules will be denied when VPN is not connected.

In the New Configuration section, administrator can add Lock-down mode exceptions rules for Windows and for 9.0R2 release onwards for macOS. Administrator has to configure these rules for which traffic need to be exempted when Lock-down mode has applied at user end.

From 9.0R2 release onwards, administrator can also select the platform from 'Platform' tab.

To configure exception rules, an administrator user needs to follow the below steps.

1. Log in to admin console.
2. Go to **Users > Pulse Secure Client**.
3. Select **Connections**.
4. **Select Options > Enable Always-on Pulse Client & VPN only access.**

Lock-down Exception can be enabled by selecting Always-on Pulse Client alone or VPN only Access.

Administrator can add more exceptions to the Core Access Rules using exception rules. DHCP, DNS, Kerberos, LDAP, SMP and Portmapper are already configured core access rules in Pulse Desktop client.

Figure 80: Connections

✓ Lockdown mode exception rules:

When Always-on VPN Feature with Lockdown mode enabled, Admin can add more exceptions to the Core Access Rules using exception rules. Exceptions already configured in client are called core Access Rules. DHCP, DNS, Kerberos, LDAP, SMP and Portmapper are already configured core access rules in client.

[New...](#) [Duplicate...](#) [Delete...](#)

	Name	Program	Protocol	Direction	Local Address	Remote Address	Local Port	Remote Port	Platform

Lock-down Exception rule can be configured in the following three ways under Resources for both “Inbound*” and “Outbound*” traffic separately.

- [Program](#)
- [Port](#)
- [Custom](#)

Inbound traffic is always directed towards user’s machine (Example: RDP).

Outbound traffic is always directed towards outside the machine (Example: Skype for Business Application).

Figure 81: New Always-on VPN Exception Rules

Pulse Secure Client > Connections > employeeconn > New Lock down exception rule

New Lock down exception rule

Windows Mac All

Name:

Description:

Inbound ☐ Outbound ☐

✓ Resources

Select exception type:

☐ Program ☐ Port ☐ Custom

[Save Changes](#) [Cancel](#)

Windows: Select **Windows** to define exception rules for only Windows.

Mac: Select **Mac** to define exception rules for only Mac.

All: Select **All** to define exception rules for both Windows and Mac.

Program-based Resource Access

To configure Program-based resource access, administrator needs to select **Program**. Then the following configuration UI appears.

Windows – Program based Resource Access

Figure 82: Windows - Program-based Resource Access

▼ Resources

Select exception type:

☒ Program ☐ Port ☐ Custom

Program path: Example: C:\Program Files (x86)\Google\Chrome\Application\chrome.exe

SHA2-256: SHA2-256 hash of program executable

macOS – Program based Resource Access

Figure 83: macOS - Program-based Resource Access

▼ Resources

Select exception type:

☒ Program ☐ Port ☐ Custom

Program path: Example: (Safari browser)
/Applications/Safari.app/Contents/MacOS/Safari
/Applications/Safari.app/Contents/MacOS/Safari
/System/Library/PrivateFrameworks/SafariSafeBrowsing.framework/Versions/A/com.apple.Safari.SafeBrowsing.Service
/System/Library/Frameworks/WebKit.framework/Versions/A/XPCServices/com.apple.WebKit.Networking.xpc/Contents/MacOS/com.apple.WebKit.Networking

SHA2-256: SHA2-256 hash of program executable

An administrator has to provide absolute path for the program that needs to be exempted and optionally provide SHA-256 checksum.

Following are the examples for Lockdown Exception rules for macOS.

1. **MAC Update program path:**

```
/Applications/App Store.app/Contents/MacOS/App Store
/System/Library/PrivateFrameworks/StoreXPCServices.framework/Versions/A/XPCServices/com.apple.appstore.PluginXPCService.xpc/Contents/MacOS/com.apple.appstore.PluginXPCService
/System/Library/Frameworks/ApplicationServices.framework/Versions/A/Frameworks/HIServices.framework/Versions/A/XPCServices/com.apple.hiservices-xpcservice.xpc/Contents/MacOS/com.apple.hiservices-xpcservice
/System/Library/CoreServices/Software Update.app/Contents/Resources/softwareupdated
/System/Library/Frameworks/WebKit.framework/Versions/A/XPCServices/com.apple.WebKit.Networking.xpc/Contents/MacOS/com.apple.WebKit.Networking
```

2. **Safari browser program path:**

```
/Applications/Safari.app/Contents/MacOS/Safari
/System/Library/PrivateFrameworks/SafariSafeBrowsing.framework/Versions/A/com.apple.Safari.SafeBrowsing.Service
/System/Library/Frameworks/WebKit.framework/Versions/A/XPCServices/com.apple.WebKit.Networking.xpc/Contents/MacOS/com.apple.WebKit.Networking
/System/Library/StagedFrameworks/Safari/WebKit.framework/Versions/A/XPCServices/com.apple.WebKit.Networking.xpc/Contents/MacOS/com.apple.WebKit.Networking -> Mac 11 & Mac 12
```

3. **Facetime program path:**

```

/System/Library/PrivateFrameworks/ApplePushService.framework/apsd
/Applications/FaceTime.app/Contents/MacOS/FaceTime
/System/Library/PrivateFrameworks/AuthKit.framework/Versions/A/Support/akd
/System/Library/PrivateFrameworks/IDS.framework/identityservicesd.app/Contents/MacOS/identityservicesd
/System/Library/PrivateFrameworks/AOSKit.framework/Versions/A/XPCServices/com.apple.iCloudHelper.xpc/Contents/MacOS/com.apple.iCloudHelper
/usr/libexec/avconferenced
/usr/libexec/nsurlsessiond

```

4. **Symantec Norton security program path:**

```

/Applications/Norton Security.app/Contents/MacOS/Norton Security
/Library/Application Support/Symantec/Silo/NFM/Daemon/SymDaemon.bundle/Contents/MacOS/SymDaemon
/Library/Application Support/Symantec/Silo/NFM/LiveUpdate/com.symantec.SymLUHelper
/Library/Application Support/Symantec/Silo/NFM/SymUIAgent/Norton.app/Contents/MacOS/Norton

```

Each process needs to configure with different process rules, and not with single process.

Figure 84: Lockdown Exception Rules - Safari

New... Duplicate... Delete...								
	Name	Program	Protocol	Direction	Local Address	Remote Address	Local Port	
1.	Safari	/Applications/Safari.app/Contents/MacOS/Safari		Outbound	any	any	any	
2.	Safari1	/System/Library/PrivateFrameworks/SafariSafeBrowsing.framework/Versions/A/com.apple.Safari.SafeBrowsing.Service		Outbound	any	any	any	
3.	Safari2	/System/Library/Frameworks/WebKit.framework/Versions/A/XPCServices/com.apple.WebKit.Networking.xpc/Contents/MacOS/com.apple.WebKit.Networking		Outbound	any	any	any	

When a lockdown is applied. Use below command to ping IPv6 address.

ping6 -S (Source IPv6) (destination IPv6)

Port-based Resource Access

To configure Port-based resource access, administrator needs to select **Port**. Then the following configuration UI appears.

Windows and macOS Port based Resource Access

Figure 85: Port-based Resource Access

▼ Resources

Select exception type:

☐ Program
 ☒ Port
 ☐ Custom

TCP ☐

UDP ☐

Local Port : Example: 80,443,5000-5010

An administrator can select either TCP or UDP and needs to add respective port number.

Custom-based Resource Access

To configure Custom-based resource access, administrator needs to select **Custom**. Then following configuration UI appears.

Figure 86: Custom-based Resource Access

Resources

Select exception type:
☐ Program ☐ Port ☒ Custom

Program path:

SHA2-256: SHA2-256 hash of program executable

Protocol:

Local IPV4/IPV6 Resources: Specify the IP address range, one per line: Examples:
 [default value is *] 10.10.10.10-10.10.10.100
 10.10.10.10/255.255.255.0
 10.10.10.50
 [2001:db8:a0b:12f0::1]
 [2001:DB8::6:0/112]
 [2001:DB8::7:50]

Remote IPV4/IPV6 Resources: Specify the IP address range, one per line: Examples:
 [default value is *] 10.10.10.10-10.10.10.100
 10.10.10.100/255.255.255.0
 10.10.10.50
 [2001:db8:a0b:12f0::1]
 [2001:DB8::6:0/112]
 [2001:DB8::7:50]

Local Port : Example:80,443,5000-5010
 Remote Port: Example:80,443,5000-5010

Once Custom is selected, administrator can configure any of the applicable rules. (Example: administrator can configure Local or Remote IPv4/IPv6 addresses or Ports or Absolute Program Path or Type of protocol).

Note:

- In custom-based resource access, it is not mandatory to configure all the options. Default value for all configurable fields are "**".
- Post Pulse Desktop client upgrade, user has to make at least one successful connection to the Pulse Connect Secure, so that the configured Lock-down exceptions can be applied on the client machine.

Retry Button

The Retry button allows the user to reconnect the connection after a time out or failure. When an authentication has failed, Always-on VPN attempts to reestablish the connection to activate the session if the session time is still open or user can reconnect to new VPN session by clicking on Retry button.

The retry option can be found on Pulse Desktop client user interface,

1. Open the **Pulse Desktop client**.
2. Click on **File > Connections > Retry to reconnect**.

Captive Portal Remediation with Pulse Client Embedded Mini-Browser

The “Enable embedded browser for captive portal” option makes it easy for end users to satisfy captive portals (for example, when in a coffee-shop that requires either credit-card payment or acceptance of an acceptable-use policy before gaining Wi-Fi access). By default, support for captive portal remediation is disabled.

To enable Embedded Browser:

1. Log in to Pulse Connect Secure admin console.
2. Select **Users> Pulse Secure Client > Connections**.
3. Click **New** to display the **New Connection set** configuration page.
4. Complete the configuration as described in **Table 13**.
5. Save the configuration.

Table 14: Pulse Client Embedded Mini-Browser Settings

Settings	Guidelines
Enable Captive Portal detection	If this option is checked, Pulse will detect if connectivity is hampered by a captive portal, then the Pulse client will automatically display an embedded browser (not an external browser, like IE or Chrome or Safari) so that the end user can traverse the captive portal and gain the network connectivity needed to establish a VPN connection.
Enable embedded browser for captive portal	Pulse will use an embedded browser for captive portal pages, applicable only if Captive Portal detection is enabled.

Although this feature can be used as a convenience independent of the Always-on VPN and VPN only access feature, it is essential (and is enabled automatically) when using lock-down mode. The embedded browser is part of the Pulse client’s internal processes, and is therefore exempt from the lock-down connectivity restrictions placed on external browsers. Lock-down mode prevents external browsers from communicating before the VPN is established, so external browsers cannot be used for captive-portal remediation with lock-down mode enabled. The embedded browser is the only option for remediating a captive portal in lock-down mode.

For macOS, Captive Portal Remediation using external browser needs admin privileges if system proxy settings are configured. Embedded browser will bypass proxy settings automatically without admin privileges.

For Windows OS, user can bypass proxy settings and perform captive portal remediation using external browser.

The Pulse client’s embedded browser is restricted to ensure that end users cannot use it for purposes other than traversing captive portals. Furthermore, certain web-browser functionality is disabled to make the embedded browser more secure.

The following table describes features which are enabled and disabled:

Table 15: Enabled and Disabled features

Features	Enabled
Display of images & playing of sound	Yes
Running scripts	Yes
Display of script errors	Yes
Display popup windows and dialogs	Yes
Running JAVA	No
Downloading or running ActiveX	No

Features	Enabled
Downloading Files	No

Policy Secure 802.1X Connection Type Options

Use this connection type to define authenticated connectivity to 802.1X devices, wired or wireless. Users cannot create 802.1X connections from the Pulse client interface. Users see 802.1X connections in the Pulse interface only when the connection has been deployed from the server and the specified network is available.

- **Adapter type**—Specifies the type of adapter to use for authentication: wired or wireless.
- **Outer username**—Enables a user to appear to log in anonymously while passing the actual login name (called the inner identity) through an encrypted tunnel. As a result, the user's credentials are secure from eavesdropping, and the user's inner identity is protected. In general, enter **anonymous**, which is the default value. In some cases, you might need to add additional text. For example, if the outer identity is used to route the user's authentication to the proper server, you might be required to use a format such as [anonymous@acme.com](#).



Note: If you leave the box blank, the client passes the user's or the machine's Windows login name as the outer identity.

- **Scan list**—If you selected wireless as the adapter type, the scan list box is available to specify the SSIDs, including non-broadcast SSIDs, to connect to in priority order. If you leave the list empty, the user can connect to any available wireless network.
- **Support Non-broadcast SSID**—Allows a user to connect to a non-broadcast wireless network from within the Pulse interface. Selecting this field enables the following options:
 - **Wireless Security Cipher**—Specify the type of encryption used by the non-broadcast network:
 - TKIP
 - AES

If the non-broadcast SSID options are configured, the Pulse connection configuration includes the values and they are used to configure the wireless profile on the endpoint.

Trusted Server List (for Policy Secure 802.1X Connection)

FQDN criteria for 802.1X/EAP server certificates (with wildcard support) can be specified in the Trusted Server List of the PCS admin console. In the name field, you can enter a fully-qualified-domain name (FQDN) that can be either an exact FQDN or an FQDN that begins with a "." and/or can contain wildcards ("*").

Note the Following:

- The "ANY" entry matches any server certificate name.
- An entry that contains "=" requires an exact Subject:DN (Distinguished Name) match.
- An entry that is neither "ANY" nor contains "=" is an FQDN. It can be either an exact value or include wildcards and/or begin with a "." character. This value will be checked against FQDNs in the server's certificate (Subject:DN:CN=..., SAN:DNS=...).
 - An entry that begins with "." will wildcard only the first subdomain (domain component) in the FQDN. For example, ".mycompany.com" will match "foo.mycompany.com" but not "foo.bar.mycompany.com". As such, a FQDN beginning with "." is equivalent to the same FQDN beginning with "." (e.g., ".mycompany.com" is equivalent to ".mycompany.com"). Note that this mechanism is more restrictive than what is described in RFC 5280.
 - FQDN may contain at most one wildcard per domain component (DC). For example, "a.mycompany.com" is not allowed and will always result in authentication failure.

- A wildcard matches 1 or more characters (but not zero characters). For example, "f*r.mycompany.com" will match "foo-bar.mycompany.com" but not "fr.mycompany.com".
 - See RFC 2818 and RFC 6125 for more details and security implications of wildcards.
 - Be careful when mixing wildcard FQDN entries with certificates that contain wildcards in their names. For example, the entry "foo*.mycompany.com" will match a certificate with the name "**bar.mycompany.com".
 - This wildcarding mechanism does not work with server certificates that contain the "?" character in their names. (This is not a common occurrence.)
- You can choose any server certificate's issuing certificate authority (CA) from the drop-down list. It could be the direct issuer or any CA at higher level in the certificate chain, up to the root.

Connect Secure or Policy Secure (L3) Connection Type Options

Use a Connect Secure or Policy Secure (L3) connection for a Layer 3 connection to Pulse Connect Secure or Pulse Policy Secure.

- **Allow user to override connection policy**—Allows a user to override the connection policy by manually connecting or disconnecting. Typically, you leave this option selected to make sure that a user can establish a connection under all conditions. If you disable this check box, the user cannot change the endpoint's connection status, suspend/resume a connection to Pulse Connect Secure or shut down Pulse Secure client.
- **Lock-down this connection**—When enabled, this option limits network connectivity while the Pulse desktop client is in the process of creating a VPN connection. When used in conjunction with Location Awareness rules, this option ensures that end users cannot access network resources outside of a VPN tunnel.



Note: Lock-down this connection feature exists in the Pulse Desktop Client for Windows only.

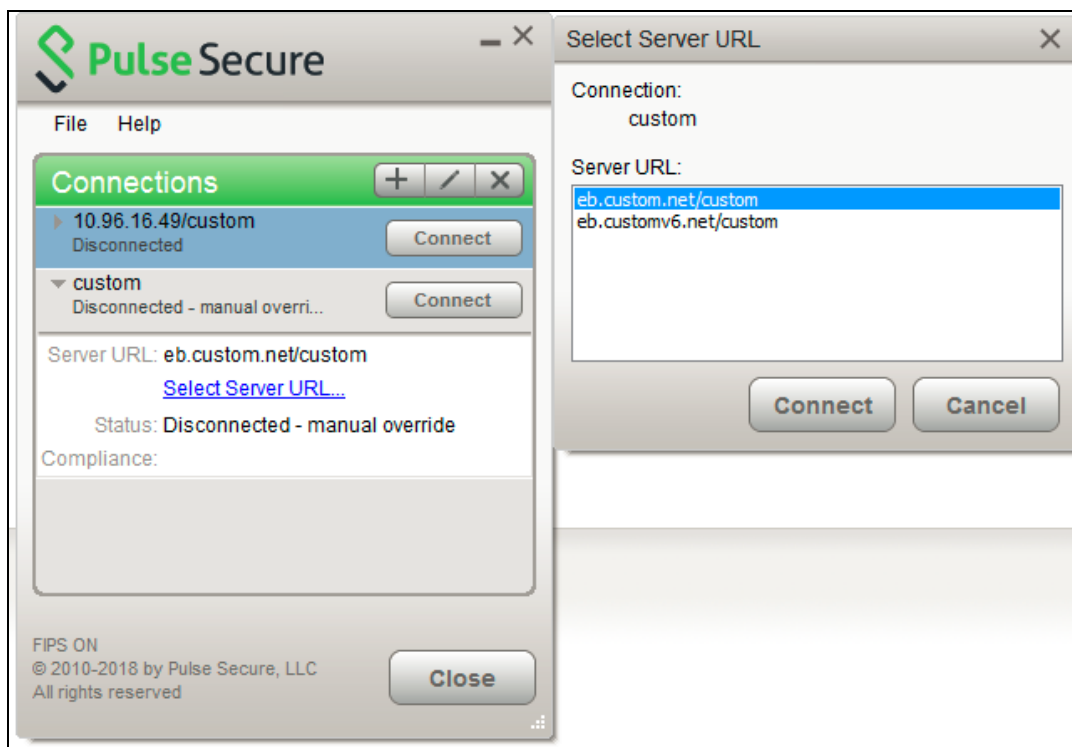
- **Support Remote Access (SSL VPN) or LAN Access (UAC) on this connection**—This option must be selected if this connection is for Pulse Policy Secure. If the connection is for Pulse Connect Secure, you can disable this check box and use the connection for accessing Pulse Collaboration meetings only by also selecting **Enable Pulse Collaboration integration on this connection**.
- **Enable Pulse Collaboration integration on this connection**—This option must be disabled if this connection is for Pulse Policy Secure. If the connection is for Pulse Connect Secure, you can enable this check box and use the connection for accessing Pulse Collaboration meetings.
- **Connect to URL of this server only**—Specifies whether the endpoint connects to this Pulse server exclusively or if it can connect to any of the servers listed in the list of connection URLs. Disable this check box to enable the List of Connection URLs.
- **List of Connection URLs**—Allows you to specify a list of Pulse servers (Pulse Policy Secure or Pulse Connect Secure) for this connection. The Pulse client attempts to reach each server in the list, in the order listed, until it succeeds. You can specify up to 8 Pulse servers.



Note: IF-MAP federation must be configured to ensure that a suspended session can be resumed to a different URL. When this feature is used with Pulse Policy Secure, all of the Pulse servers in the list must be configured for failover, so that any one of them can provision the firewall enforcer.

Figure 87 shows how the Pulse user can select a server from the connection's list of URLs.

Figure 87: Pulse for Windows Client with a List of Connection URLs



- **Attempt most recently connected URL first**—If you have specified a list of connection URLs, you can select this check box to have the Pulse client always attempt the most recent successful connection. If that connection is not successful, Pulse then starts at the top of the list. The most recently connected URL is saved across reboots.
- **Randomize URL list order**—If you have specified a list of connection URLs, select this check box to have the Pulse client ignore the order in which the servers are listed. You can select this option to spread the connection load across multiple Pulse servers.

If you enabled Attempt most recently connected URL first, then Pulse attempts that connection first. If the connection attempt fails, Pulse chooses randomly from the list for the next connection attempt. During a credential provider connection attempt, Pulse chooses the URL automatically. It does not display a window to let the user choose a URL. Connections that use machine authentication ignore this option and always use the ordered list of connection URLs. Any preferred roles and realms you specify must be applicable to all of those servers. In the case of an interrupted connection, such as temporarily losing the WiFi link, Pulse always tries to reconnect to the most recently connected URL.

The Pulse servers should be configured for IF-MAP federation to ensure that a session can be resumed to a different URL. If the Pulse servers are not federated, then Pulse might prompt for credentials.

SRX (for Dynamic VPN) Connection Type Options

Use an SRX connection for a dynamic VPN connection to an SRX Series Services Gateway.

- **Address**—Specifies the IP address of the SRX Series device.
- **Allow user to override connection policy**—Allows users to override the connection policy by manually connecting or disconnecting. Typically, you leave this option selected to make sure that a user can establish a connection under all conditions. If you disable this check box, the user cannot change the endpoint's connection status or shut down Pulse Secure client.

Pulse Connection is Established Options

For all connection types, specify how the connection is established. The options vary according to the type of connection. Automatic connections include machine authentication and credential provider connections. Connections can be established using the following options.



Note: All connections that are configured to start automatically attempt to connect to their target networks at startup time. To avoid multiple connection attempts, be sure that only one connection is configured to start automatically, or configure location awareness rules.

- Modes:
 - **User**—Enables user authentication.
 - **Machine**—Enables machine authentication, which requires that Active Directory is used as the authentication server and that machine credentials are configured in Active Directory. A machine connection is, by default, an automatic connection.
 - **Machine or user**—Enables machine authentication for the initial connection. After user authentication, the machine authentication is dropped. When the user logs out, the machine authentication connection is restored.
- Options:
 - **Connect automatically**—Connections are attempted when the conditions specified in the location awareness rules are true, and disconnected when the conditions are no longer true.
 - **Reconnect at Session Timeout or Deletion**—If this option is enabled, user initiated sessions automatically attempt to reconnect upon a session timeout or deletion. If this option is disabled, then user initiated sessions remain disconnected upon a session timeout or deletion.
 - **Enable pre-desktop login (Credential provider)**—Enables Pulse client interaction with the credential provider software on the endpoint. The user credentials are used to establish the authenticated Pulse connection to the network, login to the endpoint, and login to the domain server.

Pulse Connection is Established Examples

The following configurations show how to select the Connection is established options of a Pulse connection set for specific user login behavior:

Figure 88: Connect manually

♥ Connection is established:

Specify mode: ▼

Options:

☐ Connect automatically

☐ Enable pre-desktop login (Credential provider)

Figure 89: Connect automatically after user signs in to the desktop

♥ Connection is established:

Specify mode: User ▼

Options:

- ☒ Connect automatically
- ☐ Enable pre-desktop login (Credential provider)


Figure 90: Connect automatically when the machine starts; machine credentials are used for authentication

♥ Connection is established:

Specify mode: Machine ▼

Options:

- ☒ Connect automatically
- ☐ Enable pre-desktop login (Credential provider)



Note: When you use machine credentials for authentication and no user credentials, Pulse cannot perform user-based tasks. The following tasks can be run only when the user is logged in:

- Run session scripts
- Detect or modify proxy settings
- Run automatic Pulse client upgrade
- Install or upgrade Pulse components

Figure 91: Connect automatically when the machine starts; the connection is authenticated again when the user signs in to the desktop

♥ Connection is established:

Specify mode: Machine or User ▼

Options:

- ☒ Connect automatically
- ☐ Enable pre-desktop login (Credential provider)

The configuration in Figure 91 enables machine authentication for the initial connection. After the user connects with user credentials, the machine authentication is dropped. When the user logs out, the machine authentication connection is restored. In one typical use case, the machine credentials provide access to one VLAN, and the user credentials provide access to a different VLAN.



Note: If the machine and user have different roles, each role should map to the same connection set. Otherwise, after the user connects, the existing connection set might be replaced.

Figure 92: Connect automatically at user login

✓ Connection is established:

Specify mode: User ▼

Options:

- ☒ Connect automatically
- ☒ Enable pre-desktop login (Credential provider)

The configuration in Figure 92 enables Pulse client interaction with the credential provider software on the endpoint. The user credentials are used to establish the authenticated Pulse connection to the network, to log in to the endpoint, and to log in to the domain server.

Figure 93: Connect automatically when the machine starts; connection is authenticated again at user login

✓ Connection is established:

Specify mode: Machine or User ▼

Options:

- ☒ Connect automatically
- ☒ Enable pre-desktop login (Credential provider)

The configuration in Figure 93 enables Pulse client interaction with the credential provider software on the endpoint. Machine credentials are used to establish the authenticated Pulse connection to the network. When the user provides user credentials, the connection is authenticated again. In one typical use case, the machine credentials provide access to one VLAN, and the user credentials provide access to a different VLAN.

Location Awareness Rules

For Connect Secure or Policy Secure (L3) and SRX connections, you can define location awareness rules that enable an endpoint to connect conditionally. If you do not have location awareness rules defined, Pulse attempts to connect with each connection that is defined as an automatic connection until it connects successfully. Location awareness rules allow you to define an intelligent connection scheme. For example, the endpoint connects to Pulse Policy Secure if it is connected to the company intranet, or it connects to Pulse Connect Secure if it is in a remote location.

A Pulse connection uses the IP address of a specified interface on the endpoint to determine its network location. Each location awareness rule includes the following settings:

- **Name**—A descriptive name, for example, “corporate-DNS.” A name can include letters, numbers, hyphens, and underscores.
- **Action**—The method the connection uses to discover the IP address. Choose one of the following values:
 - **DNS Server**—Allows the endpoint to connect if the endpoint’s DNS server on the specified interface is set to one of the specified values. Use the Condition box to specify IP addresses or address ranges.

- **Resolve Address**—Allows the endpoint to connect if the hostname specified in the DNS Name box can be resolved by the DNS server for the specified interface. If one or more address ranges are specified in the Address Range box, the address must resolve to one of the ranges to satisfy the expression.
- **Endpoint Address**—Allows the endpoint to connect if the IP address of the specified interface is within a range specified in the IP Address Range box.



Note: To create a negative location awareness rule, you first create the positive state and then use rule requirement logic to use the rule as a negative condition.

Machine Connection Preferences

The Machine Connection Preferences appear if you have selected one of the machine authentication options for how the connection is established. Normally Pulse presents a selection dialog box if more than one realm or role is available to the user. However, a connection that is established through machine authentication fails if a dialog box is presented during the connection process. To suppress the selection dialogs, either ensure that only one role and realm is available to users, or specify a preferred realm and role for this connection.

- **Preferred Machine Realm**—Specify the realm that this connection uses when establishing the machine connection. The connection ignores any other realm available for the specified login credentials
- **Preferred Machine Role Set**—Specify the role or the name of rule for the role set that this connection uses when establishing the machine connection. The role or rule name used must be a member of the preferred machine realm.

User Connection Preferences

The User Connection Preferences options enable you to specify a realm and role for automatic connections that would otherwise present a selection dialog box to the user. To suppress the selection dialogs, either ensure that only one role and realm is available to users, or specify a preferred realm and role for this connection.

- **Preferred User Realm**—Specify the realm for this connection that is used when a user logs onto the endpoint. The connection ignores any other realm available for the user's login credentials

If one of the credential provider connection options is enabled, the following options are available:

- Preferred Smartcard Logon Realm—Preferred realm to be used when user logs in with a smart card.
- Preferred Password Logon Realm—Preferred realm to be used when user logs in with a password.



Note: Be sure that the authentication realms you specify exist, and that they support the appropriate login credential option.

- **Preferred User Role Set**—Specify the preferred role or the name of the rule for the role set to be used for user authentication. The role or rule name used must be a member of the preferred user realm.
- **Select client certificate from machine certificate store**—Enables you to specify the location of the client certificate on a Windows endpoint as part of a Pulse connection that verifies the identity of both the machine and the user before establishing a connection. When this check box is selected, the Pulse connection looks at client certificates located in the Local Computer personal certificate store. When this check box is not selected, the connection accesses the user certificate store as a Windows endpoint. For more information, see [“Machine and User Authentication Through a Pulse Connection for Pulse Connect Secure”](#).

Related Documentation

- [Pulse Connect Secure Overview](#)
- [Creating a Client Connection Set for Pulse Connect Secure](#)

- [Machine and User Authentication Through a Pulse Connection for Pulse Connect Secure](#)

Securing the Connection State on the Pulse Secure Client

To disable user interaction with Pulse connections on the endpoint, you can configure Pulse Secure Connections so that when they are deployed to the endpoint, users cannot shut down a connection, suspend or resume a connection, or shut down Pulse. Disabling user interaction with Pulse enables the Pulse administrator to control how endpoints connect to the network without allowing the user to override administrative control. For example, if you use machine authentication, the connection from endpoint to server is established automatically. By locking down the Pulse endpoint, users cannot change their connection.

To secure the Pulse endpoint:

1. Click **Users > Pulse Secure Connections**.
2. Edit or create a new connection.
3. Disable the check box labeled **Allow user to override connection policy**.

Related Documentation

- [Client Connection Set Options for Pulse Policy Secure](#)
- [Endpoint Security Monitoring and Management for Pulse Policy Secure](#)
- [Machine Authentication for Pulse Policy Secure Overview](#)

Creating a Client Connection Set for Pulse Connect Secure

To create a Pulse Secure Connection:

1. From the admin console, select **Users > Pulse Secure > Connections**.
2. Click **New**.
3. Enter a name and, optionally, a description for this connection set.



Note: You must enter a connection set name before you can create connections.

4. Click **Save Changes**.
5. From the main Pulse Secure Connections page, select the connection set.
6. Under Options, select or clear the following check boxes:
 - **Allow saving login information**—Controls whether the Save Settings check box is available in login credential dialog boxes in the Pulse Secure client. If you clear this check box, the Pulse Secure client always requires users to provide credentials. If you select this check box, users have the option of saving their credentials.
 - **Allow user connections**—Controls whether connections can be added by the user through the Pulse client interface.
 - **Display splash screen**—Clear this check box to hide the Pulse splash screen that normally appears when the Pulse client starts.
 - **Dynamic certificate trust**—Determines whether users can opt to trust unknown certificates. If you select this check box, a user can ignore warnings about invalid certificates and connect to the target Pulse server.



Note: By default, **Dynamic certificate trust** check box will be unchecked.

- **Dynamic connections**—Allows new connections to be added automatically to a Pulse Secure client when the user logs into a Pulse server through the server's Web portal, and then starts Pulse through the Web portal interface.

- **FIPS mode enabled**—Enable FIPS mode communications for all Pulse connections in the connection set. The Federal Information Processing Standard (FIPS) defines secure communications for the U.S. government. When a Pulse connection is operating in FIPS mode, **FIPS On** appears in the lower corner of the Pulse client interface.



Note: If the Pulse server hardware does not support FIPS mode operations, FIPS mode configuration options are not present in the admin console interface. FIPS mode operations are supported on PSA-V Series Pulse Secure Gateways and some SA series appliances. The device must be running Pulse Policy Secure R5.0 or later or Pulse Connect Secure R8.0 or later.



Note: Users cannot enable FIPS mode for connections that are created on the client. You must deploy connections with FIPS mode enabled using a pre-configured connection set with FIPS mode enabled or have users establish a browser session to the FIPS enabled Pulse server.

- **Wireless suppression**—Disables wireless access when a wired connection is available. Wireless suppression occurs only when the wired connection is connected and authorized.
7. Under Connections, click **New** to define a new connection.
 8. Enter a name and, optionally, a description for this connection.
 9. Select a type for the connection and then specify the connection. Type can be any of the following:
 - **Policy Secure (802.1X)**—Select this type if the connection establishes connectivity to an 802.1X wired or wireless device.
 - **Connect Secure or Policy Secure (L3)**—Select this type to define a connection for Pulse Connect Secure or Pulse Policy Secure.
 - **SRX**—Select this type to define a connection to an SRX Series Services Gateway.
 10. The connection configuration options that appear depend on the connection type you select.

After you have created the client connection set, create a client component set and select this connection set.

Related Documentation

- [Component Set Options for Pulse Connect Secure](#)
- [Endpoint Security Monitoring and Management for Pulse Connect Secure](#)
- [Configuring Location Awareness Rules for Pulse Secure Client](#)

Pulse Secure Client FIPS Mode for Pulse Connect Secure Overview

The United States Federal Information Processing Standard (FIPS) defines security and interoperability requirements for computer systems that are used by the U.S. government. Pulse Secure for Windows, Mac, iOS (32-bit iOS devices only), and Android support FIPS mode operations when communicating with Pulse Connect Secure and Pulse Secure for Windows and Mac support FIPS mode operations when communicating with Pulse Policy Secure. When it is operating in FIPS mode, **FIPS On** appears in the bottom corner of the Pulse for Windows and Mac clients.

If the Pulse server hardware does not support FIPS mode operations, FIPS mode configuration options are not present in the admin console interface. FIPS mode operations are supported on PSA-V Series Pulse Secure Gateways and some SA series appliances. The device must be running Pulse Policy Secure R5.0 or later or Pulse Connect Secure R8.0 or later.

You enable FIPS mode operations when you configure Pulse connections on the server. You enable FIPS mode operations for a connection set. That connection set can include any or all of the four types of Pulse connections:

- **Policy Secure (802.1X)**—Pulse client uses FIPS mode cryptography for authentication but it uses default Microsoft cryptography for the WEP/WPA wireless encryption.
- **Connect Secure or Policy Secure (L3)**—FIPS mode cryptography is supported.
- **SRX**—FIPS mode cryptography is not supported.



Note: Users cannot enable FIPS mode for connections that are created on the client. You must deploy connections with FIPS mode enabled using a pre-configured connection set with FIPS mode enabled or have users establish a browser session to the FIPS enabled Pulse server.

Endpoint Requirements

Pulse supports FIPS mode on Windows Vista and later Windows versions. FIPS is not supported by the Pulse OS X client. To support client certificate private key operations, the security policy on the endpoint must have FIPS enabled. To verify that FIPS is enabled, use the Microsoft Management Console (MMC). Make sure that the Group Policy Snap-in is installed, and then open the following item:

Local Computer Policy > Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options

Scroll through the Policy list and make sure that the following policy is enabled:

System cryptography: Use FIPS compliant algorithms for encryption, hashing and signing

Configuration Overview

The Pulse client includes all components required for FIPS mode communications. To enable FIPS mode communications, deploy one or more Pulse connections to the client that are FIPS enabled. [Figure 94](#) shows the check box in the Pulse connection set configuration screen that enables FIPS mode operations for all connections in the connection set.

Figure 94: Enabling FIPS Mode for Pulse Connections

Pulse Secure Client > Connections > New Connection Set

New Connection Set

Name:

Description:

Owner:

Last Modified: 2018-05-09 05:48:23 UTC

Server ID: 0312MVD4A0EM704VS

▼ Always-on vpn wizard

[Configure Always-on VPN using wizard](#)

▼ Options

Name	Value
Always-on Pulse Client Prevents end users from circumventing Pulse connections. This option will disable all configuration settings that allow the end user to disable or remove Pulse connections, services or software.	<input type="checkbox"/>
VPN only access When Pulse client connects to a PCB having lock down mode enabled, it will enter lock-down mode and won't let any traffic flow through unless a Locked-down VPN connection is in connected state. User is allowed to connect or disconnect any connection. User is allowed to add any new connection/server URI. User is allowed to delete a connection if the connection is not locked down.	<input type="checkbox"/>
Allow saving logon information Enables the Save settings checkbox in the certificate trust and password prompts.	<input checked="" type="checkbox"/>
Allow user connections Allows user to create connections via the Pulse UI.	<input checked="" type="checkbox"/>
Display Splash Screen Controls whether the splash screen is displayed when Pulse starts.	<input checked="" type="checkbox"/>
Dynamic certificate trust Controls whether users may accept to trust unknown certificates.	<input checked="" type="checkbox"/>
Dynamic connections Allows connections to be deployed automatically from devices.	<input checked="" type="checkbox"/>
EAP Fragment Size Maximum number of bytes in an EAPoL message from the client for 802.1x connections. Range: 450 - 3000 bytes	<input type="text" value="1400"/>
Enable captive portal detection Pulse will attempt to detect the presence of a captive portal hotspot. Only applies to Connect Secure and Policy Secure (L3) connections.	<input type="checkbox"/>
Enable embedded browser for captive portal Pulse will use an embedded web browser for captive portal pages. Only applies when captive portal detection is enabled.	<input checked="" type="checkbox"/>
Enable embedded browser for authentication Pulse will use embedded browser for saml, custom sign-in or token based authentication.	<input type="checkbox"/>
FIPS mode enabled Deploy client with Federal Information Processing Standard enabled.	<input checked="" type="checkbox"/>
Wireless suppression Disconnect all wireless interfaces when a wired interface gets connected to a network. Applies to all wireless connections (not just those managed by Pulse).	<input type="checkbox"/>
Prevent caching smart card PIN Enabling this will ensure the smart card PIN value is not cached by the client process.	<input type="checkbox"/>



Note: If the Pulse server hardware does not support FIPS mode operations, FIPS mode configuration options are not present in the admin console interface. FIPS mode operations are supported on PSA-V Series Pulse Secure Gateways and some SA series appliances. The device must be running Pulse Policy Secure R5.0 and later or Pulse Connect Secure R8.0 and later.

Related Documentation

- [Creating a Client Connection Set for Pulse Connect Secure](#)

Configuring Location Awareness Rules for Pulse Secure Client

The location awareness feature enables a Pulse Secure client to recognize its location and then make the correct connection. For example, you can define rules so that a Pulse client that is started in a remote location automatically establishes a VPN connection to Pulse Connect Secure, and then that same client automatically connects to Pulse Policy Secure when it is started in the corporate office. If Pulse detects that it is connected to the corporate LAN and it already has a VPN connection (for example, the VPN connection was suspended when the computer was put into hibernation), it first discovers that the VPN location awareness rules are no longer true, disconnects that VPN connection, and then evaluates the location awareness rules for the other configured connections.

Location awareness relies on rules you define for each Pulse connection. If the conditions specified in the rules resolve to TRUE, Pulse attempts to make the connection. If the conditions specified in the rules do not resolve to TRUE, Pulse tries the next connection. To set up the location awareness rules that select among many connections, you must define location awareness rules for each connection. Each location awareness rule is based on the endpoint's ability to reach an IP address or resolve a DNS name over a specified network interface.

The following location awareness example includes two connections. Each connection is configured to connect to only one target server. The first connection is a Pulse Policy Secure connection that resolves to TRUE when the endpoint is connected to the corporate LAN. The second connection is a Pulse Connect Secure connection that resolves to TRUE when the endpoint is located in a remote location. If Pulse detects that it is connected to the corporate LAN and it already has a VPN connection, it disconnects that VPN connection.

Pulse Policy Secure connection

If the DNS server that is reachable on the endpoint's physical network interface is one of your organization's internal DNS servers, then establish the connection.

Pulse Connect Secure connection

If the DNS server that is reachable on the endpoint's physical network interface is not one of your organization's internal DNS servers, and the DNS name of your Pulse Connect Secure device resolves to the external facing IP address of the Pulse Connect Secure device, then establish the connection.



Note: Connections can be set to manual, automatic, or controlled by location awareness rules. When the user logs in, the Pulse client attempts every connection in its connections list that is set to automatic or controlled by location awareness rules.



Note: To create a negative location awareness rule, you first create the positive state and then use rule requirement logic to use the rule as a negative condition.

To configure location awareness rules:

1. If you have not already done so, create a connection or open an existing connection.
You can configure location awareness rules for SRX connections and Connect Secure or Policy Secure (L3) connections. Location awareness rules do not apply to UAC (802.1X) connections.
2. Click the Mode list, and then select one of the options, **User**, **Machine**, or **Machine or user**.
3. If you selected **User** as the Mode, Under Options, select **Connect automatically**. If you selected **Machine** or **User or Machine**, **Connect automatically** is enabled by default.
4. Under Location awareness rules, click **New**.

Alternatively, you can select the check box next to an existing rule, and then click **Duplicate** to create a new rule that is based on an existing rule.

5. Specify a name and description for the rule.
6. In the Action list, select one of the following:
 - **DNS server**—Connect if the DNS server associated with the endpoint's network properties is (or is not) set to a certain value or set of values. Specify the DNS server IP address in the IP address box. Also specify a network interface on which the condition must be satisfied:
 - **Physical**—The condition must be satisfied on the physical interfaces on the endpoint.
 - **Pulse Secure**—The condition must be satisfied on the virtual interface that Pulse Secure client creates when it establishes a connection.
 - **Any**—Use any interface.
 - **Resolve address**—Connect if the configured hostname or set of hostnames is (or is not) resolvable by the endpoint to a particular IP address. Specify the hostname in the DNS name box and the IP address or addresses in the IP address box. Also specify a network interface on which the condition must be satisfied.



Note: The Pulse client software evaluates IP and DNS policies on network interface changes. DNS lookups occur on DNS configuration changes or when the time-to-live setting (10 minutes) expires for a particular host record. If Pulse cannot resolve the host for any reason, it polls the configured DNS server list every 30 seconds. If the host had been resolved successfully previously and the time-to-live timer has not expired, the polling continues until the timer expires. If the host had not been resolved successfully previously, the resolution attempt fails immediately.

- **Endpoint Address**—Connect if a network adapter on the endpoint has an IP address that falls within or outside of a range or a set of ranges. Specify the IP address or addresses in the IP address box. Also specify a network interface on which the condition must be satisfied.

7. Click **Save Changes**.

After you create the rule or rules, you must enable each rule you want to use for the connection. To enable a negative form of a rule, use a custom version of the rule. To enable location awareness rules:

1. In the list of connection awareness rules for a connection, select the check box next to each rule you want to enable.
2. To specify how to enforce the selected location awareness rules, select one of the following options:
 - **All of the above rules**—The condition is TRUE and the connection is attempted only when all selected location awareness rules are satisfied.
 - **Any of the above rules**—The condition is TRUE and the connection is attempted when any select location awareness rule is satisfied.
 - **Custom**—The condition is TRUE and the connection is attempted only when all selected location awareness rules are satisfied according to the Boolean logic you specify in the Custom box. Use the Boolean condition to specify a negative location rule. For example, connect to Pulse Connect Secure when Rule-1 is false and Rule-2 is true. The Boolean logic in the custom box would be: **NOT Rule-1 AND Rule-2**. The accepted Boolean operators are AND, OR, NOT, and the use of ().

3. Click **Save Changes**.

Related Documentation

- [Understanding Session Migration](#)

Component Set Options for Pulse Connect Secure

A Pulse Secure client component set includes specific software components that provide Pulse Secure client connectivity and services.



Note: Client component set options affect Web-based installations only.

- **All components**—Supports all Pulse connection types.
- **No components**—Updates existing Pulse client configurations, for example, to add a new connection. Do not use this setting for a new installation.

Related Documentation

- [Creating a Client Connection Set for Pulse Connect Secure](#)
- [Manage Pulse Secure Client Versions](#)

Creating a Client Component Set for Pulse Connect Secure

A Pulse Secure client component set includes specific software components that provide Pulse Secure client connectivity and services.



Note: Client component set options affect Web-based installations only.

To create a Pulse client component set:

1. From the admin console, select **Users > Pulse Secure > Components**.
2. Click **New** to create a new component set.
3. If you have not yet created a client connection set, select **Users > Pulse Secure > Connections** and create a new connection set. Or you can use the default client configuration, which permits dynamic connections, supports the outer username anonymous, and allows the client to automatically connect to Pulse Policy Secure or Pulse Connect Secure.
4. Specify a name for the client component set.
5. (Optional) Enter a description for this client component set.
6. Select a connection set that you have created, or use the default connection set.
7. For Pulse Secure client components, select one of the following options:
 - **All components**—Supports all Pulse connection types.
 - **No components**—Updates existing Pulse client configurations, for example, to add a new connection. Do not use this setting for a new installation.
8. Click **Save Changes**.
9. After you create a component set, distribute the client to users through a role. When users access the role, the installer automatically downloads to the endpoint. The installer components and connections are applied to the endpoint client.

If client connections associated with the component set for a role are changed even though the list of components has not, the existing configuration on the endpoint is replaced immediately if the endpoint is currently connected, or the next time the endpoint connects.

If a user is assigned to multiple roles and the roles include different component sets, the first role in an endpoint's list of roles is the one that determines which client (component set) is deployed.

Related Documentation

- [Component Set Options for Pulse Connect Secure](#)
- [Creating a Client Connection Set for Pulse Connect Secure](#)
- [Manage Pulse Secure Client Versions](#)

Manage Pulse Secure Client Versions

This feature is supported for Pulse Desktop Clients on macOS and Windows. From 9.1R2 release onwards, this feature is supported for iOS Mobile Client and Android Mobile Client also.



Note: From 9.1R2 onwards, this feature will not be applicable for Windows Universal App.

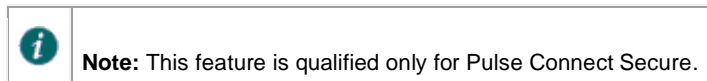
This feature allows admin to configure a minimum client version. If the client has version lower than the configured minimum version, then the PCS server will reject the client connection.

If the client is older than the configured minimum client version, then Pulse Connect Secure gateway will reject the connection. User can upgrade it later through browser or SCCM server.

For example, the gateway can host Pulse Desktop 5.2r3, but the minimum version could be Pulse Desktop 5.2r2.

Similarly, 5.2r2 client is connected to the gateway and it would prompt for an upgrade, if active version is greater than the client version.

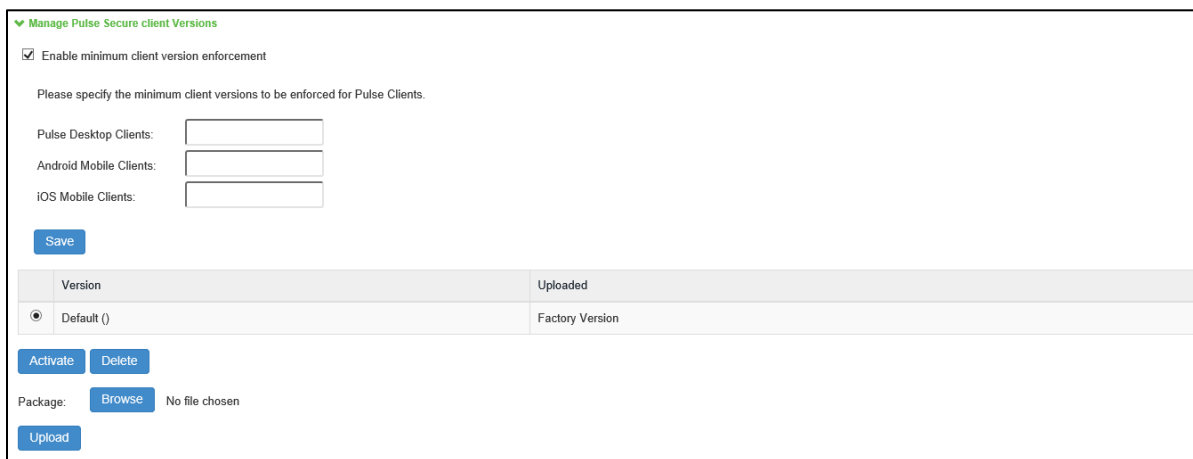
For 5.2r1 client, the connectivity would be rejected and will display an appropriate error message.



To enable this feature on Pulse Connect Secure, perform the following steps:

1. From the admin console, select **Users > Pulse Secure > Components**.
2. Select the **Enable minimum client version enforcement** check box, the following options appear:
 - Pulse Desktop Clients
 - Android Mobile Clients
 - iOS Mobile Clients

Figure 95: Minimum Client Version Enforcement



▼ Manage Pulse Secure client Versions

☒ Enable minimum client version enforcement

Please specify the minimum client versions to be enforced for Pulse Clients.

Pulse Desktop Clients:

Android Mobile Clients:

iOS Mobile Clients:

Version	Uploaded
<input checked="" type="radio"/> Default ()	Factory Version

Package: No file chosen

3. Specify the minimum client version to be enforced for the required client.
If minimum client version is not specified for any specific type of client, then any version of the client will be allowed to connect to the PCS without any minimum client version enforcement for that specific client.
4. Click **Save**.
5. Select Version to Activate/Delete the version details.
6. Click **Browse** to select a Pulse Secure Client package.
7. Click **Upload** to add packages.



Note: Pulse Secure Client allows you to add up to three packages on the server at a time, to add new packages you need to delete the existing package.



Note:

- Minimum Client Version Enforcement feature is not supported on Linux clients. Enforcement will not be applicable when Linux Pulse desktop client connects to PCS which has minimum client version enforcement enabled.
- Minimum Client Version Enforcement feature is not supported by PPS.
- Minimum Client Version Enforcement feature is not supported for the client session launched from a user browser session.

Endpoint Security Monitoring and Management for Pulse Connect Secure

You can configure and enable Host Checker policies to perform an endpoint security assessment before allowing the endpoint to connect. Host Checker is supported on the following operating systems:

- Windows (including 8.1 and later versions of Windows RT and Windows Phone)
- macOS
- Google Android
- Apple iOS

You can invoke Host Checker at the role level or the realm level to specify access requirements for endpoints seeking authentication. Host Checker policies that are implemented at the realm level occur before the user is authenticated. Host Checker policies at the role level are implemented after authentication but before the user is permitted to access protected resources. When an endpoint first connects to Pulse Connect Secure, the latest version of the IMC is downloaded to the host computer. The initial check can take 10-20 seconds to run. Outdated IMC files are automatically updated at subsequent checks.



Note: The first time an endpoint connects to Pulse Connect Secure that has a patch assessment policy, if the connection is a Layer 2 connection, the IMC cannot download. In this case, you should configure a remediation role that displays instructions to direct the user to retry with a Layer 3 connection or to contact the administrator.



Note: If a realm has a Host Checker policy enabled that is for desktop clients, and a mobile device user employs a browser on the mobile device to connect to the Web portal, the login is denied because the desktop Host Checker program is not compatible with the mobile client OS. If Pulse mobile users are mapped to multiple roles, the login operation assigns them to a role where Host Checker is not enabled if possible. If all the roles have Host Checker enabled, the mobile users will not be allowed to login from the browser. You can create and enable Host Checker policies that are specific to each mobile operating system and then Host Checker runs when the Pulse client connects to the server.

For patch management on Windows systems, Host Checker uses a list of the most current patch versions from the vendor for predefined rules in the Host Checker policy. Host Checker does not scan for non-security patches. Server and Host Checker manage the flow of information between the corresponding pairs of TNC-based integrity measurement collectors (IMCs) and integrity measurement verifiers (IMVs). IMCs are software modules that run on the endpoint and collect information such as antivirus, antispyware, patch management, firewall, and other configuration and security information

about the host. IMVs are software modules that run on the server and verify a particular aspect of a host's integrity. Each IMV works with the corresponding IMC on the client endpoint to verify that the endpoint meets the Host Checker rules. IMCs scan the endpoint frequently for changes in security status. For example, if the user turns off virus checking, the IMC can detect this and then trigger a new check to make sure the modified system complies with the requirements of the Host Checker policy. You can configure Host Checker to monitor third-party IMCs installed on client computers by using third-party IMVs that are installed on a remote IMV server.

You obtain the most current patch version information from a Pulse Secure staging site. You can manually download and import the list into the Pulse Connect Secure server, or you can automatically import the list from the Pulse Secure staging site or your own staging site at a specified interval.

Monitoring is based on one or more specified products or on specific patches, though not in the same policy. For example, you could check for Internet Explorer Version 7 with one policy, and Patch MSOO-039: SSL Certificate Validation Vulnerabilities with a second policy. Then, apply both policies to endpoints at the role or realm level to ensure that the user has the latest browser version with a specific patch. In addition, for Microsoft products, you can specify the severity level of patches that you want to ignore. For example, you could ignore low or moderate threats.

Remediation Options

Host Checker can identify issues on an endpoint. However, Host Checker and Pulse Connect Secure cannot resolve issues, that is, perform remediation tasks, on non-compliant endpoints. To repair those issues Pulse Connect Secure supports the following remediation options:

- Instructions to the user— The Pulse Secure gateway can send a message to the user describing the non-compliant patches or software and a link to where the user can obtain the required software. [Figure 96](#) shows a typical Pulse remediation message.

Figure 96: Pulse Remediation Instructions



- Initiate SMS/SCCM remediation—For remediation using Microsoft System Center Configuration Manager (ConfigMgr or SCCM), formerly Systems Management Server (SMS), a preinstalled SMS/SCCM client on the endpoint is triggered by Host Checker to get patches from a preconfigured SMS/SCCM server. This mechanism installs only those patches that are published on the SMS/SCCM server.

Related Documentation

- [Issuing a Remediation Message with Pulse Connect Secure](#)
- [Host Checker Statement of Health for Pulse Connect Secure Overview](#)

Issuing a Remediation Message with Pulse Connect Secure

If a Host Checker policy finds that an endpoint is not in compliance, Host Checker can display a message through the Pulse client interface that includes custom instructions and reason strings on how to bring the endpoint into conformance. The user must perform the steps described in the message before the endpoint is allowed to access protected resources.

To enable a remediation message for a Host Checker policy:

1. In the admin console, select **Authentication > Endpoint Security > Host Checker**.
2. In the **Policies** section, click **New** to create a new Host Checker policy.
3. As part of the Host Checker Policy, select **Enable Custom Instructions**.

When you select this option, a text box appears. Enter the instructions to display to the user on the Host Checker remediation page. You can use the following HTML tags to format text and to add links to resources such as policy servers or web sites: `<i>`, ``, `
`, ``, and `<a href>`. For example:

You do not have the latest signature files.

`Click here to download the latest signature files.`

4. Optionally, select **Send reason strings**. Select this option to display a message to users (called a reason string) that is returned by Host Checker or IMV and that explains why the client machine does not meet the Host Checker policy requirements. Reason strings describe to users what the IMV is checking on the client endpoint. This option applies to predefined rules, to custom rules, and to third-party IMVs that use extensions in the Pulse Secure TNC SDK.
5. Click **Save Changes**.

Be sure to include the Host Checker policy in the realm or role you configure for Pulse users.

Related Documentation

- [Using SMS/SCCM Remediation with Pulse Connect Secure](#)

Using SMS/SCCM Remediation with Pulse Connect Secure

Pulse Secure client supports the Microsoft System Center Configuration Manager (ConfigMgr or SCCM), formerly Systems Management Server (SMS) download method for patch deployment. If Pulse Connect Secure is configured for the SMS/SCCM method for patch deployment, the Pulse client endpoint must have the SMS/SCCM client already installed on the endpoint, otherwise remediation fails.

Endpoints configured with SMS/SCCM for software management typically poll the server for updates every fifteen minutes or longer. In a worst-case scenario, clients that are not in compliance with existing Host Checker software requirements might have to wait until the next update interval to login. Using the SMS/SCCM download method, you can force the client to initiate the software update immediately after the patch assessment check. If a user attempts to log in, and the endpoint does not have a required software version for compliance with a Host Checker patch assessment policy, Host Checker immediately notifies the client to poll the server for an immediate update. The client receives notification that an SMS/SCCM update has started.

To configure SMS/SCCM to update the client when notified, set the advertisement time on the SMS/SCCM to As soon as possible.

You assign clients to a particular group or collection on the SMS/SCCM server and then server can advertise patches for that collection. You can configure roles that correspond to collections and SMS/SCCM can send the appropriate patches for a particular role.

You must have the SMS/SCCM client installed and configured correctly on endpoints, and the SMS/SCCM server must be reachable. In a Layer 2 network, Host Checker is performed before the endpoint is connected to the network. Host Checker can obtain the IP address of the SMS/SCCM server configured for the client. If the endpoint is out of compliance and remediation is necessary, Host Checker pings the server IP address every 15 seconds until the server can be notified to update the client.

You should inform users of the expected behavior if this feature is enabled, as there is no notification to the user until the SMS/SCCM sends back the advertisement.

To enable SMS/SCCM assessment and remediation:

1. In the admin console, select **Authentication > Endpoint Security > Host Checker**.
2. In the **Policies** section, click **New** to create a new Host Checker policy.
3. Under Patch Remediation Options, select **SMS/SCCM Patch Deployment**.
4. Click **Save Changes**.

Be sure to include the Host Checker policy in the realm or role you configure for Pulse users.

Related Documentation

- [Issuing a Remediation Message with Pulse Connect Secure](#)

Pushing Pulse Configurations Between Pulse Servers of the Same Type

You can use the Push Configuration feature to centrally manage Pulse Secure Connections, components, and uploaded Pulse packages. The Push Configuration feature enables you to copy all configuration settings or selected configuration settings from one Pulse server to another Pulse server of the same type, for example, from one Pulse Connect Secure server to another Pulse Connect Secure server.

The following notes apply to pushing configurations:

- You can push to a single Pulse server or to multiple Pulse servers in one operation. You can push up to 8 targets per push operation. You can run up to 25 push operations simultaneously. The maximum number of targets is 200. If a push to a target Pulse server fails, the operation proceeds to the next target until all identified targets are updated. The results page displays the status and any problems encountered during the process.
- You can push to a Pulse server that is a member of a cluster as long as the target Pulse server is not a member of the same cluster as the source.
- Target Pulse servers can refuse pushed configuration settings. The default is to accept.
- After an update, the target Pulse server restarts its services. Brief interruptions might occur while the service restarts. We recommend that you push to targets when they are idle or when you can accommodate brief interruptions.
- Target Pulse servers do not display a warning message when they receive a pushed configuration.
- The target Pulse server automatically logs out administrators during the push process.
- The source and target Pulse servers must have the same build version and number.
- The administrator account on the source Pulse server must sign in to the target Pulse server without any human intervention. For example, you cannot have dynamic credentials or multiple roles that are not merged as these both require manual interaction.

Before you use Push Configuration, you must configure your system according to the following conditions:

- You must map to the .Administrators role, thereby creating a “super administrator” with full administration privileges. Modify Authentication > Auth Servers > Administrator Server > Users settings to add yourself to the .Administrators role.
- The target Pulse server administrator account must use static password authentication or two-factor tokens that do not use challenge/response type authentication. For example, certificates, Soft ID, and Defender Authentication are not supported. Modify Administrators > Admin Realms > [Administrator Realm] > General settings to select the proper authentication server for the administrator realm.
- Do not configure the administrator account in a way that requires the administrator to select a role to sign in to the target Pulse server. For example, do not map a single user to multiple roles, including the Push Configuration administrator role, and then fail to merge those roles. We recommend creating an account exclusively for Push Configuration administrators to guarantee that the administrator does not need to choose a role during the sign-in process and to clearly distinguish the actions of Push Configuration administrators in your log files. Use the Administrators > Admin Realms > [Administrator Realm] > Role Mapping settings to set the appropriate role-mapping rules.

To push Pulse configurations from one Pulse server to other Pulse servers of the same type:

1. If you have not already done so, define the targets by selecting **Maintenance > Push Config > Targets**.
2. From the admin console, select **Maintenance > Push Config > Push Configuration**.
3. In the What to push box, select **Selected configuration** to display the configuration categories.
4. Scroll down the list and expand the item labeled Pulse Secure.
5. Select the **Select All Configurations** check box to push all Pulse Secure client configurations on this Pulse server. Or chose none, all, or selected items from the following categories:
 - **Pulse Secure Connections**—Connection sets and connections.
 - **Pulse Secure Components**—Component sets.
 - **Pulse Secure Versions**—Pulse packages that were uploaded to the Pulse server.

6. Add the targets to the **Selected Targets** box.
7. Click **Push Configuration**.

Related Documentation

- [Enabling or Disabling Automatic Upgrades of the Pulse Secure Client](#)

Enabling or Disabling Automatic Upgrades of the Pulse Secure Client

After you deploy Pulse Secure client software to endpoints, software updates occur automatically. If you upgrade the Pulse client configuration on your Pulse server, updated software components are pushed to a client the next time it connects.



Note: If you configure Pulse Secure client to make 802.1X based connections, a reboot might be required on Windows endpoints when Pulse is upgraded.



Note: A bound endpoint receives connection set options and connections from its binding server, but it can have its Pulse client software upgraded from any Pulse server that has the automatic upgrade option enabled. During a client software upgrade the client loses connectivity temporarily.

Pulse client software upgrades are enabled by default. To change the behavior of Pulse client upgrades:

1. From the admin console, select **Maintenance > System > Options**.
2. Set or clear the **Enable automatic upgrade of Pulse Secure Clients** check box.
3. Click **Save Changes**.

Related Documentation

- [Upgrading Pulse Secure Client](#)

Upgrading Pulse Secure Client

The software image for each supported Pulse server includes a Pulse Secure client software package. When a newer version of Pulse is available, you can upload the new software to the Pulse server. You can have more than one version of Pulse on a Pulse server but only one Pulse client package can be active. If you activate a new version of Pulse, and if the Pulse server's automatic upgrade option is enabled, connected Pulse clients display an upgrade prompt to the user. The user can choose to install the upgrade or cancel the operation. If a user cancels, the upgrade prompt appears each time the client connects to the server. During a client software upgrade the Pulse client loses connectivity temporarily.

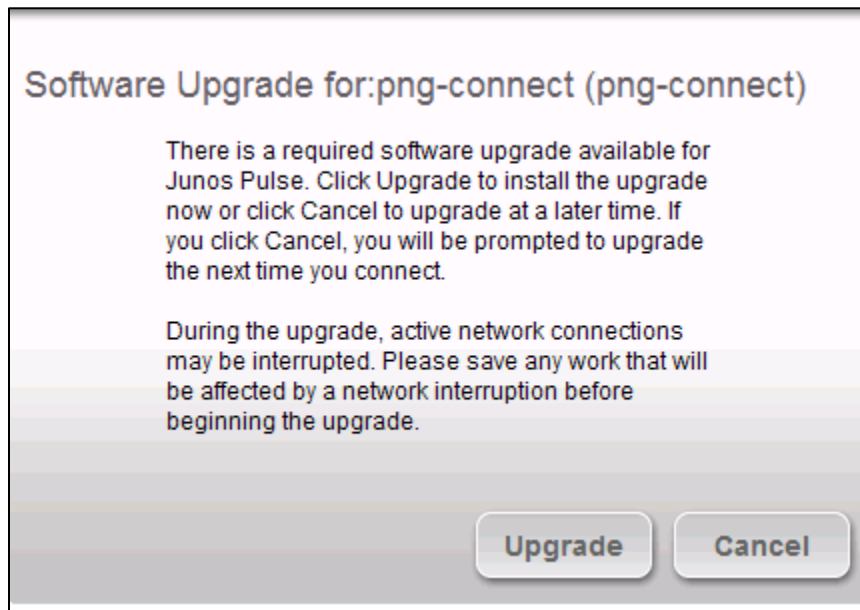


Note: The automatic update feature is supported on Pulse Connect Secure and Pulse Policy Secure servers only. SRX gateways do not support automatic Pulse software updates.



Note: If you configure Pulse to make 802.1X based connections, a reboot might be required on Windows endpoints when Pulse is upgraded.

Figure 97: Pulse Client Upgrade Message



After you have staged the new Pulse software package in a location accessible to the Pulse server, use the following procedure to upload the software to the Pulse server:

1. In the device admin console, select **Users > Pulse Secure > Components**.
2. In the section labeled Manage Pulse Secure Client Versions, click **Browse**, and then select the software package.
3. Click **Upload**.

Only one Pulse Secure client package can be active at a time. After you upload a new package, you need to enable it.

To enable a Pulse package as the default:

1. In the admin console, select **Users > Pulse Secure > Components**.
2. In the section labeled Manage Pulse Secure Client Versions, select the radio button next to a version, and then click **Activate**.

Related Documentation

- [Enabling or Disabling Automatic Upgrades of the Pulse Secure Client](#)

Pulse Collaboration Suite Overview

Pulse Collaboration Suite (formerly Secure Meeting) allows users to schedule and attend secure online meetings. In meetings, users can share their desktops and applications with one another over a secure connection. Meeting attendees can collaborate by enabling remote-control of their desktops and through text chatting. Users can schedule meetings through the Pulse Connect Secure user Web portal or, if they have the Microsoft Outlook Plug-in installed, through Microsoft Outlook.

In addition to regular meetings, Pulse Collaboration Suite supports Instant Meetings and Support Meetings. Instant meetings allow you to create meetings with static URLs for that particular type of meeting (for example, weekly status meetings). You do not need to schedule these types of meetings. The conductor starts the meeting and the invitees enter the URL to attend the meeting.

You can enable Pulse Collaboration Suite integration as part of a Pulse connection and push the connection to Pulse clients through an installer package or a configuration update. The Connect Secure or Policy Secure (L3) connection type includes

a check box that enables Pulse Collaboration integration on the connection. When the check box is selected, Pulse clients that have installed that connection display new menu items that enable users to access Pulse Collaboration Suite functions. A connection that is enabled for meetings can serve as a normal SSL VPN connection or it can be dedicated to meetings only. When a Pulse user clicks the tray icon menu item for meetings, Pulse launches a browser window and connects to the server's user Web portal.

Task Summary: Configuring Pulse Collaboration Suite on Pulse Connect Secure

The following summarizes how to enable a Pulse Connect Secure server as a meeting server for Pulse Collaboration Suite meetings.

To configure Pulse Collaboration Suite:

1. In the Pulse Connect Secure admin console, click **System > Network > Overview** and specify a network identity for the server. Pulse Collaboration Suite uses this hostname when constructing meeting URLs for e-mail notifications.
2. Configure role-level settings:
 - To enable Pulse Collaboration Suite at the role level, click **Users > User Roles > Role Name > General**.
 - To configure role-level meeting restrictions, click **Users > User Roles > Role Name > Meetings > Options**.
3. Configure the authentication settings:
 - To specify the authentication servers meeting creators can access and search click **Users > User Roles > Role Name > Meetings > Auth Servers**.
 - To allow meeting creators to invite users from an LDAP server, click **Authentication > Auth. Servers > Select LDAP Server > Meetings**.
4. Configure meeting sign-in policies:
 - To customize the user Web portal pages that meeting attendees see when they sign into a meeting, click **Authentication > Signing In > Sign-in Pages**.
 - To define the URL that meeting participants use to join a meeting, click **Authentication > Signing In > Sign-in Policies > Meeting Policy**. You also use this page to associate a meeting page with the URL.
 - To associate your meeting sign-in policy with a user sign-in policy, click **Authentication > Signing In > Sign-in Policies > User Policy**. The Pulse Connect Secure server applies the specified meeting URL to any meeting created by a user who signs into the associated user URL.
5. To configure system-level meeting settings, include session time-outs, SMTP server information, time zone settings, and color-depth settings, click **System > Configuration > Pulse Collaboration Suite** page of the admin console.
6. To enable client-side logging, click **System > Log/Monitoring > Client Logs > Settings**.
7. To view the logs that users push to the Pulse server, click **System > Log/Monitoring > Uploaded Logs**.

Configuring Pulse Connections to Support Meetings

When you configure a Pulse Secure Connection to support Pulse Collaboration Suite meetings, Pulse users on Windows endpoints can access meeting functions from the Pulse Secure tray icon. When the user clicks Start Meeting in the tray icon menu, Pulse launches a browser window that provides access to the meeting functions. The browser shows the Meetings page of the server's user Web portal, which a user can also access by using a browser to login to the Pulse Secure Access server.

The tray icon for Pulse Collaboration Suite access is available when a Pulse connection is enabled as a meeting server connection. Pulse users cannot enable the meeting function for a connection. This task must be performed by the Pulse administrator on a connection defined on the server, and then installed on endpoints through normal methods of distributing and updating Pulse client software.

The following steps summarize how to create a Pulse connection that enables Pulse Collaboration Suite functions.

1. In a Pulse connection set, create a new Pulse connection or edit an existing connection set.

Pulse Collaboration Suite is available with SSL VPN connections (connection type Connect Secure or Policy Secure (L3)) only.

2. Select the check box labeled **Enable Pulse Collaboration integration on this connection**.
3. Distribute the connection to endpoints through normal methods of distributing and updating Pulse client software.

Scheduling Meetings Through the Pulse Connect Secure User Web Portal

If you enable meeting creation abilities at the role level, users can create meetings through the Meetings page of the Pulse Connect Secure user Web portal. The user scheduling the meeting must specify all of the standard meeting details such as the meeting name, description, start time, start date, recurrence pattern, duration, password, and a list of invitees.

Additionally, the user must categorize all invitees into one of two categories:

- Pulse Connect Secure invitees—A user who signs into the same Secure Access server or cluster as the meeting creator, also called an in-network invitee. When inviting an in-network user to a meeting, the meeting creator must specify the user's username and authentication server.
- Non-Pulse Connect Secure invitees—A user who signs into a different Secure Access server or cluster as the meeting creator, also called an out-of-network invitee. When inviting an out-of-network user to a meeting, the meeting creator must specify the user's email address.



Note: If an in-network invitee uses the meeting URL instead of the Meetings page in the Pulse Connect Secure user Web portal to join a meeting, Pulse Collaboration Suite classifies the user as an out-of-network invitee.

Scheduling Meetings Through Microsoft Outlook

If you enable meeting creation abilities at the role level, Pulse Connect Secure users can create meetings through the Microsoft Outlook calendar using the Pulse Collaboration Suite Outlook plug-in.

When installing the Pulse Collaboration Suite plug-in on Microsoft Outlook 2000, the following message appears, "The form you are installing may contain macros." Since the Pulse Collaboration Suite form does not contain macros it does not matter whether you click Disable Macros or Enable Macros.



Note: You must use the same Outlook profile to remove the Pulse Collaboration Suite plug-in for Outlook as the one used to install the plug-in. Switching profiles between the installation and removal of the Plug-In is not supported.

To use this plug-in, the user must:

1. Install the plug-in from the Meetings page in the Pulse Connect Secure user Web portal.
2. Open the Pulse Collaboration Suite scheduling form in Outlook by choosing **New > Pulse Collaboration Suite**.
3. Use the Pulse Collaboration Suite tab to enter details about the Pulse Secure Access server on which the meeting should be scheduled as well as the user's sign-in credentials, realm, and a meeting password.



Note: Due to limitations with Microsoft Outlook, not all meeting details cross-populate between Microsoft Outlook and Pulse Connect Secure. For a complete list of restrictions, see the Pulse Collaboration Suite for Outlook information available from the user help system available on the Pulse Connect Secure user Web portal as well as the Pulse Collaboration Suite for Outlook plug-in installer.

4. Use the Scheduling and Appointment tabs to schedule the meeting and add invitees using standard Outlook functionality. Note that Pulse Collaboration Suite supports creating standard or recurring meetings through Outlook.



Note: The Appointment tab has a check box labeled This is an online meeting using. This check box is not related to the Meeting Server or the Pulse Collaboration Suite Outlook Plug-in and cannot be used by a third-party plug-in.

5. Save the calendar entry to send the information to the Pulse Collaboration Suite server. Note that when saving a meeting, the user might see a certificate warning because the plug-in is communicating with a secure server.
6. Outlook sends invitation e-mails to the invitees using the text and meeting URL link constructed by the Pulse Collaboration Suite Outlook plug-in. Outlook also adds the meeting to the Outlook calendars of meeting invitees. This calendar item includes all of the standard information recorded by Outlook as well as an additional Pulse Collaboration Suite tab containing the information specified by the meeting creator in the Pulse Collaboration Suite tab. Note that the Pulse Secure Access server does not send an additional e-mail using the SMTP server.
7. To delete a meeting, click **Delete Meeting from Server** on the Pulse Collaboration Suite tab. Clicking Delete from the Outlook form does not delete the meeting.

CHAPTER 4 Configuring Pulse Secure Client on SRX Series Gateways

- [Pulse Secure Client and SRX Series Gateways](#)
- [Pulse Secure Client and Dynamic VPN Configuration Overview](#)

Pulse Secure Client and SRX Series Gateways

The dynamic virtual private network (VPN) feature of SRX Series gateways simplifies remote access by enabling users to establish Internet Protocol Security (IPsec) VPN tunnels without having to manually configure VPN settings on their endpoints. Pulse Secure client for Windows and Pulse Secure client for Mac support dynamic VPN connectivity to SRX Series gateways. The VPN settings are part of a Pulse SRX connection. Depending on the version of Junos OS on the SRX gateway, you might be able to deploy Pulse to endpoints from the SRX Series gateway through a Web portal. An endpoint accesses the SRX Web portal and, after the user is authenticated, Pulse is downloaded and installed. The default installation includes a Pulse connection to the SRX Series gateway. Alternatively, you can create and deploy SRX connections from Pulse Policy Secure and Pulse Connect Secure. See the *Pulse Secure Supported Platform Guide* on [Pulse Secure.net](#) for details on the Junos OS versions that are able to deploy Pulse.

To configure a firewall access environment for Pulse clients, you must configure the VPN settings on the SRX Series gateway and create and deploy an SRX connection on the Pulse Secure client.



Note: Pulse Secure client for mobile devices can access Pulse Connect Secure only.

For SRX Series gateways that cannot deploy Pulse Secure client software, you have the following configuration and deployment options:

- In an environment that includes Pulse Connect Secure and Pulse Policy Secure, create connections of the type SRX with a target address of your SRX Series Services gateway. Users could then install the Pulse Secure client software and the connection configurations by logging in to the Web portal of the Pulse Connect Secure or Pulse Policy Secure and being assigned to a role that installs Pulse Secure client. After the installation, the endpoint has the Pulse Secure software and the connection information required to connect to the SRX Series Services gateways.
- Install the default Pulse Secure software package, and then have users create new connections that point to the SRX Series gateway.

SRX Series gateways supported an earlier access client called Juniper Networks Access Manager. You must uninstall Access Manager before you deploy Pulse Secure client to endpoints. The Pulse installation program checks for Access Manager. If Access Manager is present, the program displays a message instructing the user to uninstall Access Manager before installing Pulse.



Note: The automatic update feature is supported on Pulse Connect Secure and Pulse Policy Secure servers only. SRX gateways do not support automatic Pulse software updates.

Related Documentation

- [Pulse Secure Client and Dynamic VPN Configuration Overview](#)

Pulse Secure Client and Dynamic VPN Configuration Overview

A dynamic VPN allows administrators to provide IPsec access for Windows endpoints to a Juniper Networks SRX gateway device while also providing a way to distribute the Dynamic VPN software to remote clients through the use of a Web portal.

The following procedure lists the tasks for configuring a dynamic VPN. For detailed information on these topics, see the Junos OS documentation.

1. Configure authentication and address assignment for the remote clients:
 1. Configure an XAuth profile to authenticate users and assign addresses. You can use local authentication or an external RADIUS server. Use the **profile** configuration statement at the **[edit access]** hierarchy level to configure the XAuth profile.

To use the XAuth profile for Web authentication, use the **web-authentication** configuration statement at the **[edit access firewall-authentication]** hierarchy level.
 2. Assign IP addresses from a local address pool if local authentication is used. Use the address-assignment pool configuration statement at the **[edit access]** hierarchy level. You can specify a subnet or a range of IP addresses. Or you can specify IP addresses for DNS and WINS servers.
2. Configure the VPN tunnel:
 1. Configure the IKE policy. The mode must be aggressive. You can use basic, compatible, or standard proposal sets. Only preshared keys are supported for phase 1 authentication. Use the **policy** configuration statement at the **[edit security ike]** hierarchy level.
 2. Configure the IKE gateway. Either shared or group IKE IDs can be used. You can configure the maximum number of simultaneous connections to the gateway. Use the **gateway** configuration statement at the **[edit security ike]** hierarchy level.
 3. Configure the IPsec VPN. You can use basic, compatible, or standard proposal sets with the **policy** configuration statement at the **[edit security ipsec]** hierarchy level. Use the **vpn** configuration statement at the **[edit security ipsec]** hierarchy level to configure the IPsec gateway and policy.
 4. Configure a security policy to allow traffic from the remote clients to the IKE gateway. Use the **policy** configuration statement at the **[edit security policies from-zone zone to-zone zone]** hierarchy level.



Note: The placement of this security policy is important. You must place it above more specific, non-VPN policies so that traffic that is intended to be sent over the VPN tunnel is processed correctly.

5. Configure host inbound traffic to allow specific traffic to reach the device from systems that are connected to its interfaces. For example, IKE and HTTPS traffic must be allowed.
6. (Optional) If the client address pool belongs to a subnet that is directly connected to the device, the device would need to respond to ARP requests to addresses in the pool from other devices in the same zone. Use the proxy-arp configuration statement at the **[edit security nat]** hierarchy level. Specify the interface that directly connects the subnet to the device and the addresses in the pool.
3. Associate the dynamic VPN with remote clients:
 1. Specify the access profile for use with dynamic VPN. Use the **access-profile** configuration statement at the **[edit security dynamic-vpn]** hierarchy level.
 2. Configure the clients who can use the dynamic VPN. Specify protected resources (traffic to the protected resource travels through the specified dynamic VPN tunnel and is therefore protected by the firewall's security policies) or exceptions to the protected resources list (traffic that does not travel through the dynamic VPN tunnel and is sent in clear text). These options control the routes that are pushed to the client when the tunnel is up, therefore controlling the traffic that is sent through the tunnel. Use the **clients** configuration statement at the **[edit security dynamic-vpn]** hierarchy level.

Related Documentation

- [Pulse Secure Client Installation Overview](#)

CHAPTER 5 Session Migration

- Understanding Session Migration [jd0e11929](#)
- Task Summary: Configuring Session Migration
- Configuring Session Migration for the Pulse Client
- Configuring an IF-MAP Federated Network for Session Migration

Understanding Session Migration

This topic describes the session migration feature. It includes the following information:

- [Session Migration Overview](#)
- [Session Migration and Session Timeout](#)
- [How Session Migration Works](#)
- [Session Migration and Session Lifetime](#)
- [Session Migration and Load Balancers](#)
- [Authentication Server Support](#)

Session Migration Overview

When you enable session migration on two or more Pulse servers, a Pulse endpoint can be moved from one location to another and connect to a different Pulse server without providing additional authentication. For example, a user can be connected from home through Pulse Connect Secure, and then arrive at work and connect to Pulse Policy Secure without being reauthenticated. If session migration is not enabled, Pulse users must be reauthenticated each time they attempt to access the network through a different Pulse server.

Sessions can be migrated between Pulse Policy Secure and Pulse Connect Secure servers that are in the same IF-MAP federated network: using either the same IF-MAP server, or using IF-MAP servers that are replicas of one another.

The servers must be in the same authentication group. Authentication groups are configured through authentication realms. An authentication group is a string that you define for common usage. You can use authentication groups to group together realms with similar authentication methods. Such as, one authentication group for SecurID authentication, another authentication group for AD. A single gateway can belong to more than one authentication group, with a different authentication group per realm.

The Pulse server to which a user authenticates publishes session information to the IF-MAP server. Other IF-MAP clients in the federated network can use the information to permit access without additional authentication to users.

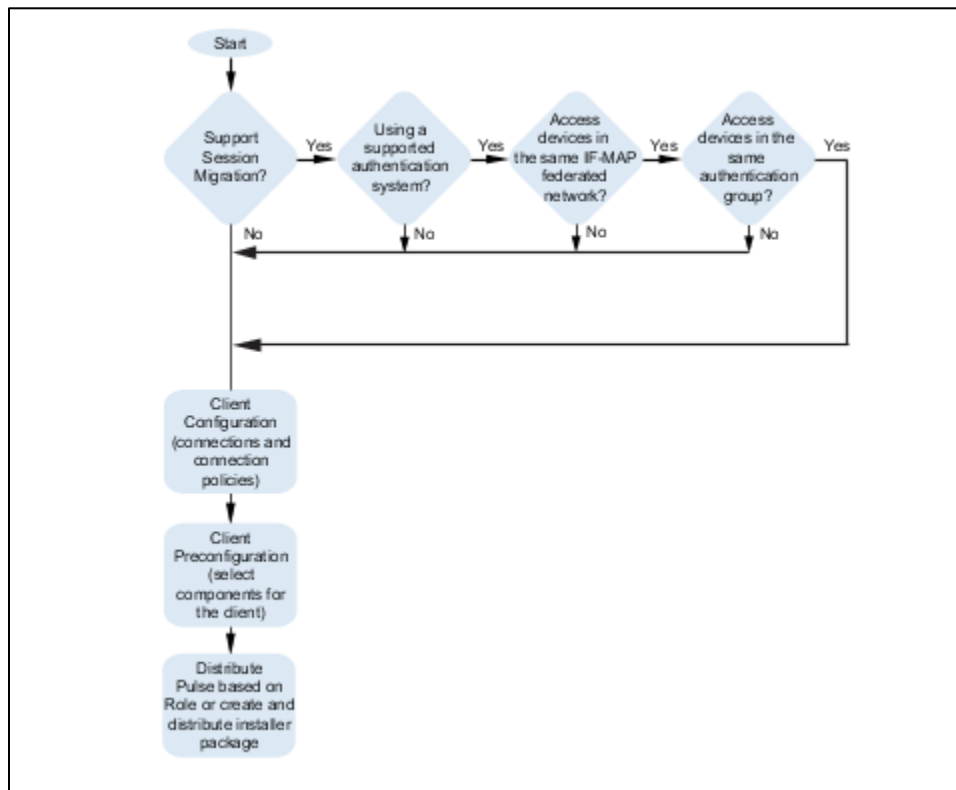
When a user session is migrated to another Pulse server, the new session information is pushed to the IF-MAP server. The IF-MAP server notifies the authenticating server, and information about the session that existed on the original server is removed leaving only session information about the current authenticating server on the IF-MAP server. The authenticating server removes information about the session from its local session table.

When a session is migrated, realm role-mapping rules determine user access capabilities. You can import user attributes when a session is migrated, or you can configure a dedicated directory server to look up attributes for migrated user sessions. To ensure that session migration retains user sessions, configure a limited access remediation role that does not require a Host Checker policy. This role is necessary because the Host Checker timeout can be exceeded if an endpoint is in hibernation or asleep. With the new remediation role, the user's session is maintained.

If additional Host Checker policies are configured on a role or realm to which a migrated session applies, the policies are performed before allowing the user to access the role or realm. Administrators of different Pulse servers should ensure that Host Checker policies are appropriately configured for endpoint compatibility.

[Figure 98](#) illustrates the task flow for enabling session migration for Pulse.

Figure 98: Requirements for Pulse Session Migration



Session Migration and Session Timeout

Session timeout on the authenticating server does not apply to a migrated session. Instead, session start time is applicable. The inbound server evaluates session timeout using the start time of the original session on the original server.

When a user reboots an endpoint for which session migration is enabled, the session is retained for a short time on the server. For sessions on the Pulse Policy Secure, sessions are retained until the heartbeat timeout expires. For Pulse Connect Secure sessions, the idle timeout determines how long the session is retained.

If an endpoint that is connected to a Pulse Policy Secure or Pulse Connect Secure is rebooted and the user does not sign out, when the endpoint is restarted and the user attempts to connect to the same access gateway, Pulse resumes the previous session without requesting user credentials if the previous session is still active.

How Session Migration Works

Session migration uses IF-MAP Federation to coordinate between servers.

When a session is established, the authenticating gateway publishes the session information, including a session identifier, to the IF-MAP server. The session identifier is also communicated to the Pulse client.

When the Pulse client connects to a migrating gateway in the same authentication group, the Pulse client sends the session identifier to the migrating gateway. The migrating gateway uses the session identifier to look up the session information in the IF-MAP server. If the session information is valid, the migrating gateway uses the session identifier to establish a local session for the endpoint that the Pulse client is running on.

The IF-MAP server notifies the authenticating gateway that the user session has migrated, and the authenticating gateway deletes the session information from the IF-MAP server.

Session Migration and Session Lifetime

Session migration is designed to give users maximum flexibility and mobility. Users are no longer tied to the office. The workplace can travel with the user, and electronic chores such as online banking can come to work. Because of this flexibility, users might be away from their machines for long periods of time, allowing their active session to expire. Session migration requires users to have an active session on Pulse Policy Secure or Pulse Connect Secure.

You can adjust session lifetime to ensure that sessions do not time out while users are away from their machines. You adjust session lifetime on the gateway by selecting **Users > User Roles > Role Name > General > Session Options** in the admin console.

Session Migration and Load Balancers

A Pulse client that connects to a Pulse server that is behind a load balancer will attempt to migrate the network connection if the connected server fails. The Pulse servers must be federated and configured for session migration. For example, a load balancer that balances to 2 Pulse servers (non-clustered) balances to Server1. If Server1 fails, the load balancer then balances to Server2. A Pulse client that is connected to Server1 is migrated to Server2 without re-authentication.

Authentication Server Support

The behavior of session migration depends to some extent on the authentication server on the inbound side.

The following list provides a summary of authentication server support:

- Local authentication server—Migration succeeds if the username is valid on the local authentication server.
- LDAP server—Migration succeeds if the LDAP authentication server can resolve the username to a distinguished name (DN).
- NIS server—Migration succeeds if the NIS authentication server can find the username on the NIS server.
- ACE server—Migration always succeeds.
- RADIUS server—Migration always succeeds. If you select **Lookup Attributes using Directory Server**, no attributes are present in the user context data.
- Active Directory—Migration always succeeds. The Lookup Attributes using Directory Server option might not work, depending on your configuration.
- Anonymous—No support for migrating sessions because sessions are not authenticated.
- Siteminder—No support for migrating sessions because Siteminder SSO is used instead.
- Certificate—No support for migrating sessions because sessions are authenticated using certificates.
- SAML—No support for migrating sessions because SAML SSO is used instead.



Note: For local, NIS, and LDAP authentication servers, the inbound username must reflect an existing account.

Related Documentation

- [Configuring Session Migration for the Pulse Client](#)
- [Task Summary: Configuring Session Migration](#)

Task Summary: Configuring Session Migration

To permit session migration for users with the Pulse client, perform the following tasks:

1. Configure location awareness rules within a client connection set to specify locations included in the scope of session migration for users. For example, configure location awareness rules for a corporate Pulse Policy Secure server connection and a Pulse Connect Secure server connection.
2. Configure an IF-MAP federated network, with the applicable Pulse servers as IF-MAP Federation clients of the same IF-MAP Federation server.
3. Ensure that user entries are configured on the authentication server for each gateway.
4. Ensure that user roles are configured for all users on each gateway.
5. Define a remediation role with no Host Checker policies to allow user sessions to be maintained when an endpoint is sleeping or hibernating.
6. Configure role-mapping rules that permit users to access resources on each gateway.
7. Enable and configure session migration from the User Realms page of the admin console.
8. Distribute the Pulse client to users.

Related Documentation

- [Understanding Session Migration](#)
- [Configuring Session Migration for the Pulse Client](#)

Configuring Session Migration for the Pulse Client



Note: Ensure that all of the Pulse Policy Secure servers and Pulse Connect Secure servers for which you want to enable session migration are IF-MAP Federation clients of the same IF-MAP Federation server. Additionally, make sure that each gateway is configured according to the procedures outlined in this section.

To configure session migration:

1. In the admin console, select **Users > User Realms**.
2. Select an existing realm, or create a new realm.
3. On the General page, select the **Session Migration** check box. Additional options appear.
4. In the **Authentication Group** box, enter a string that is common to all of the gateways that provision session migration for users. The authentication group is used as an identifier.
5. Select for either the **Use Attributes from IF-MAP** option button or the **Lookup Attributes using Directory Server** option.



Note: Select Lookup Attributes using Directory Server only if you are using an LDAP server. Attributes are served faster with an LDAP server.

Related Documentation

- [Understanding Session Migration](#)
- [Task Summary: Configuring Session Migration](#)

Configuring an IF-MAP Federated Network for Session Migration

To successfully deploy session migration, you configure a Pulse Policy Secure IF-MAP server, and you configure all of the connected Pulse servers that users access as IF-MAP clients. A Pulse Connect Secure server cannot be an IF-MAP server.

To add clients, you must specify the IP address and the security mechanism and credentials for the client.

An IF-MAP server certificate must be installed on the IF-MAP server. The client verifies the server certificate when it connects to the server. The server certificate must be signed by a Certificate Authority (CA), the client must be configured to trust certificates signed by that CA, and the hostname in the server certificate must match the hostname in the IF-MAP URL on the client.

You must identify the IF-MAP server to each Pulse server IF-MAP client. To add the server, you specify the IF-MAP URL of the server and how to authenticate to the server. Match the URL and security settings to equal those on the IF-MAP server to which the IF-MAP clients will connect.

To configure IF-MAP server settings on the Pulse Policy Secure server:

1. From the admin console select **System > IF-MAP Federation > Overview**.
2. Under Choose whether this Pulse Policy Secure server runs an IF-MAP Server, an IF-MAP client, or no IF-MAP, select the **IF-MAP Server** option button.
3. Click **Save Changes**.
4. From the admin console select **System > IF-MAP Federation > This Server > Clients**.
5. Under IF-MAP Client, enter a **Name** and an optional **Description** for this client.

For example, enter the name CS-access1.corporate.com and the description Connect Secure 1.

6. Type one or more IP addresses of the client. If the client is multi-homed, for best results list all of its physical network interfaces. If the client is a Pulse Policy Secure server or Pulse Connect Secure cluster, list the internal and external network interfaces of all nodes. You must enter all of the IP addresses for all of the interfaces because equipment failures might cause traffic between the IF-MAP client and the IF-MAP server to be re-routed through a different network interface. Listing all of the IP addresses maximizes the probability that IF-MAP Federation still works in the event of a failure.

For example, enter 172.16.100.105.

7. Under Authentication, select the Client Authentication Method: **Basic or Certificate**.
 1. If you select **Basic**, enter a Username and Password. The same information should be added to the IF-MAP server.
 2. If you select **Certificate**, choose which Certificate Authority (CA) to use to verify the certificate for this client. Optionally, specify certificate attributes or restrictions to require values for certain client certificate attributes.
8. Click **Save Changes** to save the IF-MAP Client instance on the IF-MAP server.

To configure IF-MAP client settings on the Pulse server clients:

1. From the admin console select **System > IF-MAP Federation > Overview**.
2. In a Policy Secure server, under Choose whether this server runs an IF-MAP Server, an IF-MAP client, or no IF-MAP, select the **IF-MAP Client** option button. On a Pulse Connect Secure server, select **Enable IF-MAP Client** check box.
3. Type the server URL for IF-MAP Web service on the IF-MAP server. Append the server URL with **/dana-ws/soap/dsifmap** for all Pulse Secure IF-MAP servers.

For example, https://access2.corporate.com/dana-ws/soap/dsifmap.

4. Select the client authentication method: **Basic or Certificate**.
5. If you select **Basic**, enter a username and password. This is the same as the information that was entered on the IF-MAP server.
6. If you select **Certificate**, select the device certificate to use.

Ensure that the certificate of the CA that signed the IF-MAP server certificate is added from the System > Configuration > Certificates > Trusted Server CA page.

The IF-MAP client validates the IF-MAP server certificate: if validation fails, the connection fails. Ensure that the hostname in the IF-MAP URL on the client machine matches the hostname of the server certificate on the IF-MAP server, and that the CA that signed the server certificate is configured as a trusted server CA on the IF-MAP client.

7. Click **Save Changes**.

Related Documentation

- [Understanding Session Migration](#)
- [Task Summary: Configuring Session Migration](#)

CHAPTER 6 Deploying Pulse Secure Client

- [Pulse Secure Client Installation Overview](#)
- [Adding a Pulse Configuration to a New Pulse Installation](#)
- [Installing Pulse Secure Client from the Web](#)
- [Launching Pulse Secure Client from the Pulse Server Web Portal](#)
- [Launching Pulse Desktop Client using URL](#)
- [Installing the Pulse Secure Client on Windows Endpoints Using a Preconfiguration File](#)
- [Installing the Pulse Secure Client on OS X Endpoints Using a Preconfiguration File](#)
- [Pulse Secure Client Command-line Launcher](#)
- [Using jamCommand to Import Pulse Secure Connections](#)
- [jamCommand Reference](#)

Pulse Secure Client Installation Overview

This section describes how to deploy Pulse for Windows and Pulse for macOS client software from Pulse Policy Secure and Pulse Connect Secure platforms.

Pulse Policy Secure and Pulse Connect Secure include a default connection set and a default component set. These defaults enable you to deploy the Pulse client to users without creating new connection sets or component sets. The default settings for the client permit dynamic connections, install only the components required for the connection, and permit an automatic connection to Pulse Connect Secure or Pulse Policy Secure to which the endpoint connects.

In all deployment scenarios, you must have already configured authentication settings, realms, and roles.

You can deploy the Pulse Secure client to endpoints from Pulse Connect Secure and Pulse Policy Secure in the following ways:

- **Web install**—With a Web install (also called a server-based installation), users log in to the Pulse server's Web portal and are assigned to a role that supports a Pulse installation. When a user clicks the link to run Pulse Secure client, the default installation program adds Pulse to the endpoint and adds the default component set and the default connection set. If you do not make any changes to the defaults, the endpoint receives a Pulse installation in which a connection to the Pulse server is set to connect automatically. You can edit the default connection set to add connections of other Pulse servers and change the default options.



Note: The exact mechanism used to launch and install a particular Pulse Secure client from a web browser depends on a number of factors, including:

- The Pulse Secure client (Windows/Mac desktop client, Host Checker, Windows Terminal Services, Secure Meeting client) being launched/installed.
- The endpoint operating system type and version.
- The web browser type and version.
- The security settings of the endpoint operating system and browser.

For a particular client/OS/browser combination, you may need to enable the appropriate technology on the endpoint device. For example, to launch the Pulse Secure desktop client from Firefox on Windows, you will need to

ensure that Java is enabled in Firefox on the end user's endpoint device. For more information, please consult the "Adaptive Delivery" section of the Pulse Secure Desktop Client Supported Platforms Guide.



Note: A Web install is not compatible with the Pulse rebranding tool, BrandPackager.

- *Preconfigured installer*—Create the connections that an endpoint needs for connectivity and services, download the settings file (.pulsepreconfig), and download default Pulse installation program. For Windows endpoints you run the Pulse installation program by using an msexec command with the settings file as an option. For OS X endpoints, you run the default installer and then import the .pulsepreconfig file using a separate command.
- *Default installer*—You can download the default Pulse installation program and distribute it to endpoints using your local organization's standard software distribution method (such as Microsoft SMS/SCCM). The Pulse Secure client software is installed with all components and no connections. After users install a default Pulse installation, they can add new connections manually through the Pulse client user interface or by using a browser to access a Pulse server's Web portal. For the latter, the Pulse server's dynamic connection is downloaded automatically and the new connection is added to the Pulse client's connections list when the user starts Pulse by using the Pulse server's Web portal interface. Dynamic connections are created as manual rather than automatic connections, which means that they are run only when the user initiates the connection or the user browses to a Pulse Server and launches Pulse from the server's Web interface.

If the Windows endpoints in your environment do not have admin privileges, you can use the Pulse Secure Installer program, which is available on the admin console System Maintenance Installers page. The Pulse Secure Installer allows users to download, install, upgrade, and run client applications without administrator privileges. In order to perform tasks that require administrator privileges, the Pulse Secure Installer runs under the client's Local System account (a powerful account with full access to the system) and registers itself with Windows' Service Control Manager (SCM). An Active-X control or a Java applet running inside the user's Web browser communicates the details of the installation processes to be performed through a secure channel between the Pulse server and the client system.

- Installing the Pulse Secure Installer MSI package requires administrator rights to install onto your client systems. If you plan to use the EXE version, administrator rights is not needed as long as a previous version of the access service component (deployed through, for example, JIS, Pulse, and so forth) is already present. If policies are defined for your client with the group policy "Run only Allowed Windows Application", the following files must be allowed to run in the group policy. If not, client applications might not install.
 - dsmmf.exe
 - PulseCompMgrInstaller.exe
 - PulseSetupClient.exe
 - PulseSetupClientOCX.exe
 - PulseSetupXP.exe
 - uninstall.exe
 - x86_Microsoft.*.exe
- You should ensure that the Microsoft Windows Installer exists on the client system prior to installing the Pulse Secure Installer.
- Your end-users' client systems must contain either a valid and enabled Java Runtime Engine (JRE) or a current Pulse Connect Secure ActiveX control. If the client systems do not contain either of these software components, the users will be unable to connect to the gateway. If there is no JRE on your end-users' client systems, you should download an appropriate installer package from Maintenance > System > Installers. The service appears in the Windows Services (Local) list as Neoteris Setup Service. The service starts automatically on install and during client system start up.

Related Documentation

- [Adding a Pulse Configuration to a New Pulse Installation](#)

- [Installing Pulse Secure Client from the Web](#)
- [Installing the Pulse Secure Client on Windows Endpoints Using a Preconfiguration File](#)
- [Installing the Pulse Secure Client on OS X Endpoints Using a Preconfiguration File](#)

Adding a Pulse Configuration to a New Pulse Installation

When you install the Pulse Secure client for Windows or Pulse Secure client for macOS client on an endpoint using the default Pulse installation program, the endpoint has all the Pulse components it needs to connect to Pulse servers. However, the Pulse client needs a configuration that identifies the Pulse servers it can connect to, that is, the connections. Connection properties also define how the connections are to be started, manually, automatically, or according to location awareness rules, and how Pulse connections receive updates. These connection set properties are also called machine settings. [Figure 99](#) shows the default Pulse connection set properties (machine settings) that are passed to the Pulse client as its configuration. [Figure 100](#) shows the connection set properties as they appear in a Pulse preconfiguration file, which you can use to add the Pulse configuration when you install Pulse. The preconfiguration file also includes Pulse connections.

Figure 99: Pulse Secure Client Configuration Properties Defined on the Pulse Server

Pulse Secure Client > Connections > default1

default1

Name:

Description:

Owner: DESKTOP
 Last Modified: 2018-09-18 08:28:52 UTC
 Server ID: 0320MP9R509HB0ILS

▼ Always-on vpn wizard

[Configure Always-on VPN using wizard](#)

▼ Options

Name	Value
Allow saving logon information Enables the Save settings checkbox in the certificate trust and password prompts.	<input checked="" type="checkbox"/>
Allow user connections Allows user to create connections via the Pulse UI.	<input checked="" type="checkbox"/>
Always-on Pulse Client Prevents end users from circumventing Pulse connections. This option will disable all configuration settings that allow the end user to disable or remove Pulse connections, services or software.	<input type="checkbox"/>
Display Splash Screen Controls whether the splash screen is displayed when Pulse starts.	<input checked="" type="checkbox"/>
Dynamic certificate trust Controls whether users may accept to trust unknown certificates.	<input checked="" type="checkbox"/>
Dynamic connections Allows connections to be deployed automatically from devices.	<input checked="" type="checkbox"/>
EAP Fragment Size Maximum number of bytes in an EAPOL message from the client for 802.1x connections. Range: 450 - 3000 bytes	<input type="text" value="1400"/>
Enable captive portal detection Pulse will attempt to detect the presence of a captive portal hotspot. Only applies to Connect Secure and Policy Secure (L3) connections.	<input checked="" type="checkbox"/>
Enable embedded browser for authentication Pulse will use embedded browser for saml, custom sign-in or token based authentication.	<input checked="" type="checkbox"/>
Enable embedded browser for captive portal Pulse will use an embedded web browser for captive portal pages. Only applies when captive portal detection is enabled.	<input checked="" type="checkbox"/>
FIPS mode enabled Deploy client with Federal Information Processing Standard enabled.	<input type="checkbox"/>
Prevent caching smart card PIN Enabling this will ensure the smart card PIN value is not cached by the client process.	<input type="checkbox"/>
VPN only access When Pulse client connects to a PCS having lock down mode enabled, it will enter lock-down mode and won't let any traffic flow through unless a Locked-down VPN connection is in connected state. User is allowed to connect or disconnect any connection. User is allowed to add any new connection/server URI. User is allowed to delete a connection if the connection is not locked down.	<input type="checkbox"/>
Wireless suppression Disconnect all wireless interfaces when a wired interface gets connected to a network. Applies to all wireless connections (not just those managed by Pulse).	<input type="checkbox"/>

► Connections

There are two methods for installing an initial configuration on a new Pulse client:

- Use a Pulse preconfiguration file (.pulsepreconfig) when you install Pulse on endpoints using the default Pulse installer.
- Instruct users to open a browser and login to the Pulse server Web portal where the Pulse configuration has been defined. After successful login, the user should start Pulse Secure client from the Web page. Or you can enable Auto-launch as a role option to have the Pulse installation begin automatically after login.

The first time a Pulse client connects to a server that offers a Pulse configuration, the configuration settings are installed on the client, and the client is bound to that server, which means that only that server can update the client's configuration. Any Pulse server can update the Pulse client software version if that feature is enabled, and any Pulse server can add a

connection to an existing Pulse client configuration if the Dynamic connections option is enabled as part of the connection set on the binding server. Only the binding server can update the Pulse client's configuration.

If the Pulse configuration has Dynamic connections enabled, then connections from other Pulse servers are automatically added to the Pulse client's connections list when the user connects to the other Pulse server through that server's Web portal, and the user starts Pulse using the Pulse server's Web portal interface. For example, a user has a Pulse configuration from PulseServerA (the binding server) and the Pulse configuration allows dynamic connections. If the user browses to PulseServerB and successfully authenticates through that server's Web portal and clicks the Pulse button, the server adds a PulseServerB connection to the Pulse client configuration, and it appears in the Pulse client's connection list. This new connection is set to start manually so that it does not attempt to connect when the endpoint is restarted or conflict with the connections from the binding server. A dynamic connection is added to the Pulse client's connections list. However, the connection's target URL is Pulse Web server URL; it does not use the URL that is defined for the connection in the server's Pulse connection properties. In most cases, these URLs will be the same.

You can see a Pulse configuration by creating and viewing a pulsepreconfig file. (To create the file, go to the Pulse Component screen, select a component set, and then click the Download Pulse Configuration button.) The .pulsepreconfig file contains a section that defines the machine settings and separate sections for each Pulse connection deployed to the client, as shown in [Figure 100](#).

Figure 100: Pulse Secure Client Configuration Properties in a Preconfiguration File

```

schema version {
  version: "1"
}

machine settings {
  version: "14"
  guid: "bf4801a3-527f-4f98-9ea3-7dcb7e271bc9"
  connection-source: "preconfig"
  server-id: "0241ML82A0PRD1VR"
  allow-save: "true"
  user-connection: "true"
  splashscreen-display: "true"
  dynamic-trust: "true"
  dynamic-connection: "true"
  wireless-suppression: "false"
}

ive "8211f09f-6674-4bdb-a44a-e6fa8b7402eb" {
  friendly-name: "SA"
  version: "2"
  guid: "8211f09f-6674-4bdb-a44a-e6fa8b7402eb"
  server-id: "0241ML82A0PRD1VR"
  connection-source: "preconfig"
  factory-default: "true"
  uri: "10.64.78.34"
  connection-policy-override: "true"
  use-for-secure-meetings: "false"
  use-for-connect: "true"
  connection-identity: "user"
  connection-policy: "automatic"
  client-certificate-location-system: "false"
}

8021x "06cc1f68-3714-4871-9abf-458f1c0ef4b0" {
  friendly-name: "MachAuthCnrxn"
  version: "2"
  guid: "06cc1f68-3714-4871-9abf-458f1c0ef4b0"
  server-id: "0241ML82A0PRD1VR"
  connection-source: "preconfig"
  adapter-type: "wireless"
  outer-username: "anonymous"
  scan-list: "juniper_wireless_network"
  non-broadcast-ssid: "false"
  connection-identity: "machine-only"
  connection-policy: "automatic"
}

```

The machine settings and each centrally configured connection include the server ID (server-id) of the binding server. When a user browses to a Pulse server, the server can offer a new configuration, (that is, updates to the machine settings). If the server-id under machine settings matches, the Pulse client accepts the configuration update. If the server-id does not match, the Pulse client ignores the update.

Configuration files have a version number as well. When a Pulse client connects to its binding server, Pulse compares the version of its existing configuration to the version on the server. If the server version is later than the existing client version, the client configuration is updated. The update might add, change, or remove connections and change machine settings.

If you have several Pulse servers and you want to provision the same Pulse configuration from all of the servers, the server ID of the Pulse configuration must be the same across all of the servers. To accomplish this, you create the configuration on one server, and then use the "push config" feature of the Pulse server to push the configuration to the other Pulse servers. This method ensures that the server ID of the configuration file is the same across all of the Pulse servers so that clients can receive a configuration update from any of the Pulse servers.

Related Documentation

- [Pulse Secure Client Installation Overview](#)
- [Introducing Pulse Secure Client](#)

Installing Pulse Secure Client from the Web

For a Web install, you direct users to the Web interface of the Pulse server. After a successful login, a user is assigned to a role that includes an automatic download and installation of the Pulse client software.

Note: In order to install the Pulse Secure desktop client from a web browser, you may need to enable certain browser plugins or other technologies on the endpoint device. For example, Java must be enabled on the endpoint device to install the Pulse client from Firefox, and either ActiveX or Java must be enabled to install the Pulse client from Internet Explorer.



Pulse Connect Secure 8.2r1 and Pulse Policy Secure 5.3r1 introduced a new web-installation option called "Pulse Secure Application Launcher" (PSAL). PSAL leverages "URL handler" functionality by invoking a custom URL in a manner that instructs the web browser to execute a program that launches/install the appropriate Pulse Secure client. PSAL was created to address both the restrictions placed on Java on macOS and the deprecation of Java (and ActiveX) plugins in Google Chrome version 45 and the Microsoft Edge browser. You can read more about the PSAL in Pulse Secure's KB (Knowledge Base) article [KB40102](#).

For a full discussion of this subject, see the "**Adaptive Delivery**" section of the Pulse Secure Desktop Client Supported Platforms Guide.

The default Pulse Secure client installation settings includes minimal components, which includes the Host Checker component, and a connection to the Pulse server. If you want a Web install that has customized settings, you can do any of the following:

- Edit the default connection set and add new connections. The default installer uses the default component set which includes the default connection set.
- Create a new connection set and edit the default component set to include the new connection set.
- Edit the role to specify a component set that includes the connections you want for the default installation.



Note: A Pulse installation causes a restart of active network connections on a Windows endpoint. When a user initiates a Pulse installation through a WAN connection to the Web interface of a Pulse server, the user might need to log in to their service provider again to reestablish network connectivity. Users need to be aware of this issue before they begin the installation.

Related Documentation

- [Pulse Secure Client Installation Overview](#)
- [Launching Pulse Secure Client from the Pulse Server Web Portal](#)

Launching Pulse Secure Client from the Pulse Server Web Portal

One typical method of establishing a VPN connection is for users to browse to the Pulse server's Web portal, login, and then launch Pulse from the Web page.

The following items describe the Pulse connection behaviors:

- The Pulse client has been installed on the endpoint by using the default Pulse installer. The installed Pulse client does not yet have any connections. The user browses to the Pulse server, logs into the server, and then clicks the Pulse button on the Web portal page. The following action occurs:
 1. The default Pulse connection set is automatically deployed to the client.
 2. The connection that has a URL that matches the server URL is launched.
- The Pulse client has been installed on the endpoint and it has a connection from the Pulse server. The user browses to the Pulse server, logs into the server, and then clicks the Pulse button on the Web portal page. The following action occurs:
 1. The connection that has a URL that matches the server URL is launched.
- Pulse has been installed on the endpoint and it has a connection from two different Pulse servers. The user browses to one of these Pulse servers, logs into the server, and then clicks the Pulse button on the Web portal page. The following action occurs:
 1. Only the connection that has a URL that matches the server URL is launched.
- Pulse has been installed on the endpoint. It has a connection for one Pulse server but the user browses to a different Pulse server, logs into the server, and then clicks the Pulse button on the Web portal page. The following action occurs:
 1. A new dynamic connection is created on the Pulse client for this Pulse server. (Note that the default connection on the server must be configured as a dynamic connection.) This new connection is a manual connection, that is, it does not start automatically when Pulse starts.
 2. The new connection for this Pulse server is started based on matching URLs.

Usage Notes

The Web browser method of launching Pulse is affected by the following configuration issues:

- The Pulse connection URL and the server URL must be an exact match. Pulse does not perform reverse DNS lookup to find a match.
- Connections that have the connection property **Allow user to override connection policy** disabled cannot be launched from the browser even if URLs match.

Related Documentation

- [Adding a Pulse Configuration to a New Pulse Installation](#)
- [Installing the Pulse Secure Client on Windows Endpoints Using a Preconfiguration File](#)
- [Installing the Pulse Secure Client on OS X Endpoints Using a Preconfiguration File](#)

Launching Pulse Desktop Client using URL

Launching Pulse Desktop Client using URL feature enables the user to launch the Pulse Desktop Client using the admin prescribed URL. This feature is supported for Windows only.

Administrator creates a web URL (in a prescribed format), and provides it to the user in the following ways:

- URL is placed in a web page in the form of a link and the address of the link is provided to the user.
- URL itself is provided to the user.

User clicks on the link or types the URL in the browser. Pulse Desktop Client gets launched and the connection is redirected to the gateway mentioned in the URL.

User receives a link or a URL which has been crafted by an administrator. Following is the format of the URL:

```
pulsesecureclient://connect?name=NAME&server=SERVERURL&userrealm=REALM&username=USER
&store=TRUE
```

Table 17 lists the parameter and their description mentioned in URL:

Table 16: Parameters details of URL

Parameter	Mandatory/Optional	Action
<i>pulsesecureclient</i>	Mandatory	// URI scheme for URL launching.
<i>connect</i>	Mandatory	This parameter is an action item and establishes the connection.
<i>name</i>	Mandatory	<p>This parameter is a unique parameter, which defines the name of the connection. This connection name is used to identify a specific connection.</p> <p>Connection name will be suffixed by (Auto Launch) for Pulse Desktop connection established through URL. Connection name will be displayed as Name(Auto Launch).</p> <p>Note: <i>name</i> parameter is case sensitive. For example, a connection named as Connection1 will be different from a connection named as connection1.</p>
<i>server</i>	Mandatory	<p>This parameter defines the sign-in URL, to which the Pulse Desktop Client should get connected. It can be any one of the following:</p> <ul style="list-style-type: none"> ○ FQDN ○ IP address (IPv6 and IPv4) ○ A Sign-in URL
<i>userrealm</i>	Optional	<p>This parameter defines the user realm.</p> <p>Note: <i>userrealm</i> parameter is case sensitive.</p>
<i>username</i>	Optional	<p>This parameter defines the username.</p> <p>Note: <i>username</i> parameter is case sensitive.</p>
<i>store</i>	Optional	<p>If <i>store</i> value is "True", then the connection information gets saved in the connection store.</p> <p>If <i>store</i> value is "False", then the connection information will not be saved in the connection store.</p> <p>It provides the flexibility for the user to save the connection information for future purposes.</p>

Following is the scenario to understand the behaviour of this feature:

In this scenario, user establishes a connection named as **Pulse Connection** to the PCS server with userrealm as **Users** and username as **test_user**. Also, user wants to store the connection in Pulse Desktop Client for future references.

Administrator will craft the URL with the values mentioned in below table:

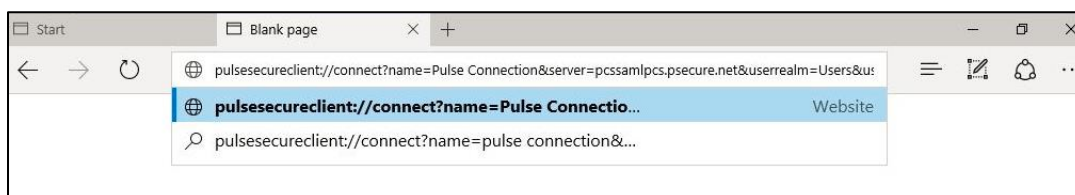
Table 17: Parameter details

Parameter	Values
<i>name</i>	Pulse Connection
<i>server</i>	https://pcssamlpcs.psecure.net/
<i>userrealm</i>	Users
<i>username</i>	test_user
<i>store</i>	true

1. User receives a link or the below mentioned URL which has been crafted by an administrator.

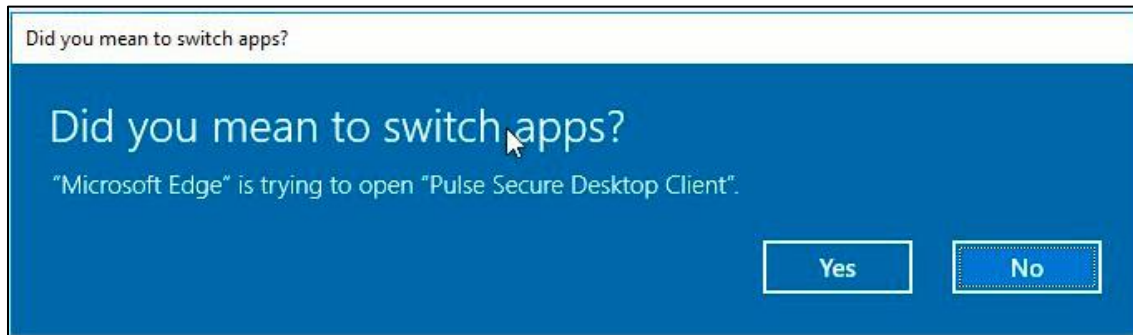
`pulsesecureclient://connect?name=Pulse Connection&server=https://pcssamlpcs.psecure.net/&userrealm=Users&username=test_user&store=true`

Figure 101: Browser URL



2. Once the user opens the URL (in Edge browser), following screen appears:

Figure 102: Switch apps



A permission dialog box appears to get the confirmation from the user to launch Pulse Desktop Client application via URL.

3. User clicks **Yes** button and Pulse Desktop Client gets launched.

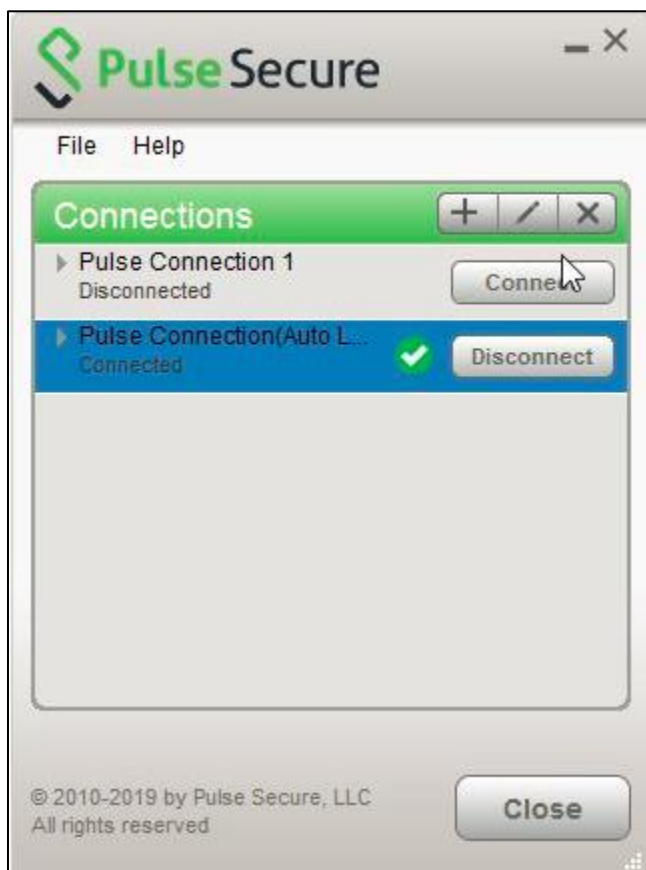
A connection with the name specified in the URL (Pulse Connection) is added in the Pulse Desktop Client and following screen appears:

Figure 103: Authentication Window

The image shows a Pulse Secure authentication window. At the top left is the Pulse Secure logo, which consists of a green stylized 'P' icon followed by the text 'Pulse Secure'. Below the logo, the text 'Connect to: Pulse Connection(Auto Launch)' is displayed. Underneath this, there are two input fields. The first is labeled 'User Name:' and contains the text 'test_user'. The second is labeled 'Password:' and is currently empty. At the bottom right of the window are two buttons: 'Connect' and 'Cancel'.

Note: Connection name will be displayed as Pulse Connection(Auto Launch). Connection name will be suffixed by (Auto Launch) for Pulse Desktop connection established through URL.

4. User enters the password and clicks the **Connect** button. Following screen appears:

Figure 104: Pulse Connection(Auto Launch) established

Now, connection named Pulse Connection(Auto Launch) with provided values as mentioned in Table 17 is established. Full connection name can be viewed in Edit window as shown in the following screen.

Figure 105: Pulse Connection(Auto Launch)



After this, Pulse Desktop Client is launched, and a connection named Pulse Connection(Auto Launch) is established. This connection is then established with username as test_user.

After successful connection establishment, if user decides to disconnect Pulse Connection(Auto Launch), click **Disconnect** button. Pulse Connection(Auto Launch) connection gets disconnected and connection details gets stored in the Pulse Desktop Client for future references, as *store* parameter is set to true in this scenario.

Otherwise, if *store* parameter is set to false, then the connection details of Pulse Connection(Auto Launch) would not be stored after disconnection. Also, next launch of the PDC with same URL will create a new connection.

If the user tries to connect the connection with same connection name but with different server URL, following error message appears:

Figure 106: Error Message



Benefits:

Following are the benefits of this feature:

- **Fast Connection** - As URL will handle the Pulse Desktop Client launch, user needs not to login through PCS, which reduces number of logins, hence time saving and fast connection.

- **Enhancing User Experience** – When PCS (IP or FQDN based), username and realm are prefilled, user just needs to enter the password to login.
- With the help of *Store* parameter in launch URL format, it will be possible to have temporary client entries. This ensures that each connection need not to be stored in the PDC and PDC does not get filled up with a pile of entries.
- **Scriptability** - Programmatically driven launch of Pulse Desktop Client lessens the burden of the Administrator.

Following is the scenario to understand the scriptability behaviour of this feature:

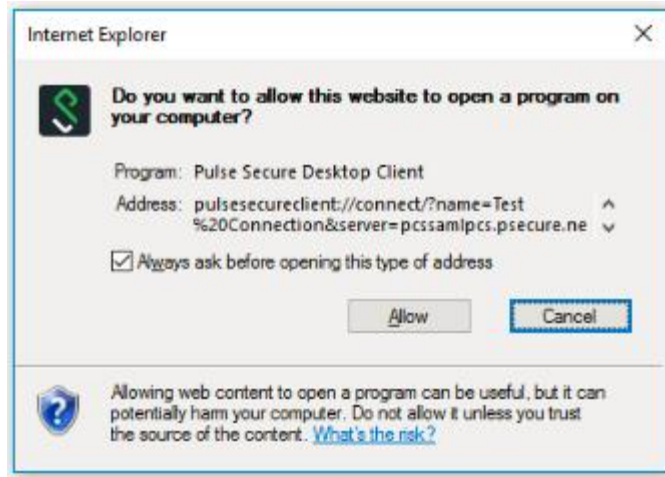
1. User enters the URL in command Prompt as shown below:

Figure 107: Command Prompt Window



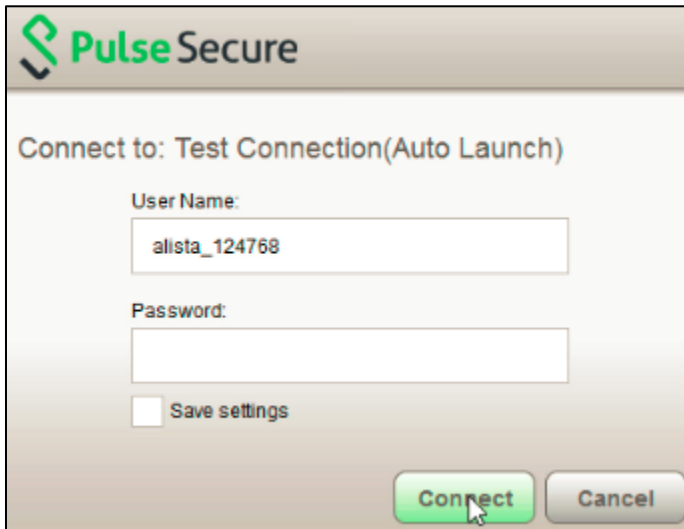
A permission dialog box appears to get the confirmation from the user to launch Pulse Desktop Client application.

Figure 108: Pop-up Window



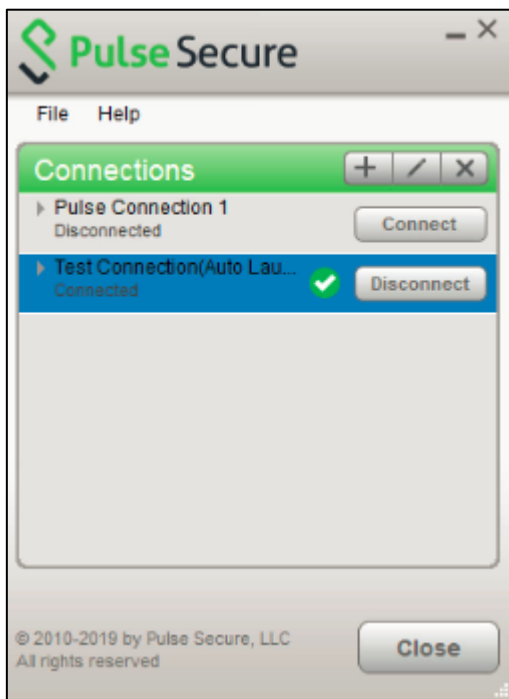
2. User clicks **Allow** button, following authentication screen appears for user to authenticate:

Figure 109: Authentication Window



3. Click **Connect**. Following screen appears:

Figure 110: Test Connection(Auto Launch) established



Connection named Test Connection(Auto Launch) with provided values in Table 17 is established.

Installing the Pulse Secure Client on Windows Endpoints Using a Preconfiguration File

The following procedures apply to Windows installations only.

After you create client connection sets and specify the connections to include within a client component set, you can create a preconfiguration file with all of the connections you want to distribute with the Pulse client. You specify the preconfiguration file as an option when you run the Pulse MSI installer program using an `msiexec` (`windows\system32\msiexec.exe`) command.

To create a preconfigured Pulse installer for distribution to Windows endpoints:

1. Select **Users > Pulse Secure > Connections** and create a connection set with the connections that you want to distribute.
2. Select **Users > Pulse Secure > Components**.
3. If necessary, create a new component set with the connection sets you want to distribute.

It does not matter which component option you select, All components or No components. The Pulse installer installs all components.

4. Select the check boxes next to the component sets that you want to distribute.
5. Click **Download Installer Configuration**.

You are prompted to save the preconfiguration. You can also specify the name of the target Pulse server for the connections, which enables you to create configuration files that are the same except for the target server.

The default filename of the `.pulsepreconfig` file is the name of the selected component set. Make note of the filename and location where you put the file. The preconfiguration file must be available to the clients either through a network share or distributed along with the PulseSecure Desktop client installation file.

6. Select **Maintenance > System > Installers**.

If necessary for your environment, download and install the Pulse Secure Installer. To install Pulse, users must have appropriate privileges. The Pulse Secure Installer allows you to bypass privilege restrictions and allow users with limited privileges to install Pulse. See *Downloading Client Installer Files* for more information about Pulse Secure Installer.

7. Download the appropriate Pulse Secure installer for your Windows environment:
 - Pulse Secure installer (32-bit)
 - Pulse Secure installer (64-bit)

Note: For a Windows installation (.msi) that uses an automated distribution mechanism and where the users do not have administrator privileges, you should ensure that the installation is run in the proper context, typically the USER context. To install in USER context, first advertise the .msi while in the SYSTEM context. For example, to advertise the 64-bit Windows installation to all users, use the following `msiexec` command:



```
msiexec /jm \PulseSecure.x64.msi
```

The advertisement allows the installation to be run in USER context even if the user is a restricted (non-admin) user. The location where the advertisement is run and where the actual installation is run must be the same. If the installation is an upgrade, you must advertise the upgrade version before running it. (Note that it is much easier to upgrade the Pulse client by not disabling the automatic upgrade feature on the Pulse server.) After the installation is run by the user, the Pulse client will use the correct user certificate and context.

Installing the Pulse Client Using Advanced Command-Line Options

The Pulse Secure installer includes the Pulse client and all the software components for all the Pulse services. The preconfiguration file contains the definitions of the Pulse connections that provide client access to specific Pulse servers and services.

Usage Notes:

- The preconfigured installer installs all Pulse components.
- When you run **msiexec**, you should append **/qn** or **/qb** (**msiexec** options) to the command line to suppress the installation program user interface. The **/qn** option specifies a silent install, so no user interface appears. The **/qb** option also hides the user interface but it displays a progress bar.
- The procedures in this topic are valid with Windows installations only. For information about installing Pulse on OS X endpoints, see [“Installing the Pulse Secure Client on OS X Endpoints Using a Preconfiguration File”](#).

You run the Pulse preconfigured installer program with **msiexec** (the command line for launching .msi programs on Windows platforms) and specify the following options.



Note: Command-line options CONFIGFILE is case sensitive and must be all caps.



Note: If the path to the pulsepreconfig file includes spaces, be sure to use quotes around the path.

- **CONFIGFILE**—This property specifies a configuration file to be imported into Pulse during installation. The property must include the full path to the configuration file. For example:

```
msiexec /i PulseSecure.x86.msi CONFIGFILE="c:\temp\my configuration..pulsepreconfig "
```

Examples

To install Pulse on a 32-bit Windows endpoint using a configuration file:

```
msiexec /i PulseSecure.x86.msi CONFIGFILE=c:\temp\myconfiguration.pulsepreconfig /qb
```

To install Pulse on a 64-bit Windows endpoint using a configuration file:

```
msiexec /i PulseSecure.x64.msi CONFIGFILE=c:\temp\myconfiguration.pulsepreconfig /qb
```

Repairing a Pulse Installation on a Windows Endpoint

Pulse Secure client uses an MSI installer, which supports a repair function. If problems with Pulse on a Windows endpoint indicate missing or damaged files or registry settings, the user can easily run the installation repair program. The repair program performs a reinstallation and replaces any missing files. The repair program does not install any files that were not part of the original installation. For example, if the file that holds Pulse connection configurations is damaged, the file installed by the repair program does not replace any Pulse connections that were created by the user or deployed to the endpoint after the original Pulse installation.

To repair a Pulse installation on a Windows endpoint:

1. On the Windows endpoint where Pulse is installed, click **Start > Programs > Pulse Secure > Repair Pulse Secure**.
2. Follow the prompts for the installation wizard.

When the program is finished, you might be prompted to reboot the system.

Related Documentation

- [Installing the Pulse Secure Client on OS X Endpoints Using a Preconfiguration File](#)

Installing the Pulse Secure Client on OS X Endpoints Using a Preconfiguration File

The following procedures apply to OS X installations only.

After you create client connection sets and specify the connections to include within a client component set, you can create a preconfiguration file with all the connections you want to distribute with the Pulse client. After you run the Pulse installer on the endpoint, you run a special command that imports the settings from the preconfiguration file into Pulse.

To create a preconfigured Pulse installer for distribution to OS X endpoints:

1. Select **Users > Pulse Secure > Connections** and create a connection set with the connections that you want to distribute.
2. Select **Users > Pulse Secure > Components**.
3. If necessary, create a new component set with the connection sets you want to distribute.

The All components or No components options apply to Web-based installations only. The Pulse installation program for OS X always installs all components.

4. Select the check boxes next to the component sets that you want to distribute.
5. Click **Download Installer Configuration**.

You are prompted to save the preconfiguration. You can also specify the name of the target Pulse server for the connections, which enables you to quickly create multiple configuration files that are the same except for the target server.

The default filename of the **.pulsepreconfig** file is the name of the selected component set. Make note of the filename and location where you put the file. The preconfiguration file must be available to the clients either through a network share or distributed along with the Pulse Secure installer file.

6. Select **Maintenance > System > Installers**.
7. Download the Pulse Secure installer, **Pulse Secure installer (Macintosh)**.

Installing the Pulse Client on OS X Endpoints Using Command-Line Options

The Pulse Secure installer includes the Pulse client and all of the software components for all of the Pulse services. The preconfiguration (**.pulsepreconfig**) file contains the definitions of the Pulse connections that provide client access to specific Pulse servers and services. After you distribute the Pulse installation package, you must first run the installer, and then run a separate program called **jamCommand**, which imports the settings from the **.pulsepreconfig** file. The **jamCommand** program is part of the Pulse installation.

The Pulse file you download from the Pulse server is in compressed (**.dmg**) format. You must unpack the file before you run the Pulse installation program.

The following steps include sample commands to install Pulse on an OS X endpoint and then import Pulse connections from a **.pulsepreconfig** file.

1. Run the Pulse installation program:

```
sudo /usr/sbin/installer -pkg <full-path-to-the-pulse-install-package> -target /
```

2. Import the settings from the **.pulsepreconfig** file:

```
/Applications/PulseSecure.app/Contents/Plugins/JamUI/.jamCommand -importfile /Users/<user profile>/<pre-config file location on local disk>/<preconfig file name>
```

Related Documentation

- [Installing the Pulse Secure Client on Windows Endpoints Using a Preconfiguration File](#)
- [jamCommand Reference](#)

Pulse Secure Command-line Launcher

The Pulse Launcher (pulselauncher.exe) is a standalone client-side command-line program that allows you to launch Pulse and connect to or disconnect from a Pulse server (Pulse Connect Secure or Pulse Policy Secure) without displaying the Pulse graphical user interface.

Pulse Launcher Usage Notes:

- Pulse Launcher runs on Windows 32-bit and 64-bit endpoints.
- The Pulse Launcher program, pulselauncher.exe, is installed as part of a Pulse client installation in **Program Files\Common Files\Pulse Secure\Integration** or **Program Files (x86)\Common Files\Pulse Secure \Integration**.
- Pulse Launcher works only for the Connect Secure or Policy Secure (L3) connection type. Pulse Launcher does not support the SRX Series or Policy Secure (802.1X) connection types.
- The Pulse Launcher program does not support the role mapping option that prompts a user to select from a list of assigned roles. If you use the Pulse Launcher and more than one role can be assigned to a user, you must configure the role mapping settings for the realm to merge settings for all assigned roles. If the realm settings require the user to select a role, the Pulse Launcher command fails and exits with return code 2.
- Pulse Launcher does not support secondary authentication.

To use the Pulse Launcher program:

1. Write a script, batch file, or application.
2. Include a call to the Pulse Launcher executable, pulselauncher.exe.
3. Include logic in your script, batch file, or application to handle the possible return codes.

Table 18 lists the Pulse Launcher arguments.

The following command shows the complete pulselauncher.exe command syntax:

```
C:\Program Files (x86)\Common Files\Pulse Secure\Integration>pulselauncher.exe
```

```
pulselauncher [-version|-help|-stop] [-url <url> -u <user> -p <password> -r <realm>] [-d <DSID> -url <url>] [-cert <client certificate> [-usesystemcert] -url <url> -r <realm>] [-signout|-suspend|-resume -url <url>] [-t Time(Seconds min:45, max:600)]
```

Table 18: Pulse Launcher Arguments

Argument	Action
-version	Display the Pulse Launcher version information, then exit.
-help	Display available arguments information.
-stop	Stop Pulse and disconnect all active connections.
-url <url>	Specify the Pulse server URL.
-u <user>	Specify the username.
-p <password>	Specify the password for authentication.
-r <realm>	Specify the realm on the Pulse server.
-d <DSID>	Passes a cookie to Pulse Launcher for a specified Pulse server from another authentication mechanism when Pulse Launcher starts. When you use the -d argument, you must also specify the -url argument to specify the Pulse server.

Argument	Action
-cert <client certificate>	<p>Specify the certificate to use for user authentication. For <client certificate> use the string specified in the Issued To field of the certificate. When using the -cert argument, you must also specify the -url and -r <realm> arguments.</p> <p>To use certificate authentication with the Pulse Launcher program, you must first configure the Pulse server to allow the user to sign in via user certificate authentication. You must also configure a trusted client CA on the Pulse server and install the corresponding client-side certificate in the Web browsers of end-users before running the Pulse Launcher.</p> <p>If the certificate is invalid, the Pulse Launcher displays an error message on the command line and logs a message in the log file.</p> <p>NOTE: If Pulse is launched through a browser, the browser handles certificate verification. If Pulse is launched through an application on Windows, the application handles certificate verification. If Pulse is launched through the Pulse Launcher on Windows, Pulse Launcher handles the expired or revoked client certificates.</p>
-usesystemcert	access certificates located in system store allowing for certificate- based authentication
-signout <url>	Signout disconnects and signs out from a specific server. Suspend puts an active connection in the suspend state without removing the session information from the server. Resume restores a suspended connection. Pulse can have multiple simultaneous connections so the -url argument is required when you use -signout, -suspend, or -resume.
-suspend <url>	
-resume <url>	
-t <timeout in seconds>	The amount of time allowed for the connection to take place before the attempt fails. Min = 45 (default), Max = 600.

Table 19: Pulse Launcher Return Codes

Code	Description
-1	Pulse is not running.
0	Success.
1	A parameter is invalid.
2	Connection has failed or Pulse is unable to connect to the specified gateway.
3	Connection established with error.
4	Connection does not exist. Example: the command attempts to sign out from a server that has not been added on the Pulse UI.
5	User cancelled connection.
6	Client certificate error.
7	Timeout error.
8	No user connection allowed from Pulse UI.
9	No policy override from Pulse UI.
25	Invalid action for current connection state. This error code would occur if you tried to suspend or resume a connection that was disconnected.
100	General error.



Note: The return codes specified in Table 20 refer to the executable's return codes. On Windows, you can display the last error level with "echo %errorlevel%" (without quotes). On OSX, the command is "echo \$?" (without quotes).

Examples

The following command is a simple login application that captures the credentials the user enters, and passes the credentials as arguments to `pulselauncher.exe`:

```
pulselauncher.exe -u JDoe -p my$Pass84 -url https://int-company.portal.com/usr -r Users
pulselauncher return code: 0
```

The following Pulse Launcher example shows a certificate authentication:

```
pulselauncher.exe -url https://int-company.portal.com/usr -cert MyCert -url https://int-company.portal.com/usr -r Users
pulselauncher return code: 0
```

The following example shows a command to use Pulse Launcher to specify a cookie (-d) for a specific Pulse server (-url):

```
pulselauncher.exe -d 12adf234nasu234 -url https://int-company.portal.com/usr
pulselauncher return code: 0
```

Related Documentation

- [Using jamCommand to Import Pulse Secure Connections](#)
- [jamCommand Reference](#)

Using jamCommand to Import Pulse Secure Connections

The `jamCommand.exe` program is a command line program that imports a `.pulsepreconfig` file into the Pulse client. The `jamCommand` program is available for Windows (Vista, Windows 7, and Windows 8) and macOS.

A `.pulsepreconfig` file includes Pulse connection parameters. You can create a `.pulsepreconfig` file on the Pulse server, and then use it as part of a Pulse installation to ensure that Pulse users have one or more Pulse connections when they start Pulse for the first time.



Note: One typical use case for `jamCommand` on a Windows endpoint is to first run `jamCommand` to import one or more Pulse connections from a `.pulsepreconfig` file, and then run `pulselauncher.exe` to start Pulse.

To install Pulse connections using `jamCommand`:

1. Create a `.pulsepreconfig` file on the Pulse server.

In the Pulse server admin console, click **Users > Pulse Secure > Components**.

2. Select the component sets you want, and then click **Download Installer Configuration**.
3. Distribute the `.pulsepreconfig` file to the Pulse endpoints.
4. Run `jamCommand` with the `.pulsepreconfig` file as an option. For example:

On Windows:

```
%Program Files%\Common Files\Pulse Secure\JamUI\jamCommand -importfile myfile.pulsepreconfig
```

On macOS:

```
/Applications/PulseSecure.app/Contents/Plugins/JamUI/.jamCommand -importfile /Users/<user profile>/<pre-config file location on local disk>/<preconfig file name>
```

If the Pulse client is running when you run `jamCommand`, the new Pulse connection or connections appear immediately. The connection name appears as it was defined when you created the connection in the Pulse server admin console.

Related Documentation

- [Pulse Secure Command-line Launcher](#)
- [Installing the Pulse Secure Client on Windows Endpoints Using a Preconfiguration File](#)
- [Installing the Pulse Secure Client on OS X Endpoints Using a Preconfiguration File](#)
- [jamCommand Reference](#)

jamCommand Reference

Syntax

`jamCommand [-import [script]] [-tray] [-log[level]]`

`/import`

`/importFile <script>`

`/tray`

`/log <level>`

`/stop`

`/suspend <GUIDS>`

`/resume <GUIDS>`

`/resume`

`/brand <brandfile>`

`/unbrand`

`/norestart`

Release Information

Introduced with Pulse R1.0.

Pulse R3.1 introduced the suspend and resume options.

Pulse R4.0.3 introduced new options to support the Pulse Secure Client Customization tool.

Description

The jamCommand.exe program is a command line program that imports a .pulsepreconfig or a **Branding.PulseBrandingPackage** file into the Pulse client. The jamCommand program is available for Windows and macOS.

Options

import—Import script from the default memory-mapped file.

importFile <script>—Import script from the specified file.

tray—Launch the tray notify application.

log—Set the global log level.

stop—Stop the Pulse UI.

suspend <GUIDS>—Suspend the Pulse UI.

resume <GUIDS>—Resume a suspended Pulse UI.

brand <brandfile>—Install the Pulse user interface changes defined in the Pulse branding file.

unbrand—Remove the changes applied by the Pulse branding file.

norestart—Do not restart Pulse after applying the Pulse branding file.

Required Privilege Level administrator

Related Documentation

- [Installing the Pulse Secure Client on Windows Endpoints Using a Preconfiguration File](#)
- [Installing the Pulse Secure Client on OS X Endpoints Using a Preconfiguration File](#)
- [Customizing Pulse Secure Client Overview](#)

CHAPTER 7 Customizing the Pulse Secure Desktop Client

- Customizing Pulse Secure Client Overview
- BrandPackager Workflow
- Setting Up the Pulse Secure Client Customization Environment
- Initializing the Pulse Secure Client Customization Environment
- Importing an Existing Customized Pulse Package
- Editing Pulse Secure Client User Interface Labels
- Editing Pulse Secure Client Messages
- Adding Custom Graphics to Pulse Secure Client
- Customizing Pulse Secure Client for Windows Online Help
- Customizing Pulse Secure Client for Apple OS X Online Help
- Validating Customizations to the Pulse Secure Client
- Building the New Pulse Secure Client Package
- Testing the Pulse Secure Client Package
- Installing or Upgrading Pulse for Windows with a Branding Package
- Installing or Upgrading Pulse for Apple OS X with a Branding Package
- Installing a Branding Package Only

Customizing Pulse Secure Client Overview

The Pulse Secure client customization tool (**BrandPackager**) enables you to customize the appearance of the Pulse Secure client Windows and Apple OS X clients. You can add your own identity graphic to the Pulse splash screen, to the program interface, and to Windows credential provider tiles. [Figure 111](#) shows graphic customizations applied to the Pulse for Windows client. You can also customize error and informational message text, the text that appears in dialog boxes and on buttons, and make limited changes to Pulse online Help. For example, you might want to add your help desk phone number to Pulse error messages and the Pulse online Help.

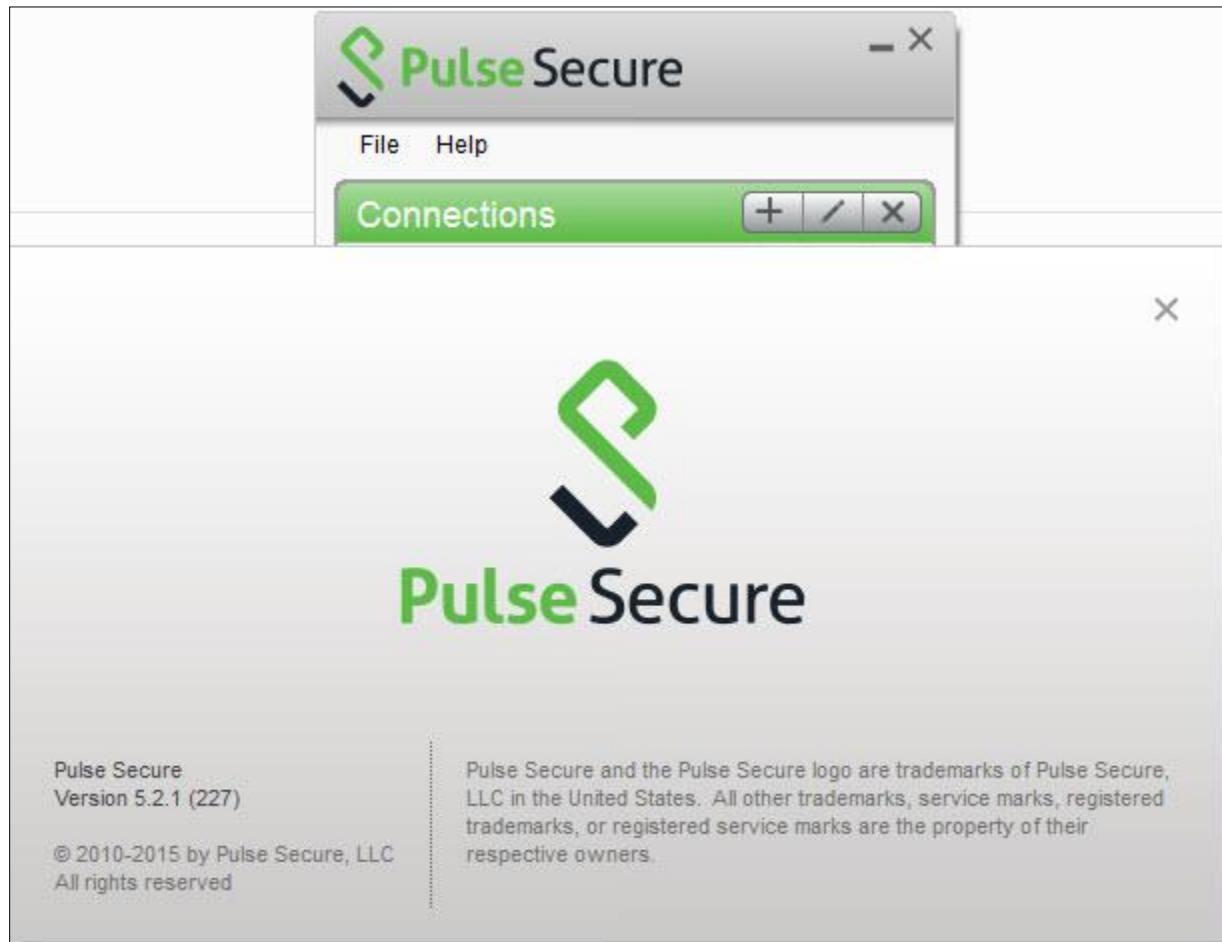
BrandPackager is available for download from [PulseSecure.net](https://pulsesecure.net).

BrandPackager runs on Windows only, but you use it to create the package files for Pulse Windows and Pulse OS X clients. A package file contains your edits to Pulse resource files. The edited resource files are installed into a special folder on the client. When the Pulse client needs to access a particular file, it checks this special folder first and uses the file if it is present. If Pulse does not find the file there, it uses the file that resides in the normal Pulse resource file location.

For Windows, you deploy the package to endpoints and use an MSIEXEC command-line installation option to instruct the installation program to apply your package file on the endpoint. For OS X, you copy the package file, the Pulse installation program, and a script file from the **BrandPackager** file set to an OS X computer, and then use them to add the package file to the Pulse installation file.

You can apply your changes to new or upgrade installations. You can also apply your customizations to an existing Pulse installation without installing or upgrading Pulse. Your changes to the Pulse user interface, message text, and online Help persist through normal client software upgrades.

Figure 111: Pulse Secure Client Interface and Splash Screen with Branding Graphics



BrandPackager Usage Notes

- **BrandPackager** supports Pulse for Windows and Pulse for macOS clients.
- **BrandPackager** is compatible with Pulse Secure client R5.0 or later.
- Pulse client customizations cannot be installed through Pulse Web portal (server) installations.
- When you edit Pulse resource files, you must preserve the UTF-8 encoding. UTF-8 files include 3 bytes {0xEF, 0xBB, 0xBF}, the Byte Order Mark (BOM), at the beginning of the file.
- The Pulse interface and the online Help include separate resource files for each of the supported languages. If you make a change in the English file, you should make the same change in the files for the other languages that you support in your environment. If you do not do so, then the edited English version is always used.
- Pulse online Help can include new information with each new release. If you edit a Help topic, your changes are retained during a Pulse client upgrade. However, if Pulse Secure changes that topic in the new release, that new information will not be available, because your edited topic will be used instead. For this reason we recommend that you make only limited changes to the online Help. For example, you can change the topic that describes how to contact customer support to direct users to contact your own help desk.

Related Documentation

- [BrandPackager Workflow](#)

BrandPackager Workflow

To create a rebranded Pulse client, you use the **BrandPackager** tool. The following procedure summarizes the steps from tool installation to client deployment. See the related documentation list for links to detailed information about the steps that are summarized here.

1. Download **PulseBrandingTools.zip** from [PulseSecure.net](https://pulsesecure.net). Create a folder on a Windows 7 or Windows 8 computer for **PulseBrandingTools.zip**, and then unzip it. Make sure that the host computer has Pulse installed, and that the version of Pulse is the one that you want to customize and distribute to users.

Set up the customization environment by installing 7Zip, a free open-source archive file program, and by running the **BrandPackager** initialization command to copy Pulse resource files to local work folders. To edit an existing package file, first import the file as part of the initialization process.

2. Edit the Pulse user interface files as needed.
3. Edit the Pulse message text files as needed.
4. Add your customization graphics.
5. Optionally, edit the Pulse online Help. There are separate procedures for the Windows and OS X online Help systems.
6. Run the **BrandPackager** script file to verify the structure of your changes and to create your package files.
7. Test your packages. The **BrandPackager** tool set provides a script to quickly activate your changes on the local machine for testing.
8. Deploying the package file is different depending on the platform:
 - For a Windows deployment, you install the package file by using an MSIEXEC command option when you run the Pulse installer.
 - For an OS X deployment, you copy the branding package, the default Pulse for OS X installation file (PulseSecure.dmg), and ConfigureInstaller to the Mac, and then run ConfigureInstaller. ConfigureInstaller is a Python script that adds the package file to the Pulse installation program. You can then run the Pulse OS X installation.

Related Documentation

- [Setting Up the Pulse Secure Client Customization Environment](#)
- [Initializing the Pulse Secure Client Customization Environment](#)
- [Editing Pulse Secure Client User Interface Labels](#)
- [Editing Pulse Secure Client Messages](#)
- [Adding Custom Graphics to Pulse Secure Client](#)
- [Customizing Pulse Secure Client for Windows Online Help](#)
- [Customizing Pulse Secure Client for Apple OS X Online Help](#)
- [Validating Customizations to the Pulse Secure Client](#)
- [Building the New Pulse Secure Client Package](#)
- [Testing the Pulse Secure Client Package](#)
- [Installing or Upgrading Pulse for Windows with a Branding Package](#)
- [Installing or Upgrading Pulse for Apple OS X with a Branding Package](#)

Setting Up the Pulse Secure Client Customization Environment

The Pulse **BrandPackager** customization tool must be run on a Windows 7 or Windows 8 computer that has Pulse 5.0 or later installed. Make sure that the Pulse installation includes all Pulse components to ensure that you have access to all of the Pulse resource files. **BrandPackager** creates the package files for Pulse Windows and Pulse OS X clients. A package file contains your edits to Pulse resource files.

To create the Pulse client customization environment:

1. If you have not already done so, download **PulseBrandingTools.zip** from [PulseSecure.net](https://pulsesecure.net). Create a folder for **PulseBrandingTools.zip**, and then unzip it. Make sure that the host computer has Pulse installed, and that the version of Pulse is the one that you want to customize and distribute to users.
2. Install 7Zip.

7Zip is a free open-source archive file program. It is used during the process of creating the Pulse customization package. You can download 7Zip from <http://7-zip.org/>.
3. If you have not already done so, install Pulse Secure client 5.0 or later on the endpoint where you will do the Pulse customization work.

Related Documentation

- [Initializing the Pulse Secure Client Customization Environment](#)

Initializing the Pulse Secure Client Customization Environment

The message text and user interface strings that appear in the Pulse client reside in text files that reside in different Pulse installation directories. After you install the **BrandPackager** tool, you run an initialization command that copies all the strings from the Pulse installation directories to two language-specific files in a reference directory called **StringReference**. The Pulse resource files are identical on Windows and OS X installations so the files from your Pulse Windows installation can be used for both Windows and OS X customizations.

During initiation, the Pulse customization tool creates the **PulseBranding** directory and copies Pulse strings from an active Pulse installation to the **StringReference** directory area for customization.

BrandPackager copies files from the local Pulse installation, so make sure that you have the Pulse version installed that you want to customize and distribute.

Make sure that the Pulse installation includes all Pulse components. You can download the Pulse installation program from a Pulse Policy Secure server or from a Pulse Connect Secure server. You can configure and include Pulse connections in the installation before you edit the Pulse client files. For more information on Pulse installation options, see the [Pulse Secure documentation](#).

To initialize the Pulse customization environment:

1. Run the following command:

BrandPackager -init

The **-init** option does not overwrite files. If there is already a **PulseBranding** directory, only missing files are written to it.

By default, the Pulse online Help files are not included. To include the Help files, specify the **-help** option:

BrandPackager -init -help

The online Help files are different between Windows and OS X. **BrandPackager** uses the Windows files from the local Pulse installation. The OS X files are included as part of the **BrandPackager** file set. The **-help** option creates two directories. The **help** directory holds the Windows files. The **PulseSecureHelp.Help** directory holds the OS X online Help files.

You can run **BrandPackager -init -help** if you have already run the **-init** option and want to just add the Help files.

Localized files in the **StringReference** directory are identified by a language identifier:

- DE – German
- EN – English
- ES – Spanish
- FR – French

- IT – Italian
- JA – Japanese
- KO – Korean
- PL – Polish
- ZH-CN – Chinese (Simplified)
- ZH – Chinese (Traditional)

Related Documentation

- [Editing Pulse Secure Client User Interface Labels](#)
- [Editing Pulse Secure Client Messages](#)
- [Adding Custom Graphics to Pulse Secure Client](#)
- [Customizing Pulse Secure Client for Windows Online Help](#)
- [Customizing Pulse Secure Client for Apple OS X Online Help](#)
- [Validating Customizations to the Pulse Secure Client](#)

Importing an Existing Customized Pulse Secure Client Package

If you already have a customized **BrandPackager** package, you can import it and make further changes to it without starting over. Also, changes to Pulse Help are not retained during a Pulse software upgrade operation. You should import the old package that has the Help file changes, create a new package, and then include that with the upgrade.



Note: If you are upgrading to a new major release of Pulse, make sure you have the latest version of BrandPackager before you create a new BrandPackager package.

To import an existing customized **BrandPackager** package into the **PulseBranding** directory:

1. Open a Command Prompt window and make the PulseBranding directory your working directory.
2. Run the following commands:

BrandPackager -init

BrandPackager -import <filename>

The **-import** option must include the filename of your existing **BrandPackager** package file. For example:

BrandPackager -import C:/Staging/PulseWin.PulseBranding

If your original **BrandPackager** package included changes to the online Help, run the optional **-help** option:

BrandPackager -init -help

BrandPackager -import <filename>

The **-import** option overwrites any files in the **PulseBranding** directory. The program prompts you for confirmation before it makes any changes.

Related Documentation

- [Editing Pulse Secure Client User Interface Labels](#)
- [Editing Pulse Secure Client Messages](#)
- [Adding Custom Graphics to Pulse Secure Client](#)
- [Customizing Pulse Secure Client for Windows Online Help](#)
- [Customizing Pulse Secure Client for Apple OS X Online Help](#)
- [Validating Customizations to the Pulse Secure Client](#)

Editing Pulse Secure Client User Interface Labels

You can modify any text string that appears in the Pulse user interface. Pulse user interface strings reside in the **StringReference\PulseResource_XX.txt** file. Your modified strings must reside in the **PulseBranding\BrandingResourceCatalog_XX.txt** file. (XX indicates the language.)



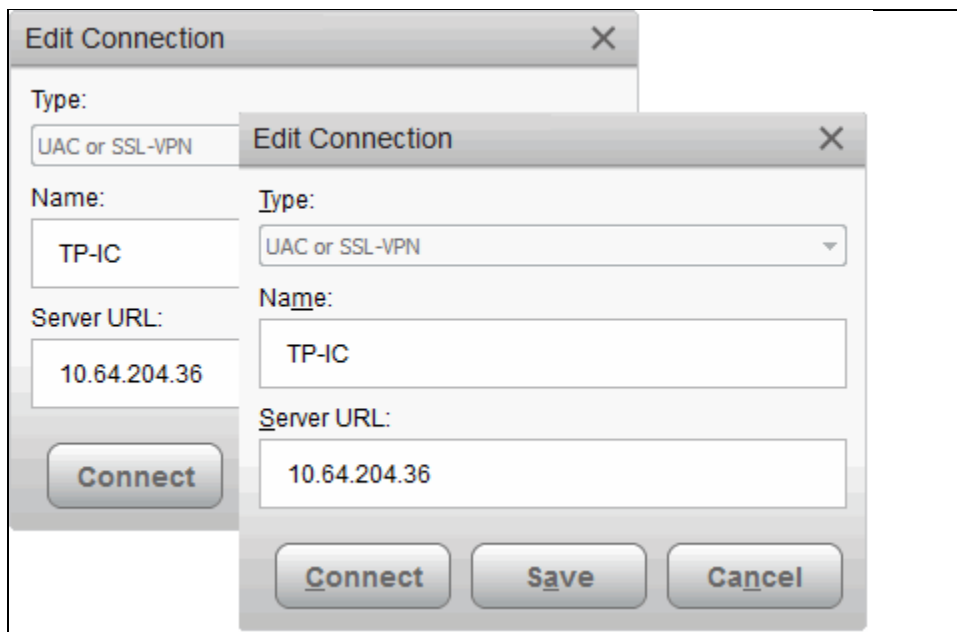
Note: If your Pulse environment uses Security Assertion Markup Language (SAML) for a Single Sign-on (SSO) authentication environment, the Pulse user sees a credential dialog box that is served from the Pulse server instead of the local Pulse client credential dialog box. The sign-in page is defined as part of the sign-in policy on the Pulse server and the Pulse client embeds the sign-in page within a Pulse client dialog box. To change the appearance of the SAML credential dialog, you must edit or create a new sign-in page on the Pulse server.

The BrandingResourceCatalog files hold only the strings you modify. The default strings in their normal files are used for all strings that you do not modify.

The following procedure describes the workflow for modifying user interface strings using the English language version of the Pulse Edit Connection dialog box as an example:

1. Start Pulse and then display the Pulse string that you want to modify. For example, in the Pulse main window, select a connection and then click **File > Connection > Edit**. Press the Alt key to show shortcut characters.

Figure 112: Pulse Dialog with Shortcut Keys Underlined



2. Take a screen shot of the screen that you want to modify.

The screen shot is not required but it can help you maintain or create a new shortcut character when you edit the string in the catalog file. It is good practice to keep track of what you change so you can verify your changes later.

3. Find the string that you want to modify.

Search **StringReference\PulseResource_EN.txt** for the string. The string might appear more than once. For example, the string **Server URL** appears twice as a value in **PulseResource_EN.txt** because that string appears in two different dialog boxes. In general, the resource ID indicates where the value is used.

Figure 113: StringReference

```
;IDS_CONNECTION_DLG_ST_NAME
[183]
Value = Name:

;IDS_CONNECTION_DLG_ST_URL
[184]
Value = &Server URL:

;IDS_CONNECTION_DLG_BTN_CONNECT
[185]
Value = &Connect
```

Many strings use an ampersand (&) to designate a keyboard shortcut key. The ampersand causes the character that follows it to appear as an underlined character in the user interface. The presence of the ampersand can affect your results when you use the editor's search function.

4. Open **PulseBranding\BrandingResourceCatalog_EN.txt** with a text editor.
5. Copy the string that you want to edit from **PulseResource_EN.txt** to **BrandingResourceCatalog_EN.txt**. Be sure to copy/paste the entire entry. For example:

```
;IDS_CONNECTION_DLG_ST_URL [184] Value = &Server URL:
```

6. Modify the string in **BrandingResourceCatalog_EN.txt**. For example:

```
;IDS_CONNECTION_DLG_ST_URL [184] Value = &Server URL:
```

Modify only the value. Do not change the string identifiers, **;IDS_CONNECTION_DLG_ST_URL** and **[184]**.

We suggest that you keep the same letter for the shortcut to avoid a conflict with other strings on the screen. If the shortcut key letter does not appear in the new string, you can include it by putting it in parentheses. For example, the following entries show how to change Close to Exit and retain the "C" as a shortcut key:

```
;IDS_MAIN_DLG_BTN_CANCEL
[188]
Value = &Close

;IDS_MAIN_DLG_BTN_CANCEL
[188]
Value = Exit(&C)
```

You should change the shortcut letter only if you are certain that the new letter is not used elsewhere in that dialog box.

Each shortcut key on a screen must be unique. You can eliminate the shortcut by deleting the ampersand. However, shortcut keys are a part of good user interface design.

7. Edit that same resource ID in each of the language files that your organization supports.

The Pulse interface includes separate files for each of the 10 supported languages. If you make a change in the English file, you should make the same change for the other languages that you support in your environment. If you do not do so, then the edited English version is always used.

After initialization, there are two files for each language in the **StringReference** directory:

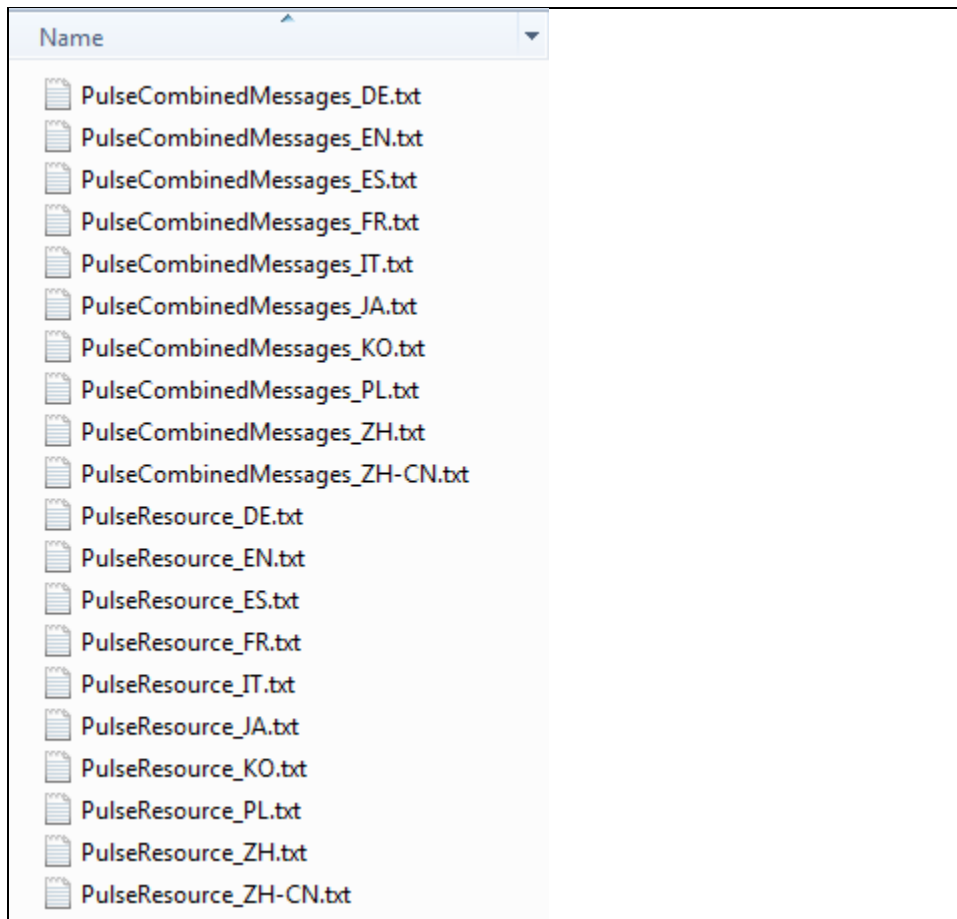
- **PulseCombinedMessages_XX.txt**

Message catalog files hold the text that appears in the Pulse program interface and dialog boxes.

- **PulseResource_XX.txt**

Resource catalog files hold the text that appears in Pulse message boxes.

Figure 114: BrandPackager StringReference Directory



To customize a particular string, you find the string you want to customize in **PulseCombinedMessages_XX.txt** or **PulseResource_XX.txt**, and then copy and paste that entire string and its resource ID to a corresponding resource or message file in the **PulseBranding** directory, where you edit it. This directory holds all of the files that make up your customization package.



Note: You must use a text editor, such as Visual Studio IDE or Notepad++ that retains the byte order mark (BOM) in the resource files. (Notepad++ is free open source software available at <http://notepad-plus-plus.org/>)



Note: See the Sample directory for an example of a customized Pulse client file set.

Related Documentation

- [Editing Pulse Secure Client Messages](#)

- [Adding Custom Graphics to Pulse Secure Client](#)
- [Customizing Pulse Secure Client for Windows Online Help](#)
- [Customizing Pulse Secure Client for Apple OS X Online Help](#)
- [Validating Customizations to the Pulse Secure Client](#)

Editing Pulse Secure Client Messages

Pulse message strings reside in the **StringReference\PulseCombinedMessages_XX.txt** file. Modified message strings must reside in the **PulseBranding\BrandingMessageCatalog_XX.txt** file. (XX indicates the language.)

The BrandingMessageCatalog files hold only the strings that you modify. The default strings in the installed resource files are used for all strings that you do not modify.

It is not always possible to set up the conditions that cause a particular message to appear in Pulse. Browsing the contents of **BrandingMessageCatalog_XX.txt** is the easiest way to identify the strings you might want to change.

You can use HTML tags within the BrandingMessageCatalog entries. For example, you can use **** and **** tags to make text appear in bold type. You can use **** tags to include a link to other HTML text you want. Make sure that your link displays the text in a new window. For example:

```
<A HREF="Http://www.myserver/my-messsge.html" target="_blank">
```

Keep in mind that the Pulse client might not be connected to the Internet when the error occurs.

Each message includes a short description and a long description. The short description is shown as a title to the longer description. There are no limits to the number of characters that you can include as the long description. However, the long description must be on one line in the message catalog file. Use HTML **
** and **</br>** tags to insert line breaks when the message is displayed.

To modify a message:

1. Find the string that you want to modify.

Search **PulseCombinedMessages_XX.txt** for the string. In general, the resource ID indicates where the value is used.

2. Open **PulseBranding\BrandingMessageCatalog_XX.txt** with a text editor.
3. Copy the string that you want to edit from **StringReference\PulseCombinedMessages_XX.txt** to **BrandingMessageCatalog_XX.txt**. Be sure to copy/paste the entire entry. For example:

```
[1731] ;kMsgCommonCertTrustPulseAuthServerIdentityNotFoundShort-desc = Authentication server not trusted.  
Long-desc = Authentication server identity not found in client's "Trusted Server List". Contact your network  
administrator.
```

4. Modify the string in **BrandingMessageCatalog_XX.txt**. For example:

```
[1731] ;kMsgCommonCertTrustPulseAuthServerIdentityNotFoundShort-desc = Authentication server not trusted.  
Long-desc = Authentication server identity not found in client's "Trusted Server List". Contact the Help Desk at  
Ext.50123.
```

5. Edit that same resource ID in each of the language files that your organization supports.

The Pulse interface includes separate files for each of the 10 supported languages. If you make a change in the English file, you should make the same change for the other languages you support in your environment. If you do not do so, then the edited English version is always used.

Related Documentation

- [Editing Pulse Secure Client User Interface Labels](#)

- [Adding Custom Graphics to Pulse Secure Client](#)
- [Customizing Pulse Secure Client for Windows Online Help](#)
- [Customizing Pulse Secure Client for Apple OS X Online Help](#)
- [Validating Customizations to the Pulse Secure Client](#)

Adding Custom Graphics to Pulse Secure Client

The **PulseBranding** directory also includes default graphics. To add your custom graphics to the Pulse interface, simply replace the default graphics with your custom graphics.

You can add a graphic to the following areas:

- Next to the Pulse logo on the main screen
- In dialog boxes
- On the About screen
- On the Pulse splash screen



Note: The Pulse connection set properties, which you define on the Pulse server, include an option to suppress the Pulse splash screen.

The **PulseBranding** directory includes two graphics:

- **BrandingLogo.png**—Appears on the Pulse splash screen and program interface. The default **BrandingLogo.png** file is an empty file with a transparent background. For best results, your graphic image should have a transparent background. The file must be a PNG file.

The default **BrandingLogo.png** file is 19 by 52 pixels. The maximum height is 37 pixels, which corresponds to the size of the Pulse logo. Maximum width is 100 pixels. A graphic larger than the recommended size might be clipped or it could obscure other graphic elements.

- **BrandingCredProv.png**—Appears as the image on credential provider tiles.

To add a custom graphic:

1. Replace **PulseBranding\BrandingLogo.png** with your graphic.
2. Replace **PulseBranding\BrandingCredProv.png** with your graphic.

If you do not want to include a custom graphic, you should delete default graphics from **PulseBranding**.

The Pulse online Help provides reference and procedural information for users. Pulse users can access the Help by clicking the Help button in the Pulse program interface. Pulse online Help is a collection of standard HTML files with CSS formatting and javascript navigation. Updating the Help requires knowledge of basic HTML coding. To edit the online Help, you must include the Help when you initialize the Pulse customization environment.



Note: If you edit a Help topic, your edited topic is used instead of the original topic. Your edited topic is retained during an upgrade. Pulse online Help can include new information with each new release. If Pulse Secure changes a topic in the new release, that new information will not be available because your edited topic is used instead. To avoid this problem, we recommend that you make only the Help topic changes described in this guide.

You might want to edit that topic and substitute your own help desk contact information. Use an HTML editor to make your changes. Do not change the filename or any of the javascript code within the topic.

Pulse Help includes the following language versions:

- DE – German
- EN – English

- ES – Spanish
- FR – French
- IT – Italian
- JA – Japanese
- KO – Korean
- PL – Polish
- ZH-CN – Chinese (Simplified)
- CN – Chinese (Traditional)

Be sure to edit the same topics for all the languages that you support.

The Pulse Help viewer includes a menu item labeled Feedback, which links to a documentation comments page on [PulseSecure.net](https://www.pulsesecure.net).

To change the Feedback destination URL or to remove the menu item:

1. Open `j_header.html` with an HTML editor.
2. Search for the following string:
<https://www.pulsesecure.net/support/>
3. Either edit or remove the link.

Related Documentation

- [Editing Pulse Secure Client User Interface Labels](#)
- [Editing Pulse Secure Client Messages](#)
- [Adding Custom Graphics to Pulse Secure Client](#)
- [Validating Customizations to the Pulse Secure Client](#)
- [Customizing Pulse Secure Client for Apple OS X Online Help](#)

Customizing Pulse Secure Client for Apple OS X Online Help

The Pulse online Help provides reference and procedural information for users. Pulse users can access the Help by clicking the Help button in the Apple menu bar. Pulse online Help is a collection of standard HTML files with CSS formatting. Apple Help application acceleration includes special metadata in the header of each topic and a particular directory structure to properly interact with OS X. Updating the Help requires knowledge of basic HTML coding. To edit the online Help, you must include the Help when you initialize the Pulse customization environment.



Note: If you edit a Help topic, your edited topic is used instead of the original topic. Your edited topic is retained during an upgrade. Pulse online Help can include new information with each new release. If Pulse Secure Networks changes a topic in the new release, that new information will not be available because your edited topic is used instead. To avoid this problem, we recommend that you make only the Help topic changes described in this guide.

You might want to edit that topic and substitute your own help desk contact information. The file resides in **PulseSecureHelp.Help\Contents\Resources\<language>.lproj\pages**. Filenames in OS X are case-sensitive.

Pulse for OS X online Help includes the following language versions:

- DE.lproj – German
- English.lproj – English
- ES.lproj – Spanish
- FR.lproj – French
- IT.lproj – Italian
- JA.lproj – Japanese

- KO.lproj – Korean
- PL.lproj – Polish
- TW.lproj – Chinese (Traditional)
- CN.lproj – Chinese (Simplified)

Be sure to edit the same topics for all the languages that you support.

Related Documentation

- [Editing Pulse Secure Client User Interface Labels](#)
- [Editing Pulse Secure Client Messages](#)
- [Adding Custom Graphics to Pulse Secure Client](#)
- [Validating Customizations to the Pulse Secure Client](#)
- [Customizing Pulse Secure Client for Windows Online Help](#)

Validating Customizations to the Pulse Secure Client

The validation process examines the files in the **PulseBranding** directory to ensure that they can be added to the Pulse installation package.

To validate your changes before building the **BrandPackager** package:

1. Run the following command:

BrandPackager -validate

Validation is a basic level of checking. After you build the new Pulse installation package, you should test the package before you deploy it.

Related Documentation

- [Building the New Pulse Secure Client Package](#)

Building the New Pulse Secure Client Package

The packaging process creates two package files, one for Windows and one for OS X, that include your changes. It does not include the Pulse installation files. You include a package when you install Pulse. Or you can apply your changes to a Pulse client without installing or upgrading Pulse.

To create a package:

1. Run the following command:

BrandPackager -package

When the command finishes, it creates two package files, **PulseWin.PulseBranding** and **PulseMac.PulseBranding**. To apply your changes on a Pulse endpoint, you include a package file when you install or upgrade Pulse.

Related Documentation

- [Testing the Pulse Secure Client Package](#)

Testing the Pulse Secure Client Package

Before you deploy the new Pulse installation package, you should verify that your changes work correctly. **BrandInstaller.bat** installs the **BrandPackager** package on the local machine. **BrandInstaller.bat** employs **jamCommand.exe**, which is a program that resides in the Pulse program directory.



Note: You must be an administrator to run BrandInstaller.

To install your **BrandPackager** package on the machine where you created it:

1. Run the following command:

BrandInstaller -brand

You can now view your changes on the local Pulse client to make sure that you have made all the modifications correctly. Verify the Pulse client by checking the following:

- View the main dialog and the About screen to make sure that the branding logo appears as you want.
- View the screens that contain any of the user interface strings that you changed.
- If you have updated the Pulse for Windows Help, invoke the Help to make sure your changes are correct.

If you are satisfied, you can install the package on endpoints.

Related Documentation

- [Installing or Upgrading Pulse for Windows with a Branding Package](#)
- [Installing or Upgrading Pulse for Apple OS X with a Branding Package](#)
- [Installing a Branding Package Only](#)

Installing or Upgrading Pulse for Windows with a Branding Package

You install or upgrade Pulse and apply the changes in **PulseWin.PulseBranding** to a Pulse Windows client by using Microsoft Exec (**msiexec**) and setting the **BRANDINGFILE** attribute to point to the branding file. This installation requires administrative privileges.

The following example shows the **msiexec** command to install or upgrade Pulse Secure client and to apply the customizations in **PulseWin.PulseBranding**:

```
msiexec /i c:\staging\PulseSecure.x64.msi BRANDINGFILE=c:\staging\PulseWin.PulseBranding
```

For more information on installing Pulse Secure client, see the [Pulse Secure documentation](#).

Related Documentation

- [Installing or Upgrading Pulse for Apple OS X with a Branding Package](#)
- [Installing a Branding Package Only](#)

Installing or Upgrading Pulse for Apple OS X with a Branding Package

To apply the branding package changes to an Apple OS X endpoint, you must copy the necessary files to an OS X endpoint and use them to update the Pulse installation program. You can also use this process to add Pulse configurations (a .pulsepreconfig file) to the Pulse installation program. You can then use that Pulse installation program to install or update Pulse on OS X endpoints. If the specified branding package is present in the Pulse installation program, the installation process creates the following directory:

/Library/Application Support/Pulse Secure /PulseBranding

The PulseBranding directory holds the changes you made to Pulse resource files and graphics. When Pulse must access a resource file, it checks this directory first.

To add **PulseMac.PulseBranding** to **PulseSecure.dmg**, perform the following steps on an OS X endpoint:

1. Create a directory on an OS X endpoint and copy the following files to it:
 - **PulseMac.PulseBranding**—The file created for OS X by **BrandPackager** that contains all of your client customizations. After you edit the resource files and run **BrandPackager**, **PulseMac.PulseBranding** is available in the same directory as **BrandPackager**.
 - **PulseSecure.dmg**—The Pulse installation program. You can download **PulseSecure.dmg** from the Downloads page of Pulse Connect Secure or Pulse Policy Secure.
 - **ConfigureInstaller**—A Python script that adds the package file to **PulseSecure.dmg**. **ConfigureInstaller** is available in the same directory as **BrandPackager**. Python is part of OS X 10.2 and greater and is included in the system PATH.
2. Open a terminal window and make the directory that holds **ConfigureInstaller** your current directory.
3. Run **ConfigureInstaller**. You can run **ConfigureInstaller** with no options to see the command summary:

```
python ./ConfigureInstaller
usage -s <source dmg> -b <brandingfile> -c <configfile> -t <target dmg>
usage -s <source dmg> -b <brandingfile> -t <target dmg>
usage -s <source dmg> -c <configfile> -t <target dmg>
```

The following example shows a command for adding a branding file and a Pulse config file to the Pulse installation program:

```
python ./ConfigureInstaller -s PulseSecure.dmg -b ~/Staging/PulseMac.PulseBranding -c
~/Staging/myfile.pulsepreconfig -t PulseSecure-new.dmg
```

When the operation completes successfully, the new Pulse installation program is ready for use.

For complete information on creating preconfigured Pulse connections and installing Pulse Secure client, see the [Pulse Secure documentation](#).

Related Documentation

- [Installing or Upgrading Pulse for Windows with a Branding Package](#)
- [Installing a Branding Package Only](#)

Installing a Branding Package Only

You can add or remove the contents of a Pulse branding package on a client machine by using **jamCommand**. The **jamCommand** program is part of every PulseSecure Desktop client installation. On Windows endpoints, **jamCommand** is located in the 32-bit program files directory:

Program Files (x86)\Common Files\Pulse Secure\JamUI\jamCommand.exe

On OS X endpoints, **jamCommand** is located in the Applications folder:

/Applications/PulseSecure.app/Contents/Plugins/JamUI/./jamCommand

For more information on installing Pulse Secure, see the [Pulse Secure documentation](#).



Note: The **jamCommand** program must be run with administrator privileges.

To apply the customizations in **PulseWin.PulseBranding** (Windows) or **PulseMac.PulseBranding** (OS X):

1. Run the following command:

```
jamCommand -brand
```

To remove your customized Pulse user interface from the endpoint and allow Pulse to use default strings:

2. Run the following command:

jamCommand -unbrand

jamCommand Usage Notes:

- Running **jamCommand** with the **-brand** or **-unbrand** option causes Pulse to restart. Connections are maintained and should be active after the restart. A restart is required to allow Pulse to access the customized settings. If you will be rebooting the system manually, or if there is no logged in user, then you can use the **-norestart** option. To avoid a restart when you run **jamCommand**, use the following option:

jamCommand -norestart

- **jamCommand** reports its results using the following numeric error codes:
 - 0 - Success.
 - 1 - General branding error.
 - 2 - Error deleting branding files. This error can also occur when you install new files because the first action **-brand** performs is to remove the old files.
 - 3 - Error branding Pulse. The new branding files cannot be written.
- The Pulse client version must be R5.0 or later. To verify your current version of Pulse, run **jamCommand** with no parameters. If the result (displayed in a window) shows the branding options (**-brand**, **-unbrand**, **-norestart**), then branding is supported.

The **jamCommand** errors are not written to the console. To see **jamCommand** errors, include a script that checks error codes. Additional error message information is written to the Pulse log files.

Related Documentation

- [Installing or Upgrading Pulse for Windows with a Branding Package](#)
- [Installing or Upgrading Pulse for Apple OS X with a Branding Package](#)

PART 2 Pulse Secure Client Compatibility

This section provides detailed information about the how Pulse Secure client features compare to Odyssey Access client and Network Connect software features.

- [Client Software Feature Comparison](#)

CHAPTER 8 Client Software Feature Comparison

- Comparing Odyssey Access Client and Pulse Secure Client
- Comparing Network Connect and Pulse Secure Client

Comparing Odyssey Access Client and Pulse Secure Client

Pulse Secure client is a single integrated, multiservice network client that provides the basic services provided by the older Network Connect and Odyssey Access Client software. Pulse supports dynamic connectivity and secure access control for Microsoft Windows and macOS devices, and connectivity, and mobile device management (MDM) for mobile devices, all with a simple, easy to use, elegant user experience.

Table 20 compares the features in Odyssey Access Client (OAC) and Pulse Secure client to help you transition to Pulse. For detailed information about supported platforms and installation requirements, see the *Pulse Secure Supported Platforms Guide*, which is available at <https://www.pulsesecure.net/techpubs/>

Table 20: Odyssey Access client and Pulse Secure Client Feature Comparison

Feature	Pulse Secure Client for OSX	Pulse Secure Client Windows	Odyssey Access Client
Wired/Wireless 802.1X Features			
Wired 802.1X support		Yes (with Microsoft Windows supplicant)	Yes
Auto scan lists		Yes (with Microsoft Windows supplicant)	Yes
Wireless suppression		Yes (with Microsoft Windows supplicant)	Yes
Support for Network Provider (scraping passwords, listing)		Yes	Yes
Association Mode and Encryption Methods			
Association mode support (for open, shared, WPA/WPA2)		Yes (with Microsoft Windows supplicant)	Yes
Encryption (for WEP, TKIP, AES, WEP with preconfigured key, WPA/WPA2 with pre-shared key)		Yes (with Microsoft Windows supplicant)	Yes
EAP Methods			
EAP-TLS outer authentication			Yes
EAP-TTLS outer authentication	Yes	Yes	Yes
• With EAP-JUAC inner authentication		Yes	Yes

Feature	Pulse Secure Client for OSX	Pulse Secure Client Windows	Odyssey Access Client
• With EAP-MSCHAPv2 inner authentication			Yes
• With EAP-GTC inner authentication			Yes
• With EAP-MD5 inner authentication			Yes
• With PAP inner authentication			Yes
• With CHAP inner authentication			Yes
• With MSCHAP inner authentication			Yes
• With MSCHAPv2 inner authentication			Yes
EAP-PEAP outer authentication			Yes
• With EAP-JUAC inner authentication			Yes
• With EAP-DD5 inner authentication			Yes
• With EAP-GTC inner authentication			Yes
Authentication Methods			
Prompt for user name and password	Yes	Yes	Yes
Certificate support (automatic, specific)	Yes	Yes	Yes
Certificates from smart card reader	Yes	Yes	Yes
Soft token support	Yes	Yes	Yes
Machine login support	N/A	Yes	Yes
Machine authentication followed by user authentication	N/A	Yes	Yes
Credential provider on 32- and 64-bit Windows Vista, Windows 7, and Windows 8	N/A	Yes	Yes
Pre-desktop login	N/A	Yes	Yes
Configurable UAC Layer 2 connection		Yes	Yes
Configurable connection association modes			Yes
Certifications			
FIPS compliance		Pulse SSL-VPN mode connection is fully FIPS compliant. Pulse server certificate verification and private key signing is FIPS compliant. Pulse IPSec is FIPS compatible. Pulse wireless is FIPS compatible. (WPA encryption is controlled by Windows.)	Yes
Installation and Upgrade Methods			
Auto-upgrade	Yes	Yes	Yes
Web-based installation	Yes	Yes	Yes

Feature	Pulse Secure Client for OSX	Pulse Secure Client Windows	Odyssey Access Client
Standalone installation	Yes (.dmg)	Yes (.msi)	Yes
Upgrade/coordinate with previous versions	Yes	Yes	Yes
Manual Uninstall	Yes	Yes	Yes
Browser based installation and upgrades	Yes	Yes	Yes
Diagnostics and Logging			
IPsec diagnostics and configuration		Yes	Yes
Host Enforcer			Yes
Log viewer			Yes
Logging and Diagnostics	Yes Set debug level	Yes Set debug level, set file size limits In addition to the Pulse log files, Pulse writes events to the Windows application event log. (Windows Vista and Windows 7 systems only.)	Yes
Other Features			
OPSWAT IMV support	Yes	Yes	Yes
Automatic patch remediation		Yes via SMS/SCCM	Yes via SMS/SCCM
Host Checker support	Yes	Yes	Yes
IPsec tunneling to Policy Enforcement Points with NAT-T	Yes	Yes	Yes
Access service and plug-ins	Yes	Yes	Yes
Block 3rd party EAP messages		N/A	Yes
Layer 3 authentication	Yes	Yes	Yes
Server-based pre-configuration of realm/role	Yes	Yes	Yes
Extend session duration	Yes	Yes	Yes
IC cardinality	Yes	Yes	Yes
Client-site management of clustered Pulse servers	Yes	Yes	Yes
Kerberos SSO		Yes	Yes
Initial configuration (intervention-less client provisioning)	Yes	Yes	Yes
Dynamically configurable	Yes	Yes	Yes

Related Documentation

- [Comparing Network Connect and Pulse Secure Client](#)

Comparing Network Connect and Pulse Secure Client

Pulse Secure client is an integrated, multiservice network client that replaces Network Connect (NC) and Odyssey Access Client (OAC) software. Pulse provides dynamic connectivity, access control, and security, with a simple, easy to use, elegant user experience.

Table 21 compares the features of NC to Pulse for Windows and Pulse for OS X to help you transition from NC to Pulse Secure client. For detailed information about supported platforms and installation requirements, see the *Pulse Secure Supported Platforms Guide*, which is available at <https://www.pulsesecure.net/techpubs/>

Table 21: Network Connect and Pulse Secure Client Feature Comparison

Feature	Pulse Secure Client Release 5.1		Network Connect Release 6.3	
	Mac 10.8 and later	Win 7 and later	Mac	Release 6.3 Win
Proxy Support				
	Pulse respects the system's understanding of the web proxy.			
Internet Explorer		Yes		Yes
Mozilla Firefox		Yes		Yes
Apple Safari	Yes			
Google Chrome	Yes	Yes		
Split Tunneling Options				
Disable split tunneling without route monitor	Yes	Yes		
Disable split tunneling with route monitor	Yes	Yes	Yes	Yes
Enable split tunneling with route monitors	Yes	Yes	Yes	Yes
Enable split tunneling without route monitors	Yes	Yes	Yes	Yes
Enable split tunneling with allowed access to local subnet	Yes	Yes	Yes	Yes
Disable split tunneling with allowed access to local subnet	Yes	Yes	Yes	Yes
Disable split tunneling but allow directly connected local subnet access	Yes	Yes		
Client Launch Options				
Command line launcher		Yes		Yes
Log out on connect		Pulse implements this behavior through machine authentication.		Yes
Launch as a standalone client	Yes	Yes	Yes	Yes
Launch from browser	Yes	Yes	Yes	Yes
Transport Mode				

Feature	Pulse Secure Client Release 5.1		Network Connect Release 6.3	
	Mac 10.8 and later	Win 7 and later	Mac	Release 6.3 Win
SSL fallback mode	Yes	Yes	Yes	Yes
		NOTE: If ESP is not available, the connection uses SSL. After a connection switches to SSL it does not go back to ESP until the connection is restarted.		
ESP	Yes	Yes	Yes	Yes
Other Features				
OPSWAT IMV support		Yes	Yes	Yes
Patch automatic remediation		Yes		
		via SMS/SCCM		
Host Checker support	Yes	Yes	Yes	Yes
Enhanced Endpoint Security support	NOTE: Enhanced Endpoint Security was discontinued in February 2013.			
Run configured scripts when client connects/disconnects	Yes	Yes	Yes	Yes
Modify DNS server search order based on server configuration	Yes	Yes	Yes	Yes
Reconnect automatically if connection breaks	Yes	Yes	Yes	Yes
Dial-up adapter support	Yes	Yes	Yes	Yes
3G wireless adapter support	Yes	Yes	Yes	Yes
Max/Idle Session Time-outs	Yes	Yes	Yes	Yes
IPv6	Yes	Yes		
Location awareness	Yes	Yes		
Customizable user interface, including message text (all supported languages)	Yes	yes		
Authentication				
Machine authentication		Yes		
SAML	Yes	Yes	Yes	Yes
Credential provider		Yes		Yes
Smart cards	Yes	Yes		?
Soft token Auth - RSA and others	Yes	Yes	Yes	Yes
Soft token integration - RSA		Yes		
Certificate Auth	Yes	yes	yes	Yes

Feature	Pulse Secure Client Release 5.1		Network Connect Release 6.3	
	Mac 10.8 and later	Win 7 and later	Mac	Release 6.3 Win
Login-logoff script support	Yes	Yes	Yes	Yes
Start before log on	No	Yes	No	yes
Password expiration notification	Yes	Yes	Yes	Yes
Password management (pass-through)	Yes	Yes	No	No
Logging				
Log to file	Yes	Yes	Yes	Yes
Upload log	Manual	Manual	Yes	Yes
Set logging level	Yes	Yes	Yes	Yes
Certifications				
FIPS		Pulse SSL-VPN mode connection is fully FIPS compliant. Pulse server certificate verification and private key signing is FIPS compliant. Pulse IPSec is FIPS compatible. Pulse wireless is FIPS compatible. (WPA encryption is controlled by Windows.)	Yes	

Pulse Split Tunneling

Table 22 lists the Network Connect split tunneling options and shows how they map to Pulse split tunneling options.

Table 22: Pulse Split Tunneling

NC Split Tunnel Option	Pulse Split Tunnel Setting	Route Override State	Route Monitor State
Disable split tunnel	Disabled	Yes	Yes
Disable split tunneling but allow local access	Disabled	No	No
Enable split tunnel	Enable	Yes	No
Enable split tunnel with route monitor	Enable	Yes	Yes
Enable split tunnel, allow local access	Enable	No	No

Related Documentation

- [Comparing Odyssey Access Client and Pulse Secure Client](#)

PART 3 Pulse Secure Client for Mobile Devices

- [Pulse Secure Client for Mobile Devices](#)
- [Pulse Secure Client for Apple iOS](#)
- [Pulse Secure Client for Google Android](#)
- [Pulse Secure Client for Windows Phone](#)

CHAPTER 9 Pulse Secure Client for Mobile Devices

- [Pulse Secure Client for Mobile Devices Overview](#)
- [Pulse Secure Mobile Clients and User Agent Strings](#)

Pulse Secure Client for Mobile Devices Overview

Pulse Connect Secure supports authenticated access from mobile (handheld) devices to corporate applications such as corporate e-mail and the corporate intranet through Pulse Connect Secure. The Pulse client software for mobile devices includes remote VPN capabilities. The Pulse for Android and Pulse for iOS clients support Pulse Collaboration for online meeting services.

Each supported mobile device requires that the user install the Pulse client software for the particular device type. The Pulse app is available as a free download from the app stores of the supported mobile devices. The type of secure connectivity and the supported security features vary according to what is supported on each mobile operating system.

Updated versions of each Pulse Secure mobile client are released independently of each other. Some client features are not available in earlier client releases. If necessary, the description of a particular feature includes the minimum release required for that feature. For VPN features, the Pulse client communicates with the Pulse Connect Secure. As a general rule, users should upgrade to the most recent Pulse client software version to insure compatibility with all server updates.

Pulse is supported on the following mobile devices:

- Pulse Secure Client for Apple® iOS (iPhone, iPad, and iPod Touch)
- Pulse Secure Client for Google Android™
- Pulse Secure Client for BlackBerry
- Pulse Secure Client for Windows® Phone

The *Pulse Secure Mobile Supported Platforms Guide* lists the mobile device OS versions supported by Pulse and the security features supported on each mobile device OS.

Related Documentation

- [Pulse Secure Client for Apple iOS Overview](#)
- [Pulse Secure Client for Android Overview](#)
- [Pulse Secure Client for Windows Phone Overview](#)

Pulse Secure Mobile Clients and User Agent Strings

A user agent string is how a Web server identifies the type of client that is requesting service. A server can use that information to provide content that is tailored for the client. For example, a Web server can serve content for a particular browser. Portions of the Pulse user interface on a mobile device, such as the login screen and intranet bookmarks, are Web pages served from the Pulse server and displayed by the embedded browser of the Pulse client. The appearance of these Web pages can be affected by how the Pulse server is configured to map user agent strings to specific client types.



Note: You can configure user agent strings at the role or the realm level to define policies for a user based on client type.

To view and edit the user agent string and client type pairings on your Pulse server:

1. Click **System > Configuration > Client Types**.

To edit an existing item, click an item in the table to select it. The string pattern is available for editing in a text box and you can select the client type from a list box.

To add a new user agent string and client type pairing, type a string pattern in the edit box at the top of the list, choose a client type from the list box, and then click **Add**. You can use the * and ? wildcard characters in your string. Note that user agent strings are not case-sensitive.

2. To reorder the list, click an item to select it and then use the up and down arrow buttons to move the item up or down in the list. When a browser requests access, the user agent string submitted by the browser is compared against the list starting at the top of the list and continuing down the list until the first match is reached.

The default pairing (User Agent String = "*", Client Type = "Standard HTML") is listed as the last entry in the table to ensure that it is used only when no other pairing applies. You cannot edit, delete, or reorder the default pairing.

3. When you finish making changes to the table, click **Save Changes**.

Table 23 lists the User Agent String and Client Type pairings for supported mobile devices in Pulse server software.

Table 23: User Agent String Client Type Pairings for Mobile Devices

Pulse Connect Secure Software	Apple iOS	Google Android
Release 7.0 and earlier	User Agent String = "*", Client Type = "Standard HTML"	User Agent String = "*", Client Type = "Standard HTML"
Release 7.1 and later	User Agent String = "**iPad*AppleWebKit", Client type = "iPad Optimized HTML" User Agent String = "AppleWebKit*Mobile*", Client Type = "Mobile Safari Optimized HTML (iPhone/iPod Touch) Full/Advanced/Basic"	User Agent String = "**Android*", Client Type = "Android Optimized HTML Full/Advanced/Basic"

Windows 8.1 Pro and RT introduced the Windows In-box Pulse Secure client as a part of the Windows operating system. For Windows Phone 8.1, Pulse Secure client for Windows Phone is available in the Windows Phone Store.

Windows In-box Pulse Secure client user agent string:

- Junos-Pulse/8.1.0.47666 (Windows 7) JunosPulseVpn/1.0.0.206

Junos-Pulse/7.4.0.0 (Windows RT; x64) JunosPulseVpn/1.0.0.206

The version (1.0.0.206) increments with each new version.

Pulse Secure client for Windows Phone user agent string:

- Junos-Pulse/7.4.0.0 (Windows Phone; ARM) JunosPulseVpn/1.0.1.6

Pulse Secure client for Desktop user agent string:

- Pulse-Secure/8.1.1.51832 (Windows X) Pulse/5.1.1.51832

Related Documentation

- [Pulse Secure Client for Apple iOS Overview](#)
- [Pulse Secure Client for Android Overview](#)
- [Pulse Secure Client for Windows Phone Overview](#)
- [Microsoft Windows In-Box Pulse Secure Client Overview](#)

CHAPTER 10 Pulse Secure Client for Apple iOS

- [Pulse Secure Client for Apple iOS Overview](#)
- [Configuring a Role and Realm for Pulse Secure Client for Apple iOS](#)
- [Allowing Pulse Secure Client for iOS Users to Save Webmail Password](#)
- [Host Checker for Pulse iOS Clients](#)
- [Configuring Host Checker for Pulse Secure iOS Clients](#)
- [Implementing Host Checker Policies for Pulse Secure Client for iOS Devices](#)
- [Installing the Pulse Secure Client for Apple iOS App](#)
- [Using iPhone Configuration Utility Profiles for Pulse Secure Client for iOS](#)
- [Collecting Log Files from Pulse Secure Client for iOS](#)
- [Launching the Pulse Secure Client for iOS App with a Command](#)
- [Pulse Secure Client for iOS Error Message Reference](#)
- [Configuring Secure Mail](#)

Pulse Secure Client for Apple iOS Overview

Pulse Secure client provides Layer 3 VPN connectivity based on SSL encryption and authentication between an Apple iOS device (iPhone, iPad, iPod Touch) and Pulse Connect Secure. Pulse Secure client enables secure connectivity to corporate applications and data based on identity, realm, and role. Pulse is designed to provide battery-friendly connectivity by automatically disconnecting from the VPN when the device is inactive while on Wi-Fi, automatically reestablishing VPN connectivity when the device reactivated, and maintaining connectivity when roaming from network to network. Pulse Secure client is available for download from the Apple App Store.



Note: Mobile client features are updated frequently and each mobile client has a release number that is independent from the other clients and from the Pulse for Windows and the Pulse for Mac clients. We recommend that users upgrade their mobile client to the latest release to ensure that all features described in this guide are supported on the devices.

The *Pulse Secure Mobile Supported Platforms Guide*, which is available at <https://www.pulsesecure.net/techpubs/pulse-client/pulse-secure-client-mobile> lists the mobile device OS versions supported by Pulse and the security features supported on each mobile device OS.

The Pulse Secure VPN app supports the following features:

- Full Layer 3 tunneling of packets
- UDP/ESP and NCP/SSL modes
- Authentication by all authentication options available on the Pulse Connect Secure server
- Certificate authentication followed by any other form of authentication
- Multi-factor authentication (cascading two different types of authentication)
- Host Checker



Note: A Host Checker policy that is configured for a VPN tunnel is not triggered if the VPN is launched automatically by VPN on Demand on iOS. If the VPN session is started through the Pulse client, Host Checker policy is correctly applied.

- Split tunneling modes:
 - Split tunneling disabled with access to local subnet
 - Split tunneling enabled
- Apple VPN on Demand

A VPN on Demand configuration enables an iOS device to automatically initiate a VPN connection when any application running on the phone initiates a connection to a host in a predefined set of hosts. A VPN on Demand connection uses client certificate-based authentication so the user does not have to provide credentials every time a VPN connection is initiated.



Note: When you configure VPN on Demand, you must create an exception for your Pulse Connect Secure server hostname. For example, if the hostname is `sslvpn.example.com` and you want Pulse clients to automatically establish the VPN whenever requests are made for hosts in the `example.com` domain, the VPN on Demand configuration should contain the following rules:

- If domain name = `sslvpn.example.com`, then never initiate VPN connection.
- If domain name = `example.com`, then always initiate VPN connection.

There are different methods for creating VPN on Demand connections:

- Create and manage VPN on Demand configurations from within the Pulse Secure client for iOS client.
 - Use the iPhone Configuration Utility. For complete information about how to create a VPN on Demand configuration using the iPhone Configuration utility, see the *iPhone OS Enterprise Deployment Guide*, which is available at www.apple.com.
- Secure Mail

An ActiveSync proxy on the Pulse Connect Secure server provides secure email services to Pulse iOS clients. The secure mail service encrypts email body content and attachments, supports email forwarding, and allows the Pulse administrator to quarantine the iOS device. Users view encrypted email body with any native iOS email client. Encrypted attachments are displayed by the secure viewer in the Pulse iOS client.

Before You Begin

Before you configure support for Apple iOS devices with Pulse Connect Secure, keep in mind the following client software behaviors:

- With Wi-Fi connectivity, Pulse reconnects the VPN tunnel automatically when the user wakes up the device. With 3G connectivity, the VPN reconnects when the user generates network traffic using an application like Safari or Mail.
- Establishing the VPN tunnel through a proxy is supported (regardless of the split tunnel mode), except for proxies that require authentication credentials.
- A Proxy Automatic Configuration (PAC) script takes effect only when split tunneling mode is disabled with access to local subnet. The PAC script does not work when the role's split tunnel mode is *Enable split tunneling*.
- Static host mapping is not created for the Pulse server/proxy hostname.
- DNS considerations:
 - When split tunneling is set to *Split tunneling disabled with access to local subnet*, Pulse uses the DNS servers that are configured on the Pulse server.
 - When split tunneling is set to *Split tunneling enabled*, DNS servers that are configured on the Pulse server are used only for hostnames within the Pulse Connect Secure domains.
- Session scripts are not supported.
- Web-based installation from a Pulse Secure gateway is not supported.
- Session timeout reminders are not supported.
- When you use client certificate authentication, and the user is enabled to select from among assigned roles, the user is prompted to enter the role name instead of being presented with a list of roles.

- To ensure that users see consistent bookmarks in the Pulse client UI no matter which server they are connected to, you can configure and enable user record synchronization, a feature of the Pulse Connect Secure platform.

Related Documentation

- [Configuring a Role and Realm for Pulse Secure Client for Apple iOS](#)
- [Host Checker for Pulse iOS Clients](#)
- [Configuring Host Checker for Pulse Secure iOS Clients](#)

Configuring a Role and Realm for Pulse Secure Client for Apple iOS

To enable SSL/VPN access from an Apple iOS device to Pulse Connect Secure, the device user must download, install, and configure the Pulse Secure app, and the Pulse administrator must configure specific realm and role settings on Pulse Connect Secure.



Note: Pulse Secure, LLC has created an App VPN SDK that enables app developers to integrate Pulse VPN connectivity within individual apps. Using an App VPN tunnel means that a mobile device user does not need to activate a Pulse connection and direct all traffic from the mobile device to the Pulse gateway. Only traffic from the app goes through the tunnel. Configuration on the Pulse server for App VPN is no different than the configuration for Pulse mobile client connections. For more information about Pulse VPN SDK, see your Pulse Secure representative.

To configure Pulse Connect Secure for Apple iOS device access:

1. Log in to the Pulse server admin console.
2. Select **User Roles > New User Role**.
3. On the New Role page, specify a name for the role and, optionally, a description. Make note of the name because later in this procedure, you create a realm and map realm users to this role.
4. In the Access Features section of the New Role page, select the **VPN Tunneling** check box.
5. Click **Save Changes** to create the role and to display the role configuration tabs.

Specifying Host Checker policies is part of the role configuration. However, you must first create the policy you want to assign to the role, so that procedure is covered later.

6. Select **Web > Bookmarks** and then click **New Bookmark**.
7. Specify a name and description for the bookmark.

You must create bookmarks to enable the buttons that appear in the Pulse for iOS user interface. Typically, you create a bookmark for your company intranet and for Web e-mail.



Note: You must create an e-mail bookmark to enable the e-mail button within the Pulse interface on the iOS device, and that e-mail bookmark must be named Mobile Webmail.

8. In the URL box, specify the Web address for access to your organization's e-mail.

Figure 115: Creating the E-mail Bookmark for the Pulse Client

Roles > Multi Role - Role 2 >
Web Bookmark

Name:

Description:

▼ **Bookmark To**

*URL: Example: http://www.domain.com/
We recommend that you use the fully qualified domain name when entering the bookmark URL.

▼ **Display Options**

☐ Open the bookmark in a new window

☐ Do not display the Web browser's URL address bar

☐ Do not display the Web browser's menu and toolbar

* Indicates required field



Note: Alternatively, you can use Web resource policies to define the bookmarks.

- On the VPN Tunneling tab, set the Split Tunneling options by selecting the following options:

Split Tunneling

- Enable**—Split tunneling resource policies specify the traffic that passes through the VPN tunnel.
- Disable**—All network traffic goes through the VPN tunnel.

Route Precedence

- Tunnel Routes**—The route table associated with the Pulse virtual adapter take precedence. Pulse overwrites the physical interface routes if there is conflict between the Pulse virtual adapter and the physical adapters. Pulse restores the original routes when the connection is ended.
- Tunnel Routes with local subnet access (Pulse on Windows and macOS only)**
- Endpoint Routes**—The route table associated with the endpoint's physical adapter take precedence.

- To change default session time-outs, select **General > Session Options**.
- In the Session lifetime section, specify **Max. Session Length** in minutes. The remaining session time appears on the Pulse interface of the mobile device client in the format days hours:minutes:seconds. The other session settings are not applied to mobile clients.
- Click **Save Changes**.
- Select **Users > Resource Policies > VPN Tunneling > Connection Profiles**.

A resource policy is a system rule that specifies resources and actions for a particular access feature. .

- Click **New Profile**.
- Specify a name and description for the connection profile.

When you define the connection profile, note the following:

- IP Address Assignment options**—When Pulse Connect Secure receives a client request to start a session, it assigns an IP address to the client based on the IP address policies you define.

- *Connection Settings*—ESP is the default transport. The Pulse for iOS VPN client supports both ESP and SSL.
 - *DNS Settings*—Searching IVE DNS first with split tunneling enabled is not supported. With split tunneling enabled, Pulse Secure client uses the IVE DNS for queries for hosts in the IVE DNS search domains only. All other queries go to the client's DNS servers.
 - *Proxy Server Settings*—The Pulse for iOS client software supports all of the proxy server settings except **Preserve client-side proxy settings**. That option is specific to the Windows client only. Automatically modifying the client proxy configuration when split tunneling is enabled is not supported.
16. In the Roles area, select **Policy applies to SELECTED roles**. Then add the role you created for iOS devices to the Selected roles list.
 17. Click **Save Changes**.
 18. Select **Users > User Realms > New User Realm**.
 19. Specify a name and description. Then click **Save Changes** to create the realm and to display the realm option tabs.
 20. In the Servers section, specify the authentication settings.

Authentication server configuration is described in *Authentication Servers*.

21. On the General tab for the realm, select the **Session Migration and Sharing** check box.
22. On the Role Mapping tab for the realm, create a new rule that maps all users to the iOS device role you created earlier in this procedure.

Related Documentation

- [Host Checker for Pulse iOS Clients](#)
- [Configuring Host Checker for Pulse Secure iOS Clients](#)
- [Allowing Pulse Secure Client for iOS Users to Save Webmail Password](#)
- [Pulse Connect Secure Split Tunneling Overview](#)

Allowing Pulse Secure Client for iOS Users to Save Webmail Password

A Web bookmark on the role for iOS users allows users to access e-mail through a Web link. You can allow users of the Pulse for iOS app to save their e-mail password when they login to the e-mail system. After you have created a Mobile Webmail bookmark for the role used by iOS users, enable password the option for user to save their e-mail password by doing the following.

1. Open the role you created for iOS users.
2. Click **General > Session Options**.
3. In the section labeled **Persistent Password Caching**, select **Enabled**.
4. Click **Save Changes**.

Related Documentation

- [Configuring a Role and Realm for Pulse Secure Client for Apple iOS](#)
- [Host Checker for Pulse iOS Clients](#)

Host Checker for Pulse iOS Clients

Host Checker is a component of Pulse Secure client that reports the integrity of iOS endpoints that are attempting to connect to Pulse Connect Secure. Host Checker runs as a Trusted Network Connect (TNC) client on the endpoint. The client evaluates the endpoint according to predefined criteria and reports to the Trusted Network Connect server, which is a part of Pulse Connect Secure. If the endpoint is not in compliance with the Host Checker policies, then the user might not get access to the network or might get limited access to the network depending upon the enforcement policies configured by the administrator.



Note: A Host Checker policy that is configured for a VPN tunnel is not triggered if the VPN is launched automatically by VPN on Demand.

For iOS clients, Host Checker can evaluate client compliance based on the following predefined criteria:

- **OS Checks**—You can specify the iOS version or minimal version that must be installed on the device.
- **Jail Breaking Detection**—Jail breaking is a process that allows Apple iPhone, iPad and iPod Touch users to gain root access to the iOS operating system and bypass usage and access limitations imposed by Apple. With a jail broken device, an iOS user can install applications that are not available through the Apple App Store. Jail broken devices expose the device to a greater risk of running malicious applications.

Related Documentation

- [Configuring Host Checker for Pulse Secure iOS Clients](#)
- [Implementing Host Checker Policies for Pulse Secure Client for iOS Devices](#)

Configuring Host Checker for Pulse Secure iOS Clients

Host Checker policies can be part of a larger Host Checker configuration that applies to many different types of clients or to iOS devices only. However, you might find it easiest to create a separate Host Checker policy specifically for iOS devices.



Note: A Host Checker policy that is configured for a VPN tunnel is not triggered if the VPN is launched automatically by VPN on Demand on iOS.

To create a Host Checker policy for iOS devices:

1. From the admin console, select **Authentication > Endpoint Security > Host Checker**.
2. In the Policies section, click **New** to open a New Host Checker Policy page.
3. Specify a name for the new policy and then click **Continue** to open the Host Checker Policy page.
The name appears in lists when you implement the policy so be sure to use a descriptive name, such as iOS HC Policy.
4. Click the Mobile tab, and then click the iOS tab.
5. In the Rule Settings section, click **Select Rule Type** and select one of the following options and then click **Add**:
 - **OS Checks**—To specify the iOS version that must be installed on the device:
 1. Specify a descriptive name for this rule. For example, Must-Be-iOS-4.1-or-higher. Rule names cannot include spaces.
 2. Specify the criteria. For example, to enforce iOS 4.1 or higher, create two conditions: Equal to 4.1 and Above 4.1.
 3. Click **Save Changes**.
 - **Jail Breaking Detection**—Jail breaking is a process that allows Apple iPhone, iPad and iPod Touch users to gain root access to the iOS operating system. and bypass usage and access limitations imposed by Apple. With a jail broken device, an iOS user can install applications that are not available through the Apple App Store. Jail broken devices possess a greater risk of running malicious applications.
 1. Specify a descriptive name for this rule. For example, No-iOS-Jailbreak.
 2. The **Don't allow Jail Broken devices** check box is enabled by default.
 3. Click **Save Changes**.
6. After you have configured all of your rules, specify how you want to enforce them by choosing one of the following options:
 - All of the rules
 - Any of the rules
 - Custom

For Custom requirements, you can specify a custom expression using Boolean operators AND and OR and also group and nest conditions using parenthesis.

7. Specify remediation options:

- **Enable custom instructions**—If you enable this check box, a text box appears and allows you to type information that appears on the user's device if Host Checker discovers an issue. For example, if you enabled the MSS rule that terminates the VPN session of Host Checker discovers a virus, you can instruct the user to run a virus scan to clear the issue before trying to connect.
- **Send reason strings**—Select this option to display a message to users (called a reason string) that explains why the client machine does not meet the Host Checker policy requirements. For example, if the jailbreak detection policy fails, Pulse displays **A jailbroken device is not allowed to access the network. Please contact your network administrator.**

8. When you are finished, click **Save Changes**.

Related Documentation

- [Host Checker for Pulse iOS Clients](#)
- [Implementing Host Checker Policies for Pulse Secure Client for iOS Devices](#)
- [Pulse Secure Client for Mobile Devices Overview](#)

Implementing Host Checker Policies for Pulse Secure Client for iOS Devices

After you create one or more Host Checker policies for iOS devices, you must implement them. Pulse Connect Secure can use Host Checker policies at the realm or the role level.

Realm Authentication—You can configure a realm authentication policy to download and run Host Checker with a particular Host Checker policy. If the iOS device does not meet the Host Checker requirements, then Pulse Connect Secure can deny access. You can provide remediation information in the Host Checker policy to describe the requirement and help users take steps to solve the issue.

To enable a Host Checker policy for a realm:

1. From the admin console, select **Users > User Realms > SelectRealm > Authentication Policy > Host Checker**. The Host Checker page displays all of the available Host Checker policies.
2. Select the check box next to each policy you want to include. Select one or both of the following check boxes next to the policy:
 - **Evaluate Policies**—Evaluates without enforcing the policy on the iOS device and allows access.
 - **Require and Enforce**—Requires that the iOS device be in compliance with the Host Checker policy. Pulse Connect Secure downloads Host Checker to the iOS device after the user is authenticated and before the user is mapped to any roles in the system. Selecting this option automatically enables the Evaluate Policies option.
3. Optionally select **Allow access to realm if any ONE of the selected "Require and Enforce" policies is passed**. This check box is available if you selected more than one Host Checker policy. If you enable this check box, an iOS device is allowed access if it passes any of the Require and Enforce policies. The Cache Cleaner policy does not apply to iOS devices.
4. Click **Save Changes**.

Role—You can configure a role to download and run Host Checker with a particular Host Checker policy. If the iOS device does not meet the Host Checker requirements, then Connect Secure can deny access or assign the user to a remediation role that has limited access. You can provide remediation information in the Host Checker policy to help users take steps to solve the issue.

To enable a Host Checker policy for a role:

5. From the admin console, select **Users > User Roles > SelectRole > General > Restrictions > Host Checker**. The Host Checker page displays all of the available Host Checker policies.
6. Select **Allow users whose workstations meet the requirements specified by these Host Checker policies**.
7. In the Available Policies list, select the policies that you want to apply to select them, and then click Add to move them to the Selected Policies list. To select a policy click it. To select more than one policy, use Ctrl+click.

8. Optionally select **Allow access to realm if any ONE of the selected “Require and Enforce” policies is passed**. This check box is available if you selected more than one Host Checker policy. If you enable this check box, an iOS device is allowed access if it passes any of the Require and Enforce policies. The Cache Cleaner policy does not apply to iOS devices.
9. Click **Save Changes**.

Related Documentation

- [Host Checker for Pulse iOS Clients](#)
- [Configuring Host Checker for Pulse Secure iOS Clients](#)

Installing the Pulse Secure Client for Apple iOS App

Pulse Secure client is available in the iTunes App Store. After installing the Pulse app, a user can manually configure it.

1. On the iOS device, launch Pulse Secure client.
2. Tap the Configuration item on the main status page to display Pulse configurations.
3. Create a new configuration with the URL. The URL for the connection is the Pulse Connect Secure sign-in URL that was created and defined for mobile devices. Then configure the certificate settings as required.



Note: When iPhone users launch Pulse for the first time, they see a security warning and a prompt for enabling Pulse SSL VPN functionality. This security precaution helps deter the silent installation of malicious VPN software. If the user declines the Pulse software, the Pulse splash screen appears until the user presses the Home button on the device. If the user accepts the Pulse software, the security warning no longer appears when Pulse is started.



Note: For certificate authentication, the Pulse Connect Secure SSL certificate must be issued by a CA. It cannot be self-signed. If the CA is not one of the built-in trusted CAs on the iOS device, then the CA certificate must be imported into the iOS device. Also, the Pulse Connect Secure must be accessed using a hostname (not an IP address), and the hostname must match the Common Name of the Pulse Connect Secure SSL certificate.

Related Documentation

- [Pulse Secure Client for Apple iOS Overview](#)

Using iPhone Configuration Utility Profiles for Pulse Secure Client for iOS

One method of creating VPN configurations is to use a Configuration Profile to define Pulse configurations for the iOS device, and then distribute the configuration profiles by e-mail or by posting them on a Web page. When users open the e-mail attachment or download the profile using Safari on their iOS device, they are prompted to begin the installation process.

You use the iPhone Configuration Utility to create configuration profiles and specify Juniper SSL as the Connection Type for the VPN Payload. You can download the iPhone Configuration Utility (3.0 or later) from the Apple support Web. For details about the utility and how to create Configuration Profiles, see the *iPhone OS Enterprise Deployment Guide*, which is available at www.apple.com.

Related Documentation

- [Pulse Secure Client for Apple iOS Overview](#)
- [Pulse Secure Client for Mobile Devices Overview](#)

Collecting Log Files from Pulse Secure Client for iOS

The iOS device user can use the following procedure to e-mail the Pulse Secure log files:

1. On the iOS device, start the Pulse Secure app.
2. Tap **Status**.
3. Tap **Logs > Send Logs**.
4. Enter an e-mail address and tap **Send**.

Related Documentation

- [Pulse Secure Client for Apple iOS Overview](#)
- [Installing the Pulse Secure Client for Apple iOS App](#)

Launching the Pulse Secure Client for iOS App with a Command

The Pulse launcher for iOS is a command that is registered with iOS when the mobile device user installs the Pulse Secure client for iOS app. The Pulse launcher starts (or stops) Pulse and can establish a VPN connection using connection parameters specified in the command. The command can specify all login parameters. If the app generates or accesses an appropriate passcode, Pulse starts and establishes a VPN connection with no input from the user. You can use the Pulse launcher command in Web pages and external apps.

When a user taps a button that is tied to a Pulse launcher command, the command launches the Pulse app if it is not already running. If Pulse is not already installed on the iOS device, an error occurs. The next step depends on the current Pulse connection status and configuration. One of the following occurs:

- If Pulse does not already have an active connection to Pulse Connect Secure, it uses an existing configuration to establish the VPN connection.
- If Pulse does not already have an active connection, and it does not already have a configuration for the target Pulse server, Pulse opens the Add Configuration screen. The target URL is already defined and the user just needs to specify a name for the connection.
- If the Pulse app is already connected to a Pulse server, the Pulse app is brought to the foreground.

To employ the Pulse launcher in your Web pages or external applications, specify the link using the following format:

```
pulsesecure://<server-host>/<server-path> ?method={vpn} &action={start|stop} &DSID=<dsid-cookie>
&SMSESSION=<smsession-cookie> &username=<username> &password=<password> &realm=<realm> &role=<role>
```

Usage notes:

- If the DSID cookie is given in the URL, the app does not use the "username", "password", "realm", or "role" parameters because no login is required.
- The values for username, realm, and role are URI-escaped values. Special characters are replaced with their hexadecimal equivalents preceded by '%'.
- If the user has specified the username, realm, and role when creating the VPN configurations in the Pulse Secure app, those values are used to auto-fill the username, realm, and role for the login pages during a Web-based login. During login, if all fields are successfully auto-filled from fields in the VPN configuration or the pulsesecure:// launch URL, the login progresses without any user input. The username, realm, and role values need to already exist in the VPN configuration for them to be auto-filled during the login process. If the user manually specifies the username, realm, or role during login, the app will not add or update these values in the VPN configuration. The user needs to explicitly update the VPN configuration with these values.
- The Pulse app does not save the password in the Password field in VPN configurations. The Pulse app does not use values from Password fields in VPN configurations installed by the iPhone Configuration Utility. Pulse Secure client will only use passwords specified in pulsesecure:// URLs.

- If the user manually specifies the username, realm, or role during login, the app stores these values in the VPN configuration and they will be auto-filled the next time the user signs in. Passwords entered by the user are not saved in the VPN configuration.
- Realm and role fields in the VPN configuration format are supported in Apple iOS 4.2 and later. If the Pulse app is run on an iOS device running iOS 4.1, the realm and role fields will not be visible in the Pulse Secure client Add/Edit configuration view.

Examples:

If the calling application has already obtained a DSID cookie from Pulse Connect Secure, the app can use the following command to start the VPN:

```
pulsesecure://<server-host>/<server-path> ?method=vpn &action=start &DSID=<dsid-cookie> &SMSESSION=<smsession-cookie>
```

If the calling application does not already have a DSID, it can use the following command to start the VPN:

```
pulsesecure://<server-host>/<server-path> ?method=vpn&action=start &username=<username> &password=<password> &realm=<realm> &role=<role>
```

If the calling application wants to stop the VPN, it can use the following command:

```
pulsesecure://<server-host>/<server-path> ?method=vpn &action=stop
```

Related Documentation

- [Pulse Secure Client for Mobile Devices Overview](#)

Pulse Secure Client for iOS Error Message Reference

The following error message summary for Pulse Secure client for iOS describes possible issues and suggests resolution actions where possible.

Table 24: Pulse Secure Client for iOS Error Messages

Message	Possible Causes	Suggested Actions
Please provide values for all the fields	A required field was not provided.	Provide a value for all the required fields and then try the operation again. Contact your mobile security provider.
A configuration with the same name already exists. Please choose a different name.	Configuration names must be unique.	Choose a configuration name that is not in use by another configuration, and then try the operation again.
An internal error occurred while creating the configuration.	An undefined error occurred.	Verify all of the values you entered, and then try the operation again. If the error occurs again, contact the Pulse administrator.
Please contact your administrator.	Host Checker policy failed and the reason string is displayed for the failure.	Tap the Cancel button and then try again after performing the remediation actions.
Your device is running operating system version x.y.z.	The iOS version running on the device is not allowed to connect.	If prompted to continue, tap Continue to connect with limited connectivity or tap Cancel to cancel the connection and try again after upgrading iOS.
Your iOS device is jailbroken.	Jail broken iOS devices are not allowed to connect.	If prompted to continue, tap Continue to connect with limited connectivity or tap Cancel to cancel the connection.

Message	Possible Causes	Suggested Actions
Host Checker is not supported with this version of Pulse Secure. Please upgrade the Pulse Secure client or contact your administrator.	Unsupported Pulse Secure client - Pulse Secure Host Checker is supported on Pulse 3.2 and later.	Check for the update of Pulse Secure client on the App store and upgrade the Pulse Secure client to 3.2 or later, and then try again.
Session disconnected due to invalid certificate.	The Pulse client downloads session information from the Pulse server and the certificate received from the server does not match the stored session certificate.	Click the Close button on the Alert dialog to return to home screen. User can retry the connection.
Failed to connect to the server.	Sign-in process failed.	Check the network connection (for example, Wi-Fi, 3G, etc.), and then retry the connection.
Compliance Check couldn't be completed.	Host Checker compliance check couldn't be completed during sign-in process.	Try to connect again.

Related Documentation

- [Pulse Secure Client for Apple iOS Overview](#)
- [Installing the Pulse Secure Client for Apple iOS App](#)

Configuring Secure Mail

For iOS devices, you can configure Secure Mail to:

- Use ActiveSync to synchronize e-mails with a Microsoft Exchange server
- Encrypt the e-mail body and attachments
- Block access to e-mail for lost or stolen devices, and erase all e-mails on the device
- Support back-end redirection, where an Exchange ActiveSync server redirects the client to another Exchange server

For other devices, an authorization-only sign-in policy can be defined that supports all of the Secure Mail features, except encryption.

Secure Mail is enabled in the user role, and requires a resource profile to specify the Exchange server and encryption options. In addition, an S/MIME certificate must be imported to Connect Secure.

For iOS devices that use Secure Mail, the Pulse Secure client must be installed during onboarding. Onboarding downloads the ActiveSync profile to the device, along with any other profiles defined for enterprise onboarding (see *Configuring Enterprise Onboarding*). Profiles are signed by the Connect Secure device certificate, which must be trusted by the client device.

- [Enabling Secure Mail at the Role Level](#)
- [Defining the Secure Mail Resource Profile](#)
- [Obtaining an S/MIME Certificate](#)
- [Configuring an Authorization-Only Policy for ActiveSync](#)

Enabling Secure Mail at the Role Level

To use Secure Mail for iOS devices, you must enable it at the user role level and then create a resource profile that specifies the mail server and encryption settings.



Note: Do not enable both Secure Mail and Email Client options in the same role. To disable Secure Mail for a role, you must first block e-mail access for all devices that are currently onboarded with this role (see *Managing Onboarded and ActiveSync-Only Devices*).

To enable Secure Mail for a user role:

1. In the admin console, choose **Users > User Roles > RoleName > General > Overview**.
2. In the Enterprise Device Onboarding section, select the **Secure Mail** check box.
3. Click **Save Changes**.
4. Click **Options** next to the **Secure Mail** check box to view or change the resource profile for Secure Mail (see [“Defining the Secure Mail Resource Profile”](#)).

Defining the Secure Mail Resource Profile

To use Secure Mail for iOS devices, you must enable it at the role level and then create a resource profile that specifies the Exchange server and encryption settings. You must also obtain and import an S/MIME certificate (see [“Obtaining an S/MIME Certificate”](#)).

To define the Secure Mail resource profile:

1. In the admin console, choose **Users > Resource Profiles > Mobile**.
2. Specify the following information:

Setting	Description
Virtual Hostname	<p>Enter a hostname alias for the Exchange server, and update your DNS server to map the alias to the IP address of the Connect Secure. The name must be unique among all virtual hostnames.</p> <p>For example, if the virtual hostname is <i>email.com</i>, and the backend URL is <i>https://mail.pulsesecure.net:8080</i>, a client request to <i>https://email.com/test1</i> via Connect Secure is converted to <i>https://mail.pulsesecure.net:8080/test1</i>. The response to the converted request is sent to the client web browser.</p>
Exchange Server	Enter the URL and port number of the Microsoft Exchange server, such as <i>https://mail.pulsesecure.net:379</i> . If the port number is omitted, it defaults to 80.
Description	Description of the Exchange server (optional).
Username	<p>Select one of the following to specify the e-mail account format used by the Exchange server:</p> <ul style="list-style-type: none"> • None – Inserts the <USER> variable for the user’s login name for Connect Secure (the default). • Exchange 2007/2010/2013 – Inserts the <NTDOMAIN>\<USER> variables to include the user’s domain before the login name. • Office 365 – Inserts <USER>@domain.com, and you can enter the appropriate domain, such as <USER>@pulsesecure.net.

Setting	Description
Secure Mail Options	<p>Select one or more of the following encryption options:</p> <ul style="list-style-type: none"> • Encrypt Body – Encrypts the body of the e-mail using an S/MIME certificate. The encrypted e-mail body can be viewed by any native e-mail client. <p>NOTE: Graphics embedded in the encrypted e-mail body are displayed twice on iOS devices.</p> <ul style="list-style-type: none"> • Encrypt Attachments – Encrypts the e-mail attachments using a key generated by Connect Secure. Encrypted attachments, which must be opened with the Pulse Secure client, are identified by a pulsesecure file extension, such as report.pdf.pulsesecure. The encrypted file types are listed in the File Extensions text box, separated by semicolons. You can add or delete file extensions from the list. <p>NOTE: If you add .gif, .jpeg, .jpg, .png or .htm to the list of encrypted file types, graphics embedded in the e-mail body are not displayed correctly on iOS devices.</p> <ul style="list-style-type: none"> • Allow Outbound E-Mail Attachments – Decrypts attachments before forwarding an e-mail to an external account. If this option is not selected, e-mails are forwarded without attachments and include a note indicating that attachments were removed. <p>If you change the encryption settings, onboarded devices must be re-onboarded to obtain the new settings (see <i>Managing Onboarded and ActiveSync-Only Devices</i>).</p>

3. Click **Save Changes**.

Obtaining an S/MIME Certificate

If you enable Secure Mail, an S/MIME is required for each client device. You can generate an S/MIME certificate for each device or use a global certificate for all devices by requesting an S/MIME certificate from a Certificate Authority (CA) and importing the certificate and private key to Connect Secure.

To generate or import an S/MIME certificate:

1. In the admin console, choose **Users > Resource Profiles > Mobile > S/MIME Certificate**.
2. Specify one of the following options:

Setting	Description
Generate per User Certificate	Select this option to use the SCEP server and a CSR template to generate a certificate for each client. Select a CSR template from the Use Certificate Template list. To create a CSR template, see <i>Configuring Enterprise Onboarding</i> .
Upload and Use Single Global Certificate	<p>Select this option to use the same certificate for all client devices. Click Import Certificate & Key, click Browse in one of the following forms to locate the certificate file, enter the password key if the file is encrypted, and then click Import.</p> <ul style="list-style-type: none"> • If certificate file includes private key—When the certificate and key are contained in one file. • If certificate and private key are separate files—When the certificate and key are in separate files. • Import via System Configuration file—When the certificate and key are contained in a system configuration file that has been exported from Connect Secure.



Note: The Import Certificate & Key button is disabled on FIPS hardware platforms because importing private keys is not allowed. On a FIPS hardware platform, you must create a CSR and then import a signed certificate from the CA.

Configuring an Authorization-Only Policy for ActiveSync

An authorization-only policy can be defined to allow almost any device to use ActiveSync to synchronize e-mails with a Microsoft Exchange server. Encryption is not supported, but if the option to allow only ActiveSync traffic is enabled, e-mail access can be blocked for devices that are lost or stolen, and back-end redirection is supported (the Exchange ActiveSync server can redirect the client to another Exchange server).

To configure an authorization-only access policy for ActiveSync:

1. In the admin console, choose **Authentication > Signing In > Sign-in Policies**.
2. To create a new authorization-only access policy, click **New URL** and select **Authorization Only Access** for the user type. Or, to edit an existing policy, click **Secure Mail** next to a URL in the Virtual Hostname column.



Note: If a resource profile for Secure Mail is defined, the virtual hostname is displayed with a link to the resource profile. The virtual hostname for the authorization-only policy must be different from the Secure Mail virtual hostname.

3. Specify the following information:

Setting	Description
Virtual Hostname	<p>Enter a hostname alias for the Exchange server, such as <i>email.com</i>, and update your DNS server to map the alias to the IP address of Connect Secure. The name must be unique among all virtual hostnames. Do not include the http(s) protocol.</p> <p>For example, if the virtual hostname is <i>email.com</i>, and the backend URL is <i>https://mail.pulsesecure.net:8080</i>, a client request to <i>https://email.com/test1</i> via Connect Secure is converted to <i>https://mail.pulsesecure.net:8080/test1</i>. The response to the converted request is sent to the client web browser.</p>
Backend URL	Enter the URL and port number of the Microsoft Exchange server, such as <i>https://mail.pulsesecure.net:8080</i> . If the port is omitted, it defaults to 80.
Description	Description of the Exchange server (optional).
Authorization Server	Select the authorization server name or No Authorization . If you select a server, ensure that the front-end server provides the SMSESSION cookie otherwise you will receive an error.
Role Option	<p>Select a user role. The role must have a Web Access policy, and only the following role options apply:</p> <ul style="list-style-type: none"> • Allow browsing untrusted SSL web sites (Users > User Roles > <i>RoleName</i> > Web > Options > View advanced options) • HTTP Connection Timeout (Users > User Roles > <i>RoleName</i> > Web > Options > View advanced options) • Source IP restrictions (Users > User Roles > <i>RoleName</i> > General > Restrictions) • Browser restrictions (Users > User Roles > <i>RoleName</i> > General > Restrictions)

Setting	Description
Protocol Option	<p>Select the Allow ActiveSync Traffic only option to verify the request is consistent with the ActiveSync protocol, and to include the device on the Device Management page, which allows e-mail access on the device to be blocked if needed (see <i>Managing Onboarded and ActiveSync-Only Devices</i>).</p> <p>If validation fails, a message is created in the user's event log. If you do not enable this option, both ActiveSync and non-ActiveSync requests are processed, but the device is not shown on the Device Management page.</p>

4. Click **Save Changes** to save your edits.

The System Status Overview page displays the number of current ActiveSync and authorization-only connections, and a histogram of the active concurrent connections (Authorization Only Access Active Connections plot in the Concurrent SSL Connections graph).

CHAPTER 11 Pulse Secure Client for Google Android

- [Pulse Secure Client for Android Overview](#)
- [Configuring a Role and Realm for Pulse for Android](#)
- [Application Acceleration on Pulse Secure Client Mobile VPN Connections](#)
- [Configuring Application Acceleration for Pulse Mobile VPN Connections](#)
- [Allowing Pulse Secure Client for Android Users to Save Webmail Password](#)
- [Host Checker for Pulse Android Clients](#)
- [Configuring Host Checker for Pulse Secure Client Android Clients](#)
- [Implementing Host Checker Policies for Pulse Secure Client for Android Clients](#)
- [Pulse Secure Client for Android Error Message Reference](#)
- [Launching the Pulse Secure Client for Android App Using a Command](#)
- [Launching the Pulse Secure Client for Android App Through a Browser Link](#)

Pulse Secure Client for Android Overview

Pulse Secure client can create an authenticated SSL session between a device running Google Android and Pulse Connect Secure. Pulse Secure client enables secure connectivity to Web-based applications and data based on identity, realm, and role. Pulse Secure Client is available for download from the Android Market. The *Pulse Secure Mobile Supported Platforms Guide*, which is available at <https://www.pulsesecure.net/techpubs/pulse-client/pulse-secure-client-mobile> lists the mobile device OS versions supported by Pulse and the security features supported on each mobile device OS.



Note: Pulse Secure has created an App VPN SDK that enables app developers to integrate Pulse VPN connectivity within individual apps. Using an App VPN tunnel means that a mobile device user does not need to activate a Pulse connection and direct all traffic from the mobile device to the Pulse gateway. Only traffic from the app goes through the tunnel. Configuration on the Pulse server for App VPN is no different than the configuration for Pulse mobile client connections. For more information about Pulse VPN SDK, see your Pulse Secure, LLC representative.



Note: The Google Android OS has limitations in its support for certificate-based authentication. For successful certificate authentication, the user certificate and the private key must be separate files. If necessary, you can separate the private key from the certificate by using openssl commands before you install the certificate and the key on the Android device. The Pulse Secure Knowledgebase includes an article, KB19692, that describes in detail how to create a certificate and key that enables successful certificate authentication for Pulse Secure client on Android.



Note: To ensure that users see consistent bookmarks in the Pulse client UI no matter which server they are connected to, you should configure and enable user record synchronization, a feature of the Pulse Connect Secure platform.

Configuring a Role and Realm for Pulse for Android

To enable access from an Android device to Pulse Connect Secure the Pulse administrator must configure specific realm and role settings on the Pulse server.

To configure Pulse Connect Secure for Android device access:

1. Log in to the Pulse server admin console.
2. Select **User Roles > New User Role**.
3. On the New Role page, specify a name for the role and, optionally, a description. Make note of the name because later in this procedure, you create a realm and map realm users to this role.
4. In the Access Features section, select **Web** and **VPN Tunneling**.
5. Click **Save Changes** to create the role and to display the role configuration tabs.
6. Select **Web > Bookmarks** and then click **New Bookmark**.

You must create bookmarks to enable the buttons that appear in the Pulse for Android user interface. Typically, you create a bookmark for your company intranet and for Web e-mail. You must create an e-mail bookmark to enable the e-mail button within the Pulse interface on the Android, and that e-mail bookmark must be named **Mobile Webmail**.

Figure 116: Creating the E-mail Bookmark for the Pulse Client

The screenshot shows the 'Web Bookmark' configuration page in the Pulse Secure admin console. The breadcrumb trail at the top reads 'Roles > Multi Role - Role 2 > Web Bookmark'. The 'Name' field is labeled 'Name:' and contains the text 'Test'. The 'Description' field is labeled 'Description:' and is empty. Below these fields is a section titled 'Bookmark To' with a sub-label '*URL:' and a text input field containing 'http://exchange3.iowa'. To the right of the URL field is a note: 'Example: http://www.domain.com/ We recommend that you use the fully qualified domain name when entering the bookmark URL.' Below the URL field is a section titled 'Display Options' with three checkboxes: 'Open the bookmark in a new window', 'Do not display the Web browser's URL address bar', and 'Do not display the Web browser's menu and toolbar'. At the bottom of the form are two buttons: 'Save Changes' and 'Save as Copy'. A small asterisk at the bottom left indicates that fields with an asterisk are required.



Note: Alternatively, you can use Web resource policies to define the bookmarks.

7. On the VPN Tunneling tab, set the Split Tunneling options by selecting the following options:

Split Tunneling

- **Enable**—Split tunneling resource policies specify the traffic that passes through the VPN tunnel.
- **Disable**—All network traffic goes through the VPN tunnel.

Route Precedence

- **Tunnel Routes**—The route table associated with the Pulse virtual adapter take precedence. Pulse overwrites the physical interface routes if there is conflict between the Pulse virtual adapter and the physical adapters. Pulse restores the original routes when the connection is ended.
 - **Tunnel Routes with local subnet access (Pulse on Windows and macOS only)**
 - **Endpoint Routes**—The route table associated with the endpoint's physical adapter take precedence.
8. To change default session time-outs, select **General > Session Options**.
 9. In the Session lifetime section, specify **Max. Session Length** in minutes. The remaining session time appears on the Pulse interface of the mobile device client in the format days hours:minutes:seconds. The other session settings are not applied to mobile clients.
 10. Select **Users > User Realms > New User Realm**.
 11. Specify a name and, optionally, a description and then click **Save Changes** to create the realm and to display the realm option tabs.
 12. On the Role Mapping tab for the realm, create a new rule that maps all users to the Android role you created earlier in this procedure.

Related Documentation

- [Host Checker for Pulse Android Clients](#)
- [Allowing Pulse Secure Client for Android Users to Save Webmail Password](#)
- [Pulse Connect Secure Split Tunneling Overview](#)

Application Acceleration on Pulse Secure Client Mobile VPN Connections

Pulse Secure client for Android and Pulse Secure client for Apple iOS support application acceleration through Riverbed® Steelhead® mobile technology. Steelhead mobile technology performs WAN optimization to improve performance of web browsing, email, and other applications. When there is a Steelhead Mobile Controller (SMC) in the network path with the Pulse Secure Access server, Pulse mobile clients can use application acceleration services over VPN connections.

Configuration is simple. If you have the Steelhead Mobile Controller installed and configured in your network, you need only enable Steelhead optimization and specify the SMC IP address as one of the settings for the role you assign to VPN users. When the client communicates with the SMC, the SMC downloads the required configuration.

Pulse mobile client users can also manually enter the SMC IP address for accelerated communications for connections that do not connect to Pulse Connect Secure. This “tunnel light” configuration is described in the Pulse mobile client user guides.

The Pulse mobile client user can view and change some acceleration settings from the Pulse client interface:

- SMC Controller information
- Health Status of the connection
- Controller status
- Policy
- Total Data Reduction
- Optimization Statistics
- Acceleration Version

Note:



- Data acceleration over UDP is not supported.
- Data acceleration for FTP is not supported. FTP might not work with optimization enabled.
- Mobile client acceleration is available for Pulse R5.0 and later for Apple iOS and Pulse R5.0 and later for Pulse for Google Android. For information and on mobile OS versions supported, see the [Supported Platforms Guide for Pulse for mobile devices](#)

Related Documentation

- [Configuring a Role and Realm for Pulse Secure Client for Apple iOS](#)
- [Configuring a Role and Realm for Pulse for Android](#)
- [Configuring Application Acceleration for Pulse Mobile VPN Connections](#)

Configuring Application Acceleration for Pulse Mobile VPN Connections

Pulse Secure client for Android and Pulse Secure client for Apple iOS support application acceleration through Riverbed® Steelhead® mobile technology. Steelhead mobile technology performs WAN optimization to improve performance of web browsing, email, and other applications. You enable Steelhead optimization and specify the SMC IP address as one of the settings for the role you assign to VPN users.

Before you begin:

- Create and configure one or more roles for Pulse mobile VPN users. See [“Configuring a Role and Realm for Pulse Secure Client for Apple iOS”](#) or [“Configuring a Role and Realm for Pulse for Android”](#).

To enable application acceleration for mobile device VPN connections:

1. Log in to the Pulse server admin console.
2. Click **Users > User Roles** and select the role to update.
3. Click **VPN Tunneling > Options**, and then scroll down to the section titled **Options for Steelhead Controller**.
4. Select **Enable Steelhead Optimization**.
5. Specify the **Server Location** as an IP address or a fully qualified hostname.
6. Click **Save Changes**.

Related Documentation

- [Application Acceleration on Pulse Secure Client Mobile VPN Connections](#)

Allowing Pulse Secure Client for Android Users to Save Webmail Password

A Web bookmark on the role for Android users allows users to access e-mail through a Web link. You can allow users of the Pulse for Android app to save their e-mail password when they login to the e-mail system. After you have created a Mobile Webmail bookmark for the role used by Android users, enable password the option for user to save their e-mail password by doing the following.

1. Open the role you created for Android users.
2. Click **General > Session Options**.
3. In the section labeled **Persistent Password Caching**, select **Enabled**.
4. Click **Save Changes**.

Related Documentation

- [Configuring a Role and Realm for Pulse for Android](#)
- [Host Checker for Pulse Android Clients](#)

Host Checker for Pulse Android Clients

Host Checker is a component of Pulse Secure Client that reports the integrity of Android endpoints that are attempting to connect to Pulse Connect Secure. Host Checker runs as a Trusted Network Connect (TNC) client on the endpoint. The client evaluates the endpoint according to predefined criteria and reports to the Trusted Network Connect server, which is a part of Pulse Connect Secure. If the endpoint is not in compliance with the Host Checker policies, then the user might not get access to the network or might get limited access to the network depending upon the enforcement policies configured by the administrator.

For Android clients, Host Checker can evaluate client compliance based on the following predefined criteria:

- **OS Checks**—You can specify the iOS version or minimal version that must be installed on the device.
- **Root Detection**—Rooting is a process that allows Android users to gain root access to the operating system and bypass usage and access limitations imposed by device manufacturers and carriers. With a rooted device, a user can install applications that are not available and have not been certified by the device manufacturer or by the app store process. Rooted devices expose the device to a greater risk of running malicious applications. Host Checker can detect rooted devices and then allows or deny network access based on the Host Checker enforcement policy.

Related Documentation

- [Configuring Host Checker for Pulse Secure Client Android Clients](#)
- [Implementing Host Checker Policies for Pulse Secure Client for Android Clients](#)
- [Pulse Secure Client for Mobile Devices Overview](#)

Configuring Host Checker for Pulse Secure Client Android Clients

Host Checker policies can be part of a larger Host Checker configuration that applies to many different types of clients or to Android devices only.

To create a Host Checker policy for Android devices:

1. From the admin console, select **Authentication > Endpoint Security > Host Checker**.
2. In the Policies section, click **New** to open a New Host Checker Policy page.
3. Specify a name for the new policy and then click **Continue** to open the Host Checker Policy page.

The name appears in lists when you implement the policy so be sure to use a descriptive name, such as Android HC Policy.

4. Click the Mobile tab, and then click the Android tab.
5. In the Rule Settings section, click **Select Rule Type** and select one of the following options and then click **Add**:
 - **OS Checks**—To specify the iOS version that must be installed on the device:
 1. Specify a descriptive name for this rule. For example, Must-Be-Android-2-2-or-higher. Rule names cannot include spaces.
 2. Specify the criteria. For example, to enforce Android 2.2 or higher, create two conditions: Equal to 2.2 and Above 2.2.

Host Checker supports Android versions 1.6 through 3.1.

3. Click **Save Changes**.
- **Rooting Detection**—Rooting is a process that allows Android users to gain root access to the operating system and bypass usage and access limitations imposed by manufacturers and service providers. With a rooted device, an Android user can install applications that have not been certified through the app store process. Rooted devices possess a greater risk of running malicious applications.
 1. Specify a descriptive name for this rule. For example, No-Android-root.
 2. The **Don't allow rooted devices** check box is enabled by default.
 3. Click **Save Changes**.
6. After you have configured all of your rules, specify how you want to enforce them by choosing one of the following options:
 - All of the rules
 - Any of the rules
 - Custom

For Custom requirements, you can specify a custom expression using Boolean operators AND and OR and group and nest conditions using parenthesis.

Specify remediation options:

- **Enable custom instructions**—If you enable this check box, a text box appears and allows you to type information that appears on the user's device if Host Checker discovers an issue. For example, if you enabled the MSS rule that terminates the VPN session of Host Checker discovers a virus, you can instruct the user to run a virus scan to clear the issue before trying to connect.
 - **Send reason strings**—Select this option to display a message to users (called a reason string) that is returned by Host Checker or integrity measurement verifier (IMV) and explains why the client machine does not meet the Host Checker policy requirements. For example, if the rooting detection policy fails, Pulse displays, "A rooted device is not allowed to access the network. Please contact your network administrator."
7. When you are finished, click **Save Changes**.

Related Documentation

- [Host Checker for Pulse Android Clients](#)
- [Implementing Host Checker Policies for Pulse Secure Client for Android Clients](#)
- [Pulse Secure Client for Mobile Devices Overview](#)

Implementing Host Checker Policies for Pulse Secure Client for Android Clients

After you create one or more Host Checker policies for Android devices, you must implement them. Pulse Connect Secure can use Host Checker policies at the realm or the role level.

Realm Authentication—You can configure a realm authentication policy to download and run Host Checker with a particular Host Checker policy. If the Android device does not meet the Host Checker requirements, then Pulse Connect Secure can deny access. You can provide remediation information in the Host Checker policy to describe the requirement and help users take steps to solve the issue.

To enable a Host Checker policy for a realm:

1. From the admin console, select **Users > User Realms > SelectRealm > Authentication Policy > Host Checker**. The Host Checker page displays all of the available Host Checker policies.
2. Select the check box next to each policy you want to include. Select one or both of the following check boxes next to the policy:
 - **Evaluate Policies**—Evaluates without enforcing the policy on the Android device and allows access.
 - **Require and Enforce**—Requires that the Android device be in compliance with the Host Checker policy. Pulse Connect Secure downloads Host Checker to the Android device after the user is authenticated and before the user is mapped to any roles in the system. Selecting this option automatically enables the Evaluate Policies option.
3. Optionally select **Allow access to realm if any ONE of the selected "Require and Enforce" policies is passed**. This check box is available if you selected more than one Host Checker policy. If you enable this check box, an Android device is allowed access if it passes any of the Require and Enforce policies. The Cache Cleaner policy does not apply to Android devices.
4. Click **Save Changes**.

Role—You can configure a role to download and run Host Checker with a particular Host Checker policy. If the Android device does not meet the Host Checker requirements, then Connect Secure can deny access or assign the user to a remediation role that has limited access. You can provide remediation information in the Host Checker policy to help users take steps to solve the issue.

To enable a Host Checker policy for a role:

1. From the admin console, select **Users > User Roles > SelectRole > General > Restrictions > Host Checker**. The Host Checker page displays all of the available Host Checker policies.
2. Select **Allow users whose workstations meet the requirements specified by these Host Checker policies**.
3. In the Available Policies list, select the policies that you want to apply to select them, and then click Add to move them to the Selected Policies list. To select a policy click it. To select more than one policy, use Ctrl+click.

4. Optionally select **Allow access to realm if any ONE of the selected “Require and Enforce” policies is passed**. This check box is available if you selected more than one Host Checker policy. If you enable this check box, an Android device is allowed access if it passes any of the Require and Enforce policies. The Cache Cleaner policy does not apply to Android devices.
5. Click **Save Changes**.

Related Documentation

- [Host Checker for Pulse Android Clients](#)
- [Configuring Host Checker for Pulse Secure Client Android Clients](#)

Pulse Secure Client for Android Error Message Reference

The following error message summary for Pulse Secure client for Android describes possible issues and suggests resolution actions where possible.

Table 25: Pulse Secure Client for Android Error Messages

Message	Possible Causes	Suggested Actions
The certificate for this server is invalid. Tap Accept to connect to this server anyway.	The server certificate for the page received from the Pulse server is not valid.	<p>Tap View Certificate to examine the certificate.</p> <p>Tap Accept and open the URL.</p> <p>Tap Decline. The connection attempt is ended. Contact the Pulse administrator.</p>
Session disconnected due to invalid certificate.	The Pulse client downloads session information from the Pulse server and the certificate received from the server does not match the stored session certificate. While accessing e-mail or intranet, the certificate received from Pulse server does not match the previously stored session certificate.	<p>Click the Close button on the Alert dialog to return to home screen.</p> <p>Try the operation again. If the error occurs again, contact the Pulse administrator.</p>
On the Pulse client status screen, the VPN status appears as Not Supported.	The device does not support VPN.	<p>Click the Close button on the Alert dialog to return to home screen.</p> <p>Try the operation again.</p>
Failed to connect to the server.	Sign-in Process failed.	Check the network connection (for example, Wi-Fi, 3G, etc.), and then try the operation again.
Failed to process the HTML information from the server.	Sign-in Process failed.	Try the operation again. If the error occurs again, contact the Pulse administrator.
Your session was terminated. Please connect again.	An invalid URL was used while accessing intranet and e-mail pages.	Reconnect to the Pulse server. If the error occurs again, contact the Pulse administrator.
Compliance Check couldn't be completed.	Host Checker compliance check could not be completed during sign-in process.	Try the operation again. If the error occurs again, contact the Pulse administrator.

Related Documentation

- [Pulse Secure Client for Android Overview](#)
- [Configuring a Role and Realm for Pulse for Android](#)

Box-Introduction to the Android Client API

The API for the Pulse Secure Android client allows a third-party application to establish a VPN with the Pulse Secure Access Service. In addition to adding, changing, and deleting VPN connections, the API lets you check the current API version, verify whether a VPN already exists, and add a certificate to the Android key store.

API Access Requirement

To access the API, the package name and signing certificate of the calling application must be submitted to Pulse Secure, LLC for incorporation into the Android client. The client stores an MD5 checksum of the following string:

```
<package name> + ":" + <signing signature>
```

If the application has multiple signing signatures, only one of the signatures is needed.

Package Name

net.pulsesecure.pulsesecure

AIDL File

The AIDL file used to define the client interface is named `IVpnProfile.aidl` and contains the following:

```
package net.pulsesecure.pulsesecure.vpnprofile;

//Package name updated in API version 3.

interface IVpnProfile {

    int doCommand(String commandXML); //deprecated

    int getVersion();

    // The following methods were added in API version 2.

    int createConnection(String jsonProfile);

    int removeConnection(String profileName);

    List<String> getAllConnections();

    String getConnection(String profileName);

    int startConnection(String profileName);

    int stopConnection(String profileName);

    int getState(String profileName);

    String getErrorString(String profileName);
}
```

This file must be included in the `src/` directory used to compile the application. For more information about the AIDL, go to:

<http://developer.android.com/guide/components/aidl.html>

Service intent filter action name

Use below action name to bind to service implementing this API. **Updated in API version 3.**

"net.pulsesecure.pulsesecure.vpnprofile.IVpnProfile"

JSON String Format for createConnection

The `createConnection` method, new in version 2 of the API, accepts a VPN profile described in a JSON string format. The JSON strings vary depending on the type of authentication used in the VPN profile, as shown in the following examples.

Single Username/Password Authentication

The following sample JSON string is for a VPN profile that uses a username/password for authentication:

```
{
  "MDM_VPN_PARAMETERS": {
    "profile_attribute": {
      "profileName": "MyVpnProfile",
      "host": "https://vpn.xyz.com/auth",
      "vpn_type": "ssl",
      "isUserAuthEnabled": true
    },
    "ssl": {
      "basic": {
        "username": "testuser",
        "password": "secret123"
      }
    },
    "vendor": {
      "realm": "",
      "role": "",
      "certAlias": ""
    }
  }
}
```

Certificate Authentication

The following sample JSON string is for a VPN profile that uses a certificate for authentication:

```
{
  "MDM_VPN_PARAMETERS": {
    "profile_attribute": {
      "profileName": "MyVpnProfile",
      "host": "https://vpn.xyz.com/auth",
      "vpn_type": "ssl",
      "isUserAuthEnabled": true
    },
    "ssl": {
      "basic": {

```

```

    },
    "vendor": {
        "realm": "",
        "role": "",
        "certAlias": "myCertName"
    }
}

```

Note that the certificate must be installed in Android's keystore before this profile can be used. The certificate name is specified in the `certAlias` field in the `vendor` section.

Username/Password and Certificate Authentication

The following sample JSON string is for a VPN profile that uses both certificate and username/password for authentication:

```

{
    "MDM_VPN_PARAMETERS": {
        "profile_attribute": {
            "profileName": "MyVpnProfile",
            "host": "https://vpn.xyz.com/auth",
            "vpn_type": "ssl",
            "isUserAuthEnabled": true
        },
        "ssl": {
            "basic": {
                "username": "testuser",
                "password": "secret123"
            }
        },
        "vendor": {
            "realm": "",
            "role": "",
            "certAlias": "myCertName"
        }
    }
}

```

Dual Username/Password Authentication

The following sample JSON string is for a VPN profile that uses two usernames/passwords for authentication:

```

{
    "MDM_VPN_PARAMETERS": {
        "profile_attribute": {
            "profileName": "MyVpnProfile",
            "host": "https://vpn.xyz.com/auth",
            "vpn_type": "ssl",
            "isUserAuthEnabled": true
        },
        "ssl": {
            "basic": {
                "username": "testuser",

```

```

        "password": "secret123"
    },
    "vendor": {
        "realm": "",
        "role": "",
        "username2": "testuser2",
        "password2": "passwd123"
    }
}

```

Note that the primary username/password are specified in the `ssl` section and the secondary username/password are specified in the `vendor` section.

XML Command Format for doCommand (Deprecated)

Many commands are specified by the `commandXML` argument of the interface `doCommand`. For a single command, the general format is:

```

<command>
  <commandName>command</commandName>
  <param1>value1</param1>
  <param2>value2</param2>
  // additional parameters for the command
</command>

```

Any parameter that is specified with an empty value is ignored.

The `commandXML` argument should specify a single command. However, multiple commands can be specified as follows:

```

<commands>
  <command>
    .
    .
    .
  </command>
  <command>
    .
    .
    .
  </command>
</commands>

```

Multiple commands are executed in the order specified, and execution stops on the first error. The `doCommand` method returns a negative integer if an error occurs on the first command; otherwise, the number of commands executed successfully is returned. If an error occurs after the first command, it is logged in the Pulse client log.

Return Codes for doCommand

The `doCommand` method returns a negative integer if an error occurs or a positive value (1 or higher) to indicate the number of commands executed successfully. The following table lists all of the possible return codes.

Table 26: Return Codes for doCommand

Return Code Code	Description
1 or n	Number of commands executed successfully

-1	COMMAND_EXECUTION_ERROR
-2	COMMAND_UNKNOWN
-3	KEY_OR_CERT_UNEXPECTED
-4	KEY_OR_CERT_MISSING
-5	PROFILE_DELETE_FAILED
-6	PROFILE_UPDATE_FAILED
-7	PROFILE_ADD_FAILED
-8	PROFILE_NOT_FOUND
-9	PROFILE_ALREADY_EXISTS
-10	COMMAND_XML_INVALID
-11	COMMAND_NOT_SUPPORTED
-12	CALLER_NOT_VERIFIED
-13	CALLER_NOT_IDENTIFIED
-14	CERT_ALIAS_NOT_FOUND_IN_KEYSTORE
-15	INCORRECT_BASE64_KEY_OR_CERT
-16	INCORRECT_KEY_OR_CERT_FILE
-17	DUPLICATE_KEY_OR_CERT_ENTRIES
-18	REQUIRED_PARAMETER_MISSING
-19	CHECK_PROFILE_URL_MISMATCH
-20	CHECK_PROFILE_USERNAME_MISMATCH
-21	CHECK_PROFILE_REALM_MISMATCH
-22	CHECK_PROFILE_ROLE_MISMATCH
-23	CHECK_PROFILE_CERT_PATH_MISMATCH
-24	CHECK_PROFILE_KEY_PATH_MISMATCH
-25	CHECK_PROFILE_CERT_ALIAS_MISMATCH

API Functions

This chapter describes each of the function calls for the Android client API.

getVersion()

Retrieves the version number of the client API.

Syntax:

```
getVersion()
```

Parameters:

None.

Returns:

An integer indicating the API version (the first version is 1). The current version is 3.

Since:

5.0R1

createConnection(String jsonProfile)

Creates a new VPN profile.

Syntax:

```
int createConnection(String jsonProfile)
```

Parameters:

The `jsonProfile` is the VPN profile in the JSON string format. See the examples in [JSON String Format for createConnection](#).

Returns:

This method returns 0 on success and 1 on error. When this method returns 1, the caller can call `getErrorString()` to obtain a description of the error.

Since:

5.0R3

removeConnection(String profileName)

Removes an existing VPN profile.

Syntax:

```
int removeConnection(String profileName)
```

Parameters:

The `profileName` is the name of a VPN profile that was created with the [addVPNConnection](#) or the [createConnection\(String jsonProfile\)](#) function.

Returns:

This method returns 0 on success, 1 if the profile does not exist, and -1 on other errors. One of the other errors occurs when an active VPN session is using the profile. When this method returns 1 or -1, the caller can call `getErrorString()` to obtain a description of the error.

Since:

5.0R3

getAllConnections()

Gets the profile names for all connections added by the same caller.

Syntax:

```
List<String> getAllConnections()
```

Parameters:

None

Returns:

This method returns the profile names for all VPN profiles added by the same caller. VPN connections added by the device user or by other callers are not included. This method could return null or an empty list.

Since:

5.0R3

getConnection(String profileName)

Obtains the JSON profile for a VPN profile created with the [createConnection\(String jsonProfile\)](#) method.

Syntax:

```
String getConnection(String profileName)
```

Parameters:

The `profileName` is the name of the VPN profile.

Returns:

This method returns the VPN profile for the name specified. This profile must be added by the same caller. This method could return null.

Since:

5.0R3

startConnection(String profileName)

Starts a VPN session using the specified VPN profile.

Syntax:

```
int startConnection(String profileName)
```

Parameters:

The `profileName` is the name of the VPN profile.

Returns:

This method returns 0 on success, 1 if the profile does not exist, and -1 on other errors.

When `startConnection()` returns 0, it means that a VPN session is getting started. It is possible that a VPN session may not be established after `startConnection()` returns 0. The caller can call `getState()` to obtain the VPN state. When the VPN fails to start, the caller can call `getErrorMessage()` to obtain a description for the error.

Since:

5.0R3

stopConnection(String profileName)

Stops an active VPN session for the specified VPN profile.

Syntax:

```
int stopConnection(String profileName)
```

Parameters:

The `profileName` is the name of the VPN profile.

Returns:

This method returns 0 on success, 1 if the profile does not exist, and -1 on other errors.

When `stopConnection()` returns 0, it means that a VPN session is being shut down, but is not yet complete. The caller can call `getState()` to obtain the VPN state.

Since:

5.0R3

getState(String profileName)

Get the current VPN state for the specified VPN profile.

Syntax:

```
int getState(String profileName)
```

Parameters:

The `profileName` is the name of the profile.

Returns:

The possible return values are:

```
private static final int STATE_IDLE = 1;
private static final int STATE_CONNECTING = 2;
private static final int STATE_DISCONNECTING = 3;
private static final int STATE_CONNECTED = 4;
private static final int STATE_FAILED = 5;
private static final int STATE_DELETED = 6;
```

When the return value is `STATE_FAILED`, the caller can call `getErrorString()` to obtain a description of the VPN failure.

Since:

5.0R3

getErrorString(String profileName)

Get the error string for the last API call or for the VPN failure.

Syntax:

```
String getErrorString(String profileName)
```

Parameters:

The `profileName` is the name of the profile.

Returns:

This method returns a string describing the error for the last API call. A No Error string will be returned when there is no error. When `getState()` returns `STATE_FAILED`, you can call this method to get a description of the error that prevented the service from starting a VPN session.

Since:

5.0R3

doCommand(String commandXML)

This API is deprecated. Use API that uses JSON string.

The `commandXML` argument of the `doCommand` supports the following functions:

- [addVPNConnection](#),
- [checkVPNConnection](#),
- [updateVPNConnection](#),
- [deleteVPNConnection](#),
- [addToKeyStore](#),

addVPNConnection

Adds a VPN profile that a user can select to connect to a Secure Access Service.

Note: VPN connection profiles created with this function cannot be changed or deleted by the user.

Syntax:

doCommand (commandXML)

commandXML:

```
<command>
  <commandName>addVPNConnection</commandName>
  <connectionName>vpnName</connectionName>
  <url>vpnServerUrl</url>
  <username></username>           // Optional
  <realm></realm>                 // Optional
  <role></role>                   // Optional
  //One of the following (all are optional):
    <aliasName></aliasName>
  //or
    <base64Key></base64Key>
    <base64Cert></base64Cert>
  //or
    <derKeyFile></derKeyFile>
    <derCertFile></derCertFile>
  //or
    <pemKeyFile></pemKeyFile>
    <pemCertFile></pemCertFile>
  <makeDefault>true/false</makeDefault> // Optional
</command>
```

Parameters:

- **connectionName** – REQUIRED – The name the user selects to create the VPN connection.
- **url** – REQUIRED – The URL of the Secure Access Service.
- **username** – Name of an authentication realm defined by the Secure Access Service. The realm defines the server used to authenticate the device.
- **realm** – Name of an authentication realm defined by the Secure Access Service. The realm defines the server used to authenticate the device.
- **role** – Name of the user role defined by the Secure Access Service. The user role defines the network resources the device can access.
- One of the following (all are optional):
 - **aliasName** – Name of a certificate in the Android key store to be used for authentication (Android 4.x or later required).
- or
 - **base64Key** – Base64-encoded private key.
 - **base64Cert** – Base64-encoded certificate.

or

- **derKeyFile** – Path name of the file containing the private key in binary format (such as `/sdcard/cert/key.der`).
- **derCertFile** – Path name of the file containing the certificate in binary format.

or

- **pemKeyFile** – Path name of the file containing the private key in Base64-encoded format.
- **pemCertFile** – Path name of the file containing the certificate in Base64-encoded format.

Note: The Pulse client requires read access to the key and certificate files, which must not be password protected. For security reasons, the caller should delete the files after the call.

- **makeDefault** – Indicates whether Pulse attempts to start the VPN when the user clicks **Connect** on the VPN home screen (true or false). The user can change the default VPN at any time.

Returns:

If the connection name already exists, a -9 is returned (see [Return Codes](#)).

Since:

5.0R1

checkVPNConnection

Verifies whether a VPN profile already exists. Only the `connectionName` is required. Pulse verifies each additional parameter specified, except for `makeDefault`.

Syntax:

`doCommand (commandXML)`

`commandXML`:

```
<command>
  <commandName>checkVPNConnection</commandName>
  <connectionName>vpnName</connectionName>
  <url>vpnServerUrl</url>           // Optional
  <username></username>             // Optional
  <realm></realm>                   // Optional
  <role></role>                     // Optional
  //One of the following (all are optional):
  <aliasName></aliasName>
  //or
  <base64Key></base64Key>
  <base64Cert></base64Cert>
  //or
  <derKeyFile></derKeyFile>
  <derCertFile></derCertFile>
  //or
  <pemKeyFile></pemKeyFile>
  <pemCertFile></pemCertFile>
</command>
```

Parameters:

- **connectionName** – REQUIRED – (string) The connection name to be verified.
- Any of the parameters used to add a VPN profile can be specified, except for `makeDefault` (see [addVPNConnection](#)).

Returns:

If the connection name does not exist, a -8 is returned. If the connection name is found, but one of the specified parameters does not match, the return code indicates the mismatched parameter (see [Return Codes](#)). If all the specified parameters match the existing values, the return code is 1.

Since:

5.0R1

updateVPNConnection

Updates a VPN profile that was created with the [addVPNConnection](#) function. VPN profiles created by an application can be updated only by that application and not by the user.

Syntax:

doCommand (commandXML)

commandXML:

```
<command>
  <commandName>updateVPNConnection</commandName>
  <connectionName>vpnName</connectionName>
  <url>vpnServerUrl</url>
  <username></username>           // Optional
  <realm></realm>                 // Optional
  <role></role>                   // Optional
  //One of the following (all are optional):
    <aliasName></aliasName>
  //or
    <base64Key></base64Key>
    <base64Cert></base64Cert>
  //or
    <derKeyFile></derKeyFile>
    <derCertFile></derCertFile>
  //or
    <pemKeyFile></pemKeyFile>
    <pemCertFile></pemCertFile>
  <makeDefault>true/false</makeDefault> // Optional
</command>
```

Parameters:

- connectionName – REQUIRED – The name the VPN profile to be changed.
- url – REQUIRED – The URL of the Secure Access Service. This can be the current URL or a new one.
- The parameters that you want to update (see [addVPNConnection](#)).

Returns:

If the connection name does not exist or the application did not create the profile, a -8 is returned (see Return Codes).

Since:

5.0R1

deleteVPNConnection

Deletes a VPN profile that was created with the addVPNConnection or the createConnection (String jsonProfile) function. VPN profiles created by an application can be deleted only by that application and not by the user.

Syntax:

doCommand (commandXML)

commandXML:

```
<command>
  <commandName>deleteVPNConnection</commandName>
  <connectionName>vpnName</connectionName>
</command>
```

Parameters:

- connectionName – REQUIRED – The name of VPN profile to be deleted.

Returns:

If the connection name does not exist or the application did not create the profile, a -8 is returned (see [Return Codes](#)).

Since:

5.0R1

addToKeyStore

Adds a certificate to the Android key store without creating a VPN connection profile (Android 4.x or later required).

Syntax:

doCommand (commandXML)

commandXML:

```
<command>
  <commandName>addToKeyStore</commandName>
  <passwordProtectedCertFile></passwordProtectedCertFile>
</command>
```

Parameters:

- passwordProtectedCertFile – REQUIRED – Password protected file containing Base64-encoded private key and certificate.

Returns:

A positive or negative integer (see [Return Codes](#)).

Since:

5.0R1

Launching the Pulse Secure Client for Android App through a Browser Link

Android device users can launch Pulse from the browser on their device by logging into the Pulse Connect Secure Web portal, and then clicking the VPN link. The default page that appears when an Android device user logs into the Web portal already includes a VPN link to launch Pulse.

Tapping the VPN link on the Web page launches the Pulse app if it is not already running. If Pulse is not already installed on the Android device, an error occurs. You cannot deploy the Pulse for Android app from the Pulse server.

When a user taps a button that is tied to a Pulse launcher command, the command launches the Pulse app if it is not already running. If Pulse is not already installed on the iOS device, an error occurs. The next step depends on the current Pulse connection status and configuration. One of the following occurs:

- If Pulse does not already have an active connection to Pulse Connect Secure, it uses an existing configuration to establish the VPN connection.
- If Pulse does not already have an active connection, and it does not already have a configuration for the target Pulse server, Pulse opens the Add Configuration screen. The target URL is already defined and the user just needs to specify a name for the connection.
- If the Pulse app is already connected to a Pulse server, the Pulse app is brought to the foreground.

To employ the Pulse launcher in your Web pages or external applications, specify the link using the following format:

**pulsesecure://<server-host>/<server-path> ?method={vpn} &action={start|stop} &DSID=<dsid-cookie>
&SMSESSION=<smsession-cookie> &username=<username> &password=<password> &realm=<realm> &role=<role>**

Usage notes:

- If the DSID cookie is given in the URL, the app does not use the "username", "password", "realm", or "role" parameters because no login is required.
- The values for username, realm, and role are URI-escaped values. Special characters are replaced with their hexadecimal equivalents preceded by '%'.
- If the user has specified the username, realm, and role when creating the VPN configurations in the Pulse Secure app, those values are used to auto-fill the username, realm, and role for the login pages during a Web-based login. During login, if all fields are successfully auto-filled from fields in the VPN configuration or the pulsesecure:// launch URL, the login progresses without any user input. The username, realm, and role values need to already exist in the VPN configuration for them to be auto-filled during the login process. If the user manually specifies the username, realm, or role during login, the app will not add or update these values in the VPN configuration. The user needs to explicitly update the VPN configuration with these values.
- If the user manually specifies the username, realm, or role during login, the app stores these values in the VPN configuration and they will be auto-filled the next time the user signs in. Passwords entered by the user are not saved in the VPN configuration.

Examples:

If the calling application has already obtained a DSID cookie from Pulse Connect Secure, the app can use the following command to start the VPN:

pulsesecure://<server-host>/<server-path> ?method=vpn &action=start &DSID=<dsid-cookie> &SMSESSION=<smsession-cookie>

If the calling application does not already have a DSID, it can use the following command to start the VPN:

**pulsesecure://<server-host>/<server-path> ?method=vpn&action=start &username=<username> &password=<password>
&realm=<realm> &role=<role>**

If the calling application wants to stop the VPN, it can use the following command:

pulsesecure://<server-host>/<server-path> ?method=vpn &action=stop

Related Documentation

- [Launching the Pulse Secure Client for Android App Using a Command](#)

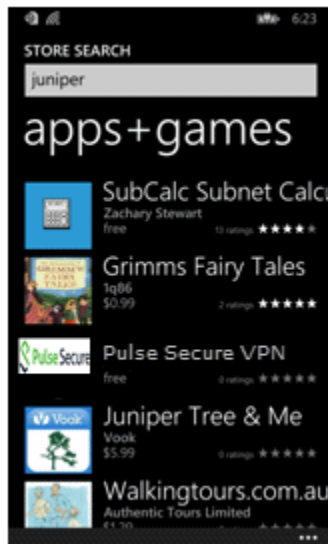
CHAPTER 12 Pulse Secure Client for Windows Phone

- Pulse Secure Client for Windows Phone Overview
- Configuring Pulse Connect Secure for Pulse Secure Client for Windows Phone VPN Connections
- Configuring a Pulse Secure Connection for Windows Phone – Manual Configuration
- Host Checker for Pulse Secure Client for Windows Phone
- Configuring Host Checker for Pulse Secure Client for Windows Phone
- Implementing Host Checker Policies for Pulse Secure Client for Windows Phone

Pulse Secure Client for Windows Phone Overview

Pulse Secure client for Windows Phone provides secure connectivity between a Windows Phone and Pulse Connect Secure. Pulse for Windows Phone is available from the [Windows Phone Store](#). After installing the **Pulse Secure VPN** app on a Windows Phone (Windows Phone 8.1 or later), the user can configure a connection and establish Layer 3 VPN (SSL) communications. Pulse Secure client for Windows Phone can also be configured through mobile device management (MDM).

Figure 117: Mobile Device Management



Configuration on the Pulse server to support Pulse Secure client for Windows Phone is the same as for the Pulse for Windows client. You use sign-in policies, authentication realms, roles, and VPN tunnel policies to define authentication and access permissions. A typical Pulse server configuration for Windows Phone access is to create a realm, a role, and a remediation role that are designed for Windows Phone users.



Note: Users must have their server's root certificate installed on the Windows Phone before attempting to connect to the Pulse server to avoid the error, Server certificate is untrusted.

Pulse Secure Client for Windows Phone Supported Platforms

Pulse Secure client for Windows Phone is supported on Windows Phone 8.1 and later.

Pulse Secure client for Windows Phone is supported on Pulse Secure client Secure Access Service R8.0 and Pulse Connect Secure 8.1 and later. Host Checker support for Pulse for Windows Phone requires Pulse Secure client Secure Access Service 8.0R5 or Pulse Connect Secure 8.1 and later.

Pulse Secure Client for Windows Phone Supported Features

The following list describes the supported features for the Pulse Secure client for Windows Phone client.

- Pulse Secure client for Windows Phone supports VPN (SSL) connections to a Pulse Secure client SSL/VPN server. Only one connection at a time can be active.
- The user can manually connect and disconnect.
- Username and password.
- Username and RSA token code. (User PIN and system PIN are supported.)
- Client certificate, smart card, and virtual smart card.
- Authentication server prompts for retry, change password, create PIN, change PIN, and specify next token code.
- Realm and role selection and preferred realm and role. (The user cannot choose to save a connection preference.)
- Sign-in notification messages.
- Secondary authentication.
- HTTPS proxy.
- IPv4 and IPv6.
- Host Checker.

You must create and enable the Windows Phone OS Check rule if you are using a Host Checker policy that is applied across endpoints running different operating systems. Host Checker for Windows Phone is supported on Pulse Secure Access Service 8.0R5 and later with Endpoint Security Assessment Plugin (ESAP) 2.6.3 and later, and Pulse Connect Secure 8.1 and later.

Pulse Secure client for Windows Phone supports the following tunneling functions:

- Split tunneling enabled or disabled.



Note: Pulse for Windows Phone connections always have local subnet access enabled.

- SSL-VPN connections.
- Split tunneling policies: IPv4 inclusion and exclusion routes, and IPv6 inclusion routes.
- In split-tunneled mode, the DNS search order options do not apply. Pulse forwards only those DNS requests contained by the configured DNS suffixes to the specified DNS servers. You can specify the VPN option **Search device DNS only** to forward all DNS requests to configured DNS servers.

Pulse Secure Client for Windows Phone Limitations

The following Pulse features are not available with Pulse Secure client for Windows Phone:

- Save realm or role preference
- Machine authentication
- Location awareness rules
- Logon and logoff scripts
- WINS server tunnel parameter
- UDP-ESP tunnel (SSL mode only)
- Certificate trust override prompt
- RSA soft-token integration
- Session extension
- Suspend/resume tunnel
- Single sign-on using Security Assertion Markup Language (SAML)

Related Documentation

- [Configuring Pulse Connect Secure for Pulse Secure Client for Windows Phone VPN Connections](#)
- [Configuring a Pulse Secure Connection for Windows Phone – Manual Configuration](#)

Configuring Pulse Connect Secure for Pulse Secure Client for Windows Phone VPN Connections

Pulse Secure client enables you to secure your company resources using authentication realms, user roles, and resource policies. For complete information on the Pulse access management framework, see the [Pulse Secure server documentation](#).

A Pulse server checks the authentication policy defined for the authentication realm. The user must meet the security requirements you define for a realm's authentication policy, or else the Pulse server does not forward the user's credentials to the authentication server. At the realm level, you can specify security requirements based on various elements such as the user's source IP address or the possession of a client-side certificate. If the user meets the requirements specified by the realm's authentication policy, the Pulse server forwards the user's credentials to the appropriate authentication server. If this server successfully authenticates the user, then the Pulse server evaluates the role mapping rules defined for the realm to determine which roles to assign to the user.



Note: Users must have their server's root certificate installed on the endpoint before attempting to connect to the Pulse server to avoid the error, Server certificate is untrusted.



Note: Pulse for Windows Phone supports Host Checker. You must create and enable the Windows Phone OS Check rule if you are using a Host Checker policy that is applied across endpoints running different operating systems. Host Checker for Windows Phone is supported on Pulse Secure Access Service 8.0R5 and later and Pulse Connect Secure 8.1 and later and Pulse Connect Secure 8.1 and later with Endpoint Security Assessment Plugin (ESAP) 2.6.3 and later.

The following is a generalized example of configuring a Pulse server for the Pulse for Windows Phone app.

1. Click **Users > User Roles**, and then create a new role.

You can use an existing role. However, because Host Checker supports different options for each type of device operating system, a typical approach is to create different roles for different devices.

2. Specify a name and optional description for the role, for example, **WinPhoneRole, Windows Phone VPN role**.
3. To use certificate authentication at the role level, click **Restrictions > Certificate** on the role's General tab, and add the required certificate information.
4. Enable certificate authentication by clicking **Only allow users with a client-side certificate signed by Certification Authority to sign in**.

One typical method of installing the client certificate on the Windows Phone is to send the certificate as an attachment to the Windows Phone user. The certificate must be installed on the Windows Phone before the user can connect. The user is prompted to select the certificate during the initial Pulse VPN connection process.

5. Define the client certificate, click **Add**, and then click **Save Changes**.

For complete information on certificate authentication, see [Understanding Digital Certificate Security](#).

6. Set the options on the role's Web and Files tabs as needed.
7. Click **Users > User Realms**, and then create a new realm or select an existing realm.

Configure and save your options on the General and the Authentication Policy tabs.

8. On the Role Mapping tab, click **New Rule** to create a new role mapping rule.

One option for a role mapping rule is to create a custom expression that uses the user agent string to identify a Windows Phone. The Pulse for Windows Phone user agent string is **Junos-Pulse/7.4.0.0 (Windows Phone; ARM) JunosPulseVpn/1.0.1.5**. You can use all or part of the string in a custom expression that uses the `userAgent` variable. For example, `userAgent = **Windows Phone**`.

9. Select the role that you created earlier for the Windows Phone users, add it to the Selected Roles list, and then click **Save Changes**

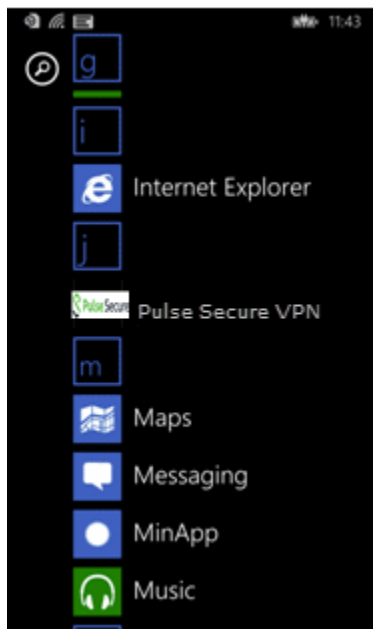
Related Documentation

- [Pulse Secure Client for Windows Phone Overview](#)
- [Configuring a Pulse Secure Connection for Windows Phone – Manual Configuration](#)

Configuring a Pulse Secure Connection for Windows Phone – Manual Configuration

Pulse Secure client for Windows Phone is available from the [Windows Phone Store](#). After the user installs the app, the user can create Pulse Secure VPN connections. [Figure 118](#) shows Pulse Secure VPN after it has been installed on a Windows Phone.

Figure 118: Windows Phone Apps List



Note: To configure a VPN connection, or to initiate a manual VPN connection, use Settings on the phone. Tapping Pulse in the apps list simply opens an information screen.

You create, manage, and delete Pulse connections by using Windows Phone Settings. Pulse connections appear as VPN connections in the Networks list.



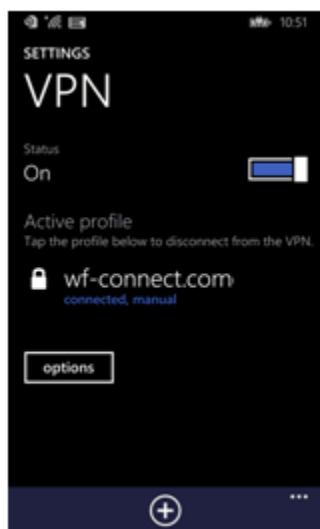
Note: The client certificate must be installed on the Windows Phone before Pulse can connect. One typical way of installing a certificate is to e-mail it to the user. The user simply taps the certificate in the e-mail and Windows Phone installs it.

To create a Pulse Secure VPN connection on a Windows Phone:

1. Tap **Settings**, and then tap **VPN**.

If the status slider is set to On, the phone displays a list of existing VPN connections. [Figure 119](#) shows the Windows dialog where you configure the connection.

Figure 119: Manually Adding a Pulse Connection



2. To create a new connection, tap the plus icon at the bottom of the screen. The Add Profile screen appears.
3. In the Server name or IP address box, specify the target for this connection.

You can identify the server using the server IP address, the hostname, or a URL that optionally specifies the port the connection uses and the sign-in policy. To specify a URL, use the following format:

`https://hostname[:port][/?][sign-in policy]`

The brackets indicate options. If you specify a sign-in policy, be sure that the name you specify matches what is defined on the Pulse server. (Authentication > Signing in > Sign-in Policies.)

4. Tap the Type box to expand it, and then tap **Pulse Secure VPN** to select it.
5. Specify a username and password.

If you specify a username and password, the prompt for this information does not appear when you activate the connection. For token code authentication, leave the username and password fields blank.

6. Enable or disable **Connect automatically** as needed.
7. The IP ranges option is available if you have enabled the Connect automatically slider. The IP ranges option lets you identify specific IP addresses that can trigger this Pulse VPN connection. When you attempt a connection to an IP address in the specified range, and that address is not reachable, the Pulse VPN connection is activated.
8. The Profile name defaults to the value you entered for Server name or IP address box. The Profile name appears in the VPN list; you can change it to something more meaningful.
9. Tap **Advanced** to set the following:
 - **Proxy**—If you enable the Proxy setting, the app opens a screen where you can specify the settings for connecting to the Pulse server through a proxy server.
 - **Don't use VPN on company WiFi**—When you are in the company office, network traffic uses the company WiFi network without first establishing a VPN connection.
 - **DNS suffix**—If you have automatic connections enabled, a request to access information within the specified domain name suffix causes Windows to initiate a VPN connection before connecting to the target.

- **Don't use VPN for home WiFi traffic**—Network traffic uses the home WiFi network without first establishing a VPN connection.

After the user saves the new connection, it appears in the VPN list. The user can tap the connection to initiate a VPN connection. When a VPN connection is active, a small lock icon appears next to the WiFi status icon.

Related Documentation

- [Pulse Secure Client for Windows Phone Overview](#)
- [Host Checker for Pulse Secure Client for Windows Phone](#)
- [Configuring Pulse Connect Secure for Pulse Secure Client for Windows Phone VPN Connections](#)

Host Checker for Pulse Secure Client for Windows Phone

Host Checker is a component of Pulse Secure client that reports the integrity of an endpoint that is attempting to connect to the Pulse sever. Host Checker runs as a Trusted Network Connect (TNC) client on the endpoint. Host Checker evaluates the endpoint according to predefined criteria and reports to the Trusted Network Connect server, which is a part of the Pulse server. If the endpoint is not in compliance with the Host Checker policies, then the user might not get access to the network or might get limited access to the network depending upon the enforcement policies configured by the administrator.

Pulse for Windows Phone Host Checker can evaluate client compliance based on the following predefined criteria:

- **OS Checks**—You can specify the Windows Phone version or minimal version that must be installed on the device. You must create and enable the Windows Phone OS Check rule if you are using a Host Checker policy that is applied across endpoints running different operating systems.



Note: Host Checker for Windows Phone is supported on Pulse Secure Access Service 8.0R5 and later and Pulse Connect Secure 8.1 and later with Endpoint Security Assessment Plugin (ESAP) 2.6.3 and later.

Related Documentation

- [Configuring Host Checker for Pulse Secure Client for Windows Phone](#)
- [Implementing Host Checker Policies for Pulse Secure Client for Windows Phone](#)

Configuring Host Checker for Pulse Secure Client for Windows Phone

Host Checker policies can be part of a larger Host Checker configuration that applies to many different types of clients or to Windows Phone devices only. You must create and enable the Windows Phone OS Check rule if you are using a Host Checker policy that is applied across endpoints running different operating systems.



Note: Host Checker for Windows Phone is supported on Pulse Secure Access Service 8.0R5 and later and Pulse Connect Secure 8.1 and later with Endpoint Security Assessment Plugin (ESAP) 2.6.3 and later.

To create a Host Checker policy for Windows Phone:

1. From the admin console, select **Authentication > Endpoint Security > Host Checker**.
2. In the Policies section, click an existing Host Checker Policy to open it or click **New** to open a New Host Checker Policy page.
3. For a new policy, specify a name for the policy and then click **Continue** to open the Host Checker Policy page.

The name appears in lists when you implement the policy so be sure to use a descriptive name, such as WinPhone HC Policy.

4. Click the Mobile tab, and then click the Windows Phone tab.
5. In the Rule Settings section, click the **Select Rule Type list box**, click Predefined: **OS Checks**, and then click **Add**.
6. Specify a descriptive name for this rule. For example, Must-Be-8.1-or-higher. Rule names cannot include spaces.

7. In the Criteria section, select the check box for **WindowsPhone VPN-Plugin**.
8. Click **Save Changes**.

Related Documentation

- [Host Checker for Pulse Secure Client for Windows Phone](#)
- [Implementing Host Checker Policies for Pulse Secure Client for Windows Phone](#)

Implementing Host Checker Policies for Pulse Secure Client for Windows Phone

After you create one or more Host Checker policies for Windows Phone devices, you must implement them. The Pulse server can use Host Checker policies at the realm or the role level.

To enable a Host Checker policy for a realm:

1. From the admin console, select **Users > User Realms > SelectRealm > Authentication Policy > Host Checker**. The Host Checker page displays all of the available Host Checker policies.
2. Select the check box next to each policy you want to include. Select one or both of the following check boxes next to the policy:
 - **Evaluate Policies**—Evaluates without enforcing the policy on the device and allows access.
 - **Require and Enforce**—Requires that the device be in compliance with the Host Checker policy. Selecting this option automatically enables the Evaluate Policies option.
3. Optionally select **Allow access to realm if any ONE of the selected “Require and Enforce” policies is passed**. This check box is available if you selected more than one Host Checker policy. If you enable this check box, a device is allowed access if it passes any of the Require and Enforce policies. The Cache Cleaner policy does not apply to Windows Phone devices.
4. Click **Save Changes**.

To enable a Host Checker policy for a role:

1. From the admin console, select **Users > User Roles > SelectRole > General > Restrictions > Host Checker**. The Host Checker page displays all of the available Host Checker policies.
2. Select **Allow users whose workstations meet the requirements specified by these Host Checker policies**.
3. In the Available Policies list, click the policies that you want to apply to select them, and then click **Add** to move them to the Selected Policies list. To select more than one policy, use Ctrl+click.
4. Optionally select **Allow access to realm if any ONE of the selected “Require and Enforce” policies is passed**. This check box is available if you selected more than one Host Checker policy. If you enable this check box, a device is allowed access if it passes any of the Require and Enforce policies. The Cache Cleaner policy does not apply to Windows Phone devices.
5. Click **Save Changes**.

Related Documentation

- [Host Checker for Pulse Secure Client for Windows Phone](#)
- [Configuring Host Checker for Pulse Secure Client for Windows Phone](#)

PART 4 Windows In-Box Pulse Secure Client

- [Windows Pro and Windows RT In-Box Pulse Secure Client](#)

CHAPTER 13 Windows Pro and Windows RT In-Box Pulse Secure Client

- [Pulse Secure VPN App for Chrome.](#)
- [Pulse Secure Universal App for Windows.](#)
- [Microsoft Windows In-Box Pulse Secure Client User Interface](#)
- [Windows PowerShell Script Examples for Microsoft Windows In-Box Pulse Secure Client](#)
- [Host Checker for Windows In-Box Pulse Secure Client Overview](#)
- [Configuring Host Checker for Windows In-Box Pulse Secure Client](#)
- [Implementing Host Checker Policies for Windows In-Box Pulse Secure Client](#)
- [Host Checker Statement of Health for Pulse Connect Secure Overview](#)
- [Configuring a Statement of Health Host Checker Policy for Pulse Connect Secure](#)
- [Enabling Statement of Health on the Windows Endpoint for the Windows In-Box Pulse Secure Client](#)
- [Viewing Windows In-Box Pulse Secure Client Log Messages](#)

Pulse Secure VPN App for Chrome

Pulse Secure client for Chrome OS provides secure connectivity between a device running Chrome OS and Pulse Connect Secure. Pulse Secure client for Chrome OS is available from the [Chrome Web Store](#). After installing the Pulse Secure client app on a Chrome OS device, the user can configure a connection and establish Layer 3 VPN (SSL) communications.

Configuration on the Pulse Connect Secure gateway to support Pulse Secure clients for Chrome OS is the same as that of Pulse for Windows and macOS. Use the sign-in policies, authentication realms, roles and VPN tunnel policies to define authentication and access permissions. A typical Pulse server configuration for Chrome OS access is to create a realm, a role and a remediation role that are designed for Chrome OS users.

Pulse Secure Universal App for Windows

The [Pulse Secure Universal App for Windows](#) is the successor to both the Pulse Secure “In-Box” VPN Plugin for Windows 8.1 and the Pulse Secure Windows Phone 8.1 app. The Universal App provides Layer 3 VPN (SSL) secure connections between a Windows 10 and later device (whether PC, tablet, smartphone, Xbox, or [Windows 10 IoT](#)) and a Pulse Connect Secure (PCS) gateway. The Pulse Secure Universal App, with support for localization, is available for download at the [Microsoft Store](#). (Microsoft calls this an *in-box* application.) The Windows in-box Pulse client appears as a VPN Provider network option within Windows 8.1 and later endpoints, including Windows RT endpoints. The user can establish a Layer 3 VPN connection to Pulse Connect Secure. You can create, manage, and remove Pulse VPN connections on the Windows endpoint through Windows PowerShell scripts. The user can also create connections manually on the endpoint. The Windows in-box Pulse client provides a subset of the features that are available through the Pulse Secure for Windows client. Windows PowerShell is a command-line shell and scripting language for Windows system administration. For more information about PowerShell, see the [PowerShell documentation](#) on Microsoft Tech Net. PowerShell commands are called *cmdlets*. For information about the VPN Client cmdlets, see the [Microsoft Tech Net topic on VPN Client cmdlets](#).

Configuration on the Pulse server to support the Windows in-box Pulse client is the same as for the Pulse for Windows client. You create sign-in, realm, role, VPN tunnel, and Host Checker policies. You can use the same roles and connection profiles for the Windows in-box Pulse client as you use for the Pulse for Windows client. However, the Windows in-box Pulse client supports only OS Check Statement of Health (SoH) Host Checker rules. Statement of Health is a component of Network Access Protection (NAP), a Microsoft policy enforcement platform built into Windows 8, Windows 7, Windows Vista, and Windows Server 2008 operating systems. NAP SoH lets you enforce system health compliance. Before you can use the Statement of Health Host Checker rule with the Windows in-box Pulse client, you must enable NAP SoH functionality on the endpoint.

Microsoft Windows In-Box Pulse Secure Client Supported Platforms

The Windows in-box Pulse client is supported on Pulse Secure Access Service R7.4 and later and Pulse Connect Secure 8.1 and later. Client upgrades are distributed through Windows Update and Windows HotFix distributions.



Note: OS Check and the Statement of Health rule are the only Host Checker rules supported by the Windows in-box Pulse client. The Statement of Health rule is available only with Pulse Secure Access Service (now Pulse Connect Secure) R8.0 and later. If you are using R7.4, then only basic Host Checker functions are available.

Microsoft Windows In-Box Pulse Secure Client Supported Features

The following list describes the supported features for the Windows in-box Pulse client.

- The Windows in-box Pulse client supports connections to Pulse Connect Secure. Only one connection at a time can be active.



Note: Users must have their server's root certificate installed on the endpoint before attempting to connect to the Pulse server to avoid the error, Server certificate is untrusted.

- The user can manually connect and disconnect. The Pulse administrator can also configure a Windows in-box Pulse VPN connection to connect automatically when the user starts a particular app.
- The Windows in-box Pulse client supports the following authentication functions:
 - Username and password.
 - Username and token code.
 - Authentication server prompts for retry, change password, create a PIN, change PIN, and specify next token code.
 - Realm and role selection and preferred realm and role.
 - Sign-in notification messages.
 - Secondary authentication.
 - Client certificate, smart card and virtual smart card support. (Suite B cryptography.)
 - RSA and ECDSA
 - TLS 1.2
 - PIN prompt
- Sign-in messages.
- Cached credentials.
- Secondary authentication.
- HTTPS proxy.
- IPv4 and IPv6.
- Host Checker (OS Check and Statement of Health rules only).
- Automatic tunnel connection application launch.

The Windows in-box Pulse client supports the following tunneling functions:

- Split tunneling enabled or disabled.



Note: Windows in-box Pulse client connections always have local subnet access enabled.

- SSL-VPN connections.
- Split tunneling policies: IPv4 inclusion and exclusion routes, and IPv6 inclusion routes.

- In split-tunneled mode, the DNS search order options do not apply. Pulse forwards only those DNS requests contained by the configured DNS suffixes to the specified DNS servers. You can specify the VPN option **Search device DNS only** to forward all DNS requests to configured DNS servers.
- Proxy server and Proxy Automatic Configuration (PAC) file.

The outer HTTPS connection can use system proxy settings. After the connection is established the proxy settings configured on the Pulse server are applied to the endpoint.

Windows OS integration support:

- Native Windows network (NUI) VPN plugin.
- Native Network and Sharing; Adapter settings.
- Secure stored saved username, password and PINs (Microsoft -RememberCredential option).

Microsoft Windows In-Box Pulse Secure Client Limitations

The Windows in-box Pulse client supports connections to Pulse Connect Secure only.

The following Pulse features are not available with the Windows in-box Pulse client:

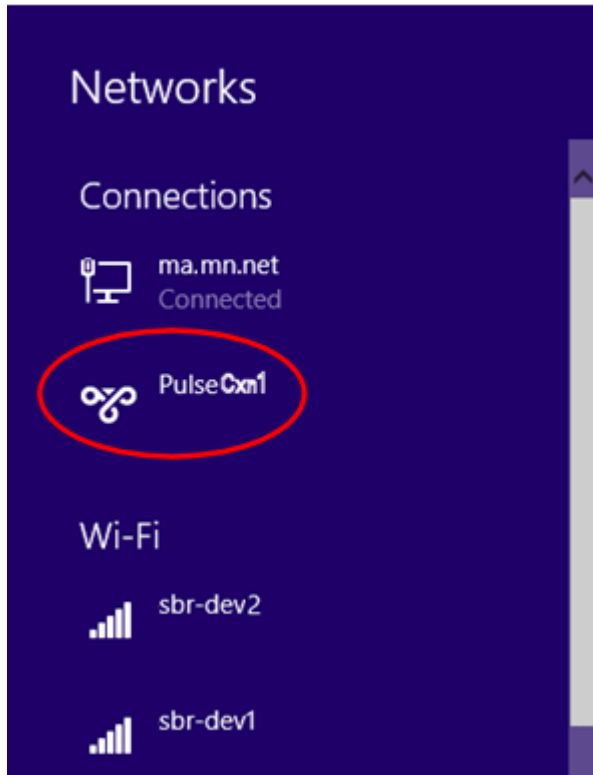
- Save realm or role preference
- Machine authentication
- Location awareness rules
- Logon and logoff scripts. (As defined in the role. As an alternative, you can use the Microsoft Task Scheduler (schtasks.exe) to run scripts on the endpoint.
- WINS server tunnel parameter
- UDP-ESP tunnel (SSL mode only)
- Certificate trust override prompt
- Proxied HTTPS connections
- Generalized URI connection addresses. (You can use **port** and **uri** options in the network connection schema instead).
- Preconfigured Pulse settings (.pulsepreconfig).
- RSA soft-token integration
- Session extension
- Suspend/resume tunnel
- Host Checker automatic remediation actions
- Pre logon VPN plugins

Related Documentation

- [Microsoft Windows In-Box Pulse Secure Client User Interface](#)
- [Windows PowerShell Script Examples for Microsoft Windows In-Box Pulse Secure Client](#)
- [Host Checker Statement of Health for Pulse Connect Secure Overview](#)
- [Configuring a Statement of Health Host Checker Policy for Pulse Connect Secure](#)
- [Enabling Statement of Health on the Windows Endpoint for the Windows In-Box Pulse Secure Client](#)
- [Viewing Windows In-Box Pulse Secure Client Log Messages](#)

Microsoft Windows In-Box Pulse Secure Client User Interface

Microsoft Windows 8.1 introduced Pulse Secure client as part of the Windows operating system. (Microsoft calls this an “in-box” application.) The Windows in-box Pulse client uses Windows operating system dialogs. In addition, there is no user assistance provided with the Windows in-box Pulse client. Pulse connections appear as Windows network connections. Users can create, manage, and delete Pulse connections. Pulse connections appear as VPN connections in the Networks list. [Figure 120](#) shows a Pulse VPN connection labeled PulseCxn1.

Figure 120: Windows Networks List

To manually create a new connection, the user should do the following:

1. Click **Settings > Change PC Settings > Network > Add a VPN Connection**.

Figure 121 shows the Windows dialog to configure the connection.

Figure 121: Manually Adding a Pulse Connection

Add a VPN connection

VPN provider
Juniper Networks Junos Pulse

Connection name
10.64.8.94

Server name or address
10.64.8.94

Type of sign-in info
User name and password

User name (optional)

Password (optional)

☒ Remember my sign-in info

Save Cancel

Pulse Secure appears as a selection in the **VPN provider** box. The **Connection name** can be anything the user wants but the user needs to provide either the IP address or the hostname (for example, server1.mycompany.net) of the Pulse server in the **Server name or address** box.



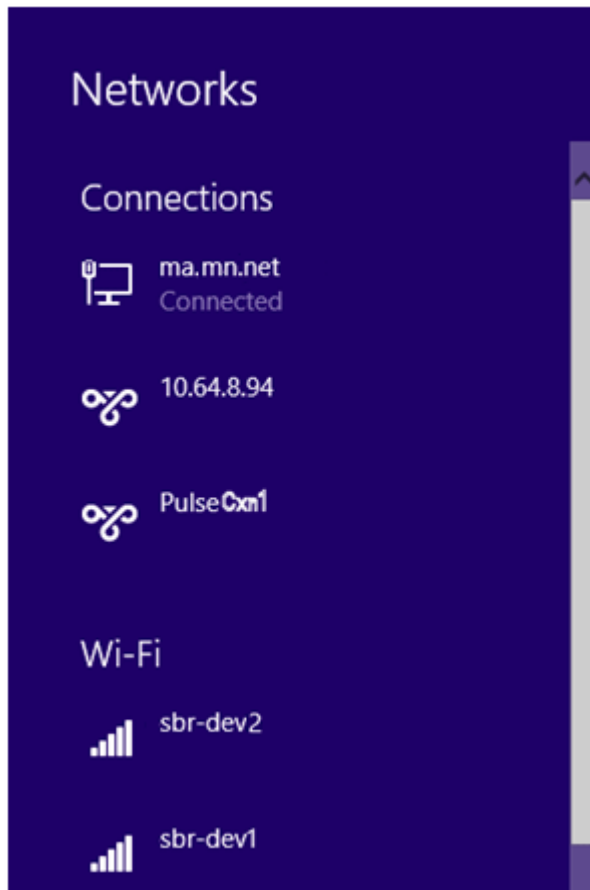
Note: To avoid an Invalid parameter error, do not include http or https in the server address or hostname.

Because credential prompts are configured on and served from the Pulse server, the **Type of sign-in info**, **User name**, and **Password** boxes are not available. If **Remember my sign-in info**, is selected, Windows preserves the values the user specifies for username and password during the first logon operation.



Note: Users must have their server's root certificate installed on the endpoint before attempting to connect to the Pulse server to avoid the error, Server certificate is untrusted.

After the user saves the new connection, it appears in the Networks list as shown in [Figure 122](#). The user can click the connection to initiate a VPN connection.

Figure 122: Windows Networks List with New Pulse Connection

Pulse connection prompts (Figure 123) and messages (Figure 124) appear as Windows user interface elements.

Figure 123: Pulse Secure Credentials Dialog

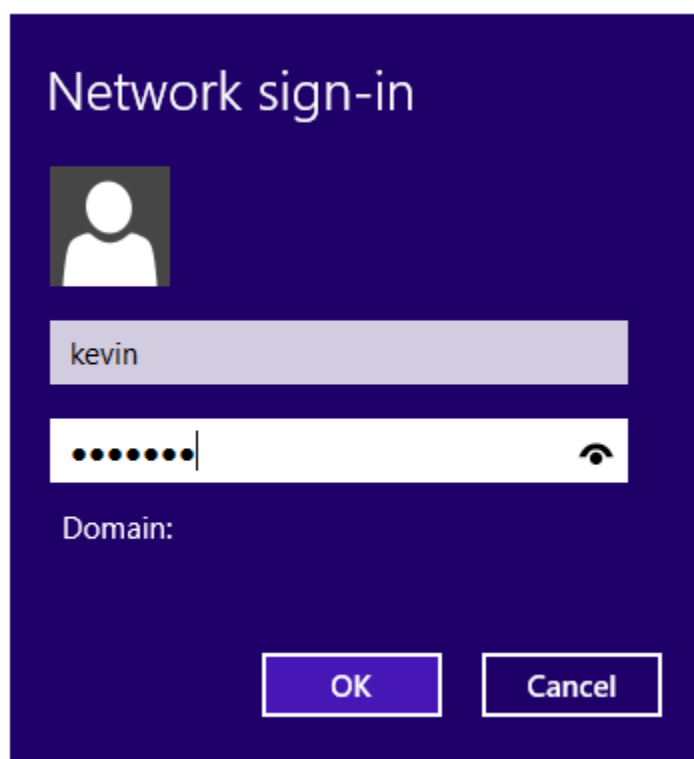
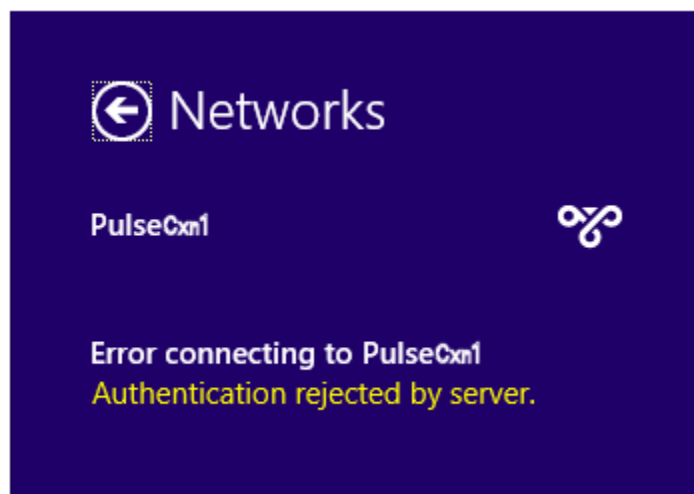


Figure 124: Pulse Secure Message Box



Related Documentation

- [Microsoft Windows In-Box Pulse Secure Client Overview](#)
- [Windows PowerShell Script Examples for Microsoft Windows In-Box Pulse Secure Client](#)
- [Host Checker Statement of Health for Pulse Connect Secure Overview](#)
- [Configuring a Statement of Health Host Checker Policy for Pulse Connect Secure](#)
- [Enabling Statement of Health on the Windows Endpoint for the Windows In-Box Pulse Secure Client](#)
- [Viewing Windows In-Box Pulse Secure Client Log Messages](#)

Windows PowerShell Script Examples for Microsoft Windows In-Box Pulse Secure Client

Microsoft Windows 8.1 introduced support for Pulse Secure client as part of the operating system. You can create, manage, and remove Pulse connections on the Windows endpoint by using Windows PowerShell scripts on the endpoint. PowerShell is a command-line shell and scripting language for system administration. To configure Pulse Secure Connections, you should have a working knowledge of PowerShell. For detailed information on PowerShell, see the [Microsoft Tech Net library](#).

Windows PowerShell commands are called cmdlets. To manage Pulse connections, you use the VPN Client cmdlets. For detailed information on the [VPN Client cmdlets](#), see the VPN Client section of the [Microsoft Tech Net library](#).



Note: PowerShell scripts must be signed to run on client computers that have a default PowerShell configuration. For more information, see the [Microsoft Tech Net library](#).



Note: You use Windows PowerShell scripts to administer Windows in-box Pulse client connections. Pulse for Windows connections do not respond to PowerShell scripts.

The following PowerShell script examples show how to create and manage the most commonly used PowerShell cmdlets to create and manage Pulse connection configurations. Most PowerShell VPN Client cmdlets require that you specify the application ID. For Pulse, the application ID is **JuniperNetworks.JunosPulseVpn_cw5n1h2txyewy**. For a complete list of cmdlet options, see the VPN Client section of the [Microsoft Tech Net library](#).



Note: All connections are HTTPS and use a server certificate, therefore you must install the server root CA to connect.

Add Pulse connection that uses split tunneling

This script creates a Pulse VPN connection named PulseCxn1 that connects to a Pulse server with an IP address of 10.17.1.216.

```
$xml = "<pulse-schema></pulse-schema>"
$sourceXml=New-Object System.Xml.XmlDocument
$sourceXml.LoadXml($xml)
```

```
Add-VpnConnection -Name "PulseCxn1" -ServerAddress "10.17.1.216" -SplitTunneling -PluginApplicationID
"JuniperNetworks.JunosPulseVpn_cw5n1h2txyewy" -CustomConfiguration $sourceXml
```



Note: Some VPN Client cmdlet options are not applicable to creating Pulse connections. The following Add-VpnConnection options cause an error if you use them when creating a Pulse connection:

- AuthenticationMethod
- EncryptionLevel
- L2tpPsk
- MachineCertificateEKUFilter
- MachineCertificateIssuerFilter
- UseWinlogonCredential

Add Pulse connection that saves the user credentials after the first login and configures certificate authentication

The -RememberCredentials option applies to smart cards and certificate PINs, and to usernames and passwords.

```
$xml = "<pulse-schema ></pulse-schema>"
```

```
$sourceXml=New-Object System.Xml.XmlDocument
```

```
$sourceXml.LoadXml($xml)
```

```
Add-VpnConnection -Name "PulseCxn2" -ServerAddress "10.17.1.217" -RememberCredential -
PluginApplicationID "JuniperNetworks.JunosPulseVpn_cw5n1h2txyewy" -CustomConfiguration $sourceXml
```

Add Pulse connection that uses a specified role and realm and a non-standard port

This script's pulse-schema statement includes schema options that specify the realm and role that are used for this connection. If there are multiple realms or roles available to the user, and you do not specify the preferred values, then the user is prompted for selections.

```
$xml = "<pulse-
schema><port>4444</port><preferredRealm>Users</preferredRealm><preferredRole>TestRole</preferredRole><u
ri>/local</uri></pulse-schema>"
```

```
$sourceXml=New-Object System.Xml.XmlDocument
```

```
$sourceXml.LoadXml($xml)
```

```
Add-VpnConnection -Name "PulseCxn3" -ServerAddress "10.17.1.216" -SplitTunneling -RememberCredential -
PluginApplicationID "JuniperNetworks.JunosPulseVpn_cw5n1h2txyewy" -CustomConfiguration $sourceXml
```

Delete Pulse connection

To delete a Pulse connection, use the following command:

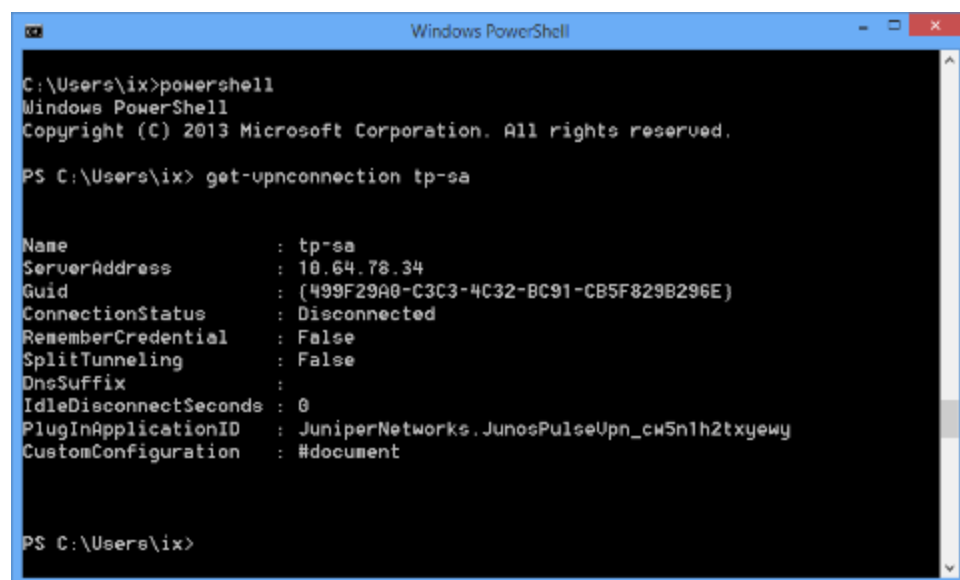
```
Remove -VpnConnection -Name <connection_name>
```

Get Pulse connection information

Figure 125 shows the output from the following command to show the properties of a VPN connection:

```
Get -VpnConnection -Name <connection_name>
```

Figure 125: Get-VpnConnection Output



Start Pulse connection on application launch

You can associate a Pulse connection with an application. When the user starts that application, the specified Pulse VPN connection is initiated.

```
Add -vpnConnectionTriggerApplication -ConnectionName "PulseCxn1" [-ApplicationID] <String[]>
```

Start Pulse connection when an application attempts to access a specified domain name

You can associate a Pulse connection with a specific DNS suffix to provide connectivity based on location awareness. When the user attempts to access a resource within the suffix, the specified Pulse VPN connection is initiated.

```
Add-VpnConnectionTriggerDnsConfiguration -ConnectionName "PulseCxn1" -DnsSuffix "pulsesecure.net" -DnsIPAddress "172.28.144.15","172.24.245.15" -PassThru
```

Use Microsoft Task Scheduler and PowerShell to update a Pulse connection

You can combine PowerShell and other Windows tools to provide flexibility for your VPN connections. The following example uses Windows Task Scheduler. The scheduled task behaves like a connection startup script. The command creates a scheduled task that runs the specified PowerShell script when the PulseTest connection has established a VPN connection.

```
schtasks /create /F /TN "VPN Connection Update" /TR "Powershell.exe -NonInteractive -command  
\test\create-conn.ps1 \test\scripts\settings.xml" /SC ONEVENT /EC Application /MO "[System[(Level=4 or  
Level=0) and (EventID=20225)]] and *[EventData[Data='PulseTest']]"
```

Table 27 lists options that you can use in your Pulse connection PowerShell scripts.

Table 27: Schema Options

Option	Description
"port"	Use a connection port other than the standard port, 443.
"uri"	Specifies a sign on policy path override to use when connecting to the server address.
"preferredRealm"	Specify the preferred connection realm. The user must be a member of the specified authentication realm.
"preferredRole"	Specify the preferred role. The user must be eligible for the role according to the role mapping rules in effect for the realm.
"optimizeForLowCostNetwork"	true/false Specifies that the connection uses a wired connection if one is available.
"isSingleSignOnCredential"	true/false Specifies that the credentials be used to access resources that require authentication after the tunnel is established.

Related Documentation

- [Microsoft Windows In-Box Pulse Secure Client Overview](#)
- [Microsoft Windows In-Box Pulse Secure Client User Interface](#)
- [Host Checker Statement of Health for Pulse Connect Secure Overview](#)
- [Configuring a Statement of Health Host Checker Policy for Pulse Connect Secure](#)
- [Enabling Statement of Health on the Windows Endpoint for the Windows In-Box Pulse Secure Client](#)
- [Viewing Windows In-Box Pulse Secure Client Log Messages](#)

Host Checker for Windows In-Box Pulse Secure Client Overview

Host Checker is a component of Pulse Secure client that reports the integrity of an endpoint that is attempting to connect to Pulse Connect Secure. Host Checker runs as a Trusted Network Connect (TNC) client on the endpoint. The client evaluates the endpoint according to predefined criteria and reports to the Trusted Network Connect server, which is a part of the Pulse

server. If the endpoint is not in compliance with the Host Checker policies, then the user might not get access to the network or might get limited access to the network depending upon the enforcement policies configured by the administrator.

For Windows In-box Pulse Secure clients, Host Checker can evaluate client compliance based on the following predefined criteria:

- **OS Check**—You can specify the version or minimal version that must be installed on the device.
- **Statement of Health**—Statement of Health (SoH) components evaluate an endpoint's state of health and make policy decisions for network access based on the result of the health check. To use SoH with the Windows in-box Pulse client, you must also enable the SoH functionality on the endpoint.

Related Documentation

- [Host Checker Statement of Health for Pulse Connect Secure Overview](#)
- [Configuring a Statement of Health Host Checker Policy for Pulse Connect Secure](#)
- [Enabling Statement of Health on the Windows Endpoint for the Windows In-Box Pulse Secure Client](#)
- [Configuring Host Checker for Windows In-Box Pulse Secure Client](#)
- [Implementing Host Checker Policies for Windows In-Box Pulse Secure Client](#)

Configuring Host Checker for Windows In-Box Pulse Secure Client

Host Checker policies can be part of a larger Host Checker configuration that applies to many different types of clients or to Windows in-box Pulse Secure clients only. However, you might find it easiest to create a separate Host Checker policy specifically for Windows in-box Pulse Secure clients.

To create a Host Checker policy for Windows Phone:

1. From the admin console, select **Authentication > Endpoint Security > Host Checker**.
2. In the Policies section, click **New** to open a New Host Checker Policy page.
3. Specify a name for the new policy and then click **Continue** to open the Host Checker Policy page.

The name appears in lists when you implement the policy so be sure to use a descriptive name, such as WinInBox HC Policy.

4. In the Rule Settings section, click **Select Rule Type** and select the following option, and then click **Add**:
 - **OS Checks**—To specify the version of Windows that must be installed on the device:
 1. Specify a descriptive name for this rule. For example, Must-Be-8.1-or-higher. Rule names cannot include spaces.
 2. Specify the criteria. For example, to enforce 8.1 or higher, create two conditions: Equal to 8.1 and Above 8.1.
 3. Click **Save Changes**.
5. You can also add a custom rule for Statement of Health to the Host Checker policy. (See the Related Documentation links.)
6. After you have configured all of your rules, specify how you want to enforce them by choosing one of the following options:
 - All of the rules
 - Any of the rules
 - Custom

For Custom requirements, you can specify a custom expression using Boolean operators AND and OR and also group and nest conditions using parenthesis.

7. Specify remediation options:
 - **Enable custom instructions**—If you enable this check box, a text box appears and allows you to type information that appears on the user's device if Host Checker discovers an issue.
 - **Send reason strings**—Select this option to display a message to users (called a reason string) that explains why the endpoint does not meet the Host Checker policy requirements.

8. When you are finished, click **Save Changes**.

Related Documentation

- [Host Checker Statement of Health for Pulse Connect Secure Overview](#)
- [Configuring a Statement of Health Host Checker Policy for Pulse Connect Secure](#)
- [Enabling Statement of Health on the Windows Endpoint for the Windows In-Box Pulse Secure Client](#)
- [Host Checker for Windows In-Box Pulse Secure Client Overview](#)

Implementing Host Checker Policies for Windows In-Box Pulse Secure Client

After you create one or more Host Checker policies, you must enable them. Pulse Connect Secure can use Host Checker policies at the realm or the role level.

Realm Authentication—You can configure a realm authentication policy to download and run Host Checker with a particular Host Checker policy. If the device does not meet the Host Checker requirements, then Pulse Connect Secure can deny access. You can provide remediation information in the Host Checker policy to describe the requirement and help users take steps to solve the issue.

To enable a Host Checker policy for a realm:

1. From the admin console, select **Users > User Realms > SelectRealm > Authentication Policy > Host Checker**. The Host Checker page displays all of the available Host Checker policies.
2. Select the check box next to each policy you want to include. Select one or both of the following check boxes next to the policy:
 - **Evaluate Policies**—Evaluates without enforcing the policy on the device and allows access.
 - **Require and Enforce**—Requires that the device be in compliance with the Host Checker policy. Selecting this option automatically enables the Evaluate Policies option.
3. Optionally select **Allow access to realm if any ONE of the selected “Require and Enforce” policies is passed**. This check box is available if you selected more than one Host Checker policy. If you enable this check box, a device is allowed access if it passes any of the Require and Enforce policies. The Cache Cleaner policy does not apply to Windows Phone devices.
4. Click **Save Changes**.

Role—You can configure a role to download and run Host Checker with a particular Host Checker policy. If the device does not meet the Host Checker requirements, then Connect Secure can deny access or assign the user to a remediation role that has limited access. You can provide remediation information in the Host Checker policy to help users take steps to solve the issue.

To enable a Host Checker policy for a role:

1. From the admin console, select **Users > User Roles > SelectRole > General > Restrictions > Host Checker**. The Host Checker page displays all of the available Host Checker policies.
2. Select **Allow users whose workstations meet the requirements specified by these Host Checker policies**.
3. In the Available Policies list, select the policies that you want to apply to select them, and then click Add to move them to the Selected Policies list. To select a policy click it. To select more than one policy, use Ctrl+click.
4. Optionally select **Allow access to realm if any ONE of the selected “Require and Enforce” policies is passed**. This check box is available if you selected more than one Host Checker policy. If you enable this check box, a device is allowed access if it passes any of the Require and Enforce policies.
5. Click **Save Changes**.

Related Documentation

- [Host Checker for Windows In-Box Pulse Secure Client Overview](#)
- [Configuring Host Checker for Windows In-Box Pulse Secure Client](#)

Host Checker Statement of Health for Pulse Connect Secure Overview

You can use the open standard Statement of Health (SoH) rule in a Host Checker policy for the Pulse for Windows client and for the Windows in-box Pulse client. SoH components evaluate an endpoint's state of health and make policy decisions for network access based on the result of the health check. To use SoH with the Windows in-box Pulse client, you must also enable the SoH functionality on the endpoint.



Note: The Statement of Health rule is the only Host Checker rule supported by the Windows in-box Pulse client, and the Statement of Health rule is available only with Connect Secure release 8.0 and later.

You can use the SoH health state validation to determine which roles or realms can be accessed by endpoints. If an endpoint fails the SoH check, or if the SoH cannot be negotiated successfully, the Host Checker policy fails.

You can check the following system health indicators:

- Antivirus is enabled.
- Antivirus is up to date.
- Antispyware is enabled.
- Antispyware is up to date.
- Firewall is enabled.
- Automatic updating is enabled

Related Documentation

- [Microsoft Windows In-Box Pulse Secure Client Overview](#)
- [Microsoft Windows In-Box Pulse Secure Client User Interface](#)
- [Windows PowerShell Script Examples for Microsoft Windows In-Box Pulse Secure Client](#)
- [Host Checker for Windows In-Box Pulse Secure Client Overview](#)
- [Enabling Statement of Health on the Windows Endpoint for the Windows In-Box Pulse Secure Client](#)
- [Viewing Windows In-Box Pulse Secure Client Log Messages](#)

Configuring a Statement of Health Host Checker Policy for Pulse Connect Secure

You can use the open standard Statement of Health rule in a Host Checker policy for both the Pulse for Windows client and the Windows in-box Pulse client.

To configure a Statement of Health rule in a Host Checker policy:

1. In the admin console, select **Authentication > Endpoint Security > Host Checker**.
2. In the Policies section, click **New** to create a new policy, or click an existing policy.
3. For a new policy, specify a name for the policy and then click **Continue**.
4. Click the **Windows** tab. Statement of Health is available for Windows endpoints only.
5. Under Rule Settings, select **Custom: Statement of Health**, and then click **Add**.
6. Type a Rule Name for this rule.

To configure the SoH rule, you must select one or more of the Statement of Health parameters.

1. Under Criteria, enter a **Label** for the selected SoH parameter, or accept the default.
2. Select an SoH policy option from the Parameter menu, and then click **Add** for the following types:
 - Antivirus Enabled
 - Antivirus up to date
 - Antispyware enabled
 - Antispyware up to date
 - Firewall Enabled
 - Automatic Updating Enabled

3. Select additional options from the Parameter list to add additional SoH parameters.
4. (Optional) For each rule, select the **Enable automatic remediation** check box. If you select this option for a rule, the user receives a remediation message from the SoH agent, and appropriate remediation is performed, if possible. If the box is not selected, the user receives a remediation message, but no remediation action is performed.



Note: Automatic remediation works for the Pulse for Windows client only. The Windows in-box Pulse client does not support automatic remediation.

5. Click **Save Changes**.

Related Documentation

- [Microsoft Windows In-Box Pulse Secure Client Overview](#)
- [Microsoft Windows In-Box Pulse Secure Client User Interface](#)
- [Windows PowerShell Script Examples for Microsoft Windows In-Box Pulse Secure Client](#)
- [Host Checker Statement of Health for Pulse Connect Secure Overview](#)
- [Enabling Statement of Health on the Windows Endpoint for the Windows In-Box Pulse Secure Client](#)
- [Viewing Windows In-Box Pulse Secure Client Log Messages](#)

Enabling Statement of Health on the Windows Endpoint for the Windows In-Box Pulse Secure Client

The Statement of Health (SoH) is a basic health check that can be performed on a Windows endpoint. SoH is a component of Network Access Protection (NAP), a Microsoft policy enforcement platform built into Windows 8, Windows 7, Windows Vista, and Windows Server 2008 operating systems that lets you protect network assets by enforcing compliance with system health requirements. NAP SoH functionality is enabled as part of the Pulse for Windows installation. However, before you can use the Statement of Health Host Checker rule with the Windows in-box Pulse client, you must enable the 3rd party Modem VPN NAP client on the endpoint and start the service.

To enable NAP SoH functionality, set the service to run automatically, and to enable the service using PowerShell:

1. Start Windows PowerShell and run the following commands :

```
netsh nap client set enforcement ID = 79624 admin = "enable"set-service napagent -startuptype automaticstart-service
-name napagent
```



Note: You must run the commands as an administrator.

Related Documentation

- [Microsoft Windows In-Box Pulse Secure Client Overview](#)
- [Microsoft Windows In-Box Pulse Secure Client User Interface](#)
- [Windows PowerShell Script Examples for Microsoft Windows In-Box Pulse Secure Client](#)
- [Host Checker Statement of Health for Pulse Connect Secure Overview](#)
- [Configuring a Statement of Health Host Checker Policy for Pulse Connect Secure](#)

Viewing Windows In-Box Pulse Secure Client Log Messages

The Windows in-box Pulse client uses the Windows Event Log for logging.

To view Windows in-box Pulse client log messages, perform the following steps:

1. Start the Windows Event Viewer.
2. On the navigation pane, open the following path:

Applications and Service Logs > Microsoft > Windows > VpnPluginPlatform > OperationalVerbose.

Related Documentation

- [Microsoft Windows In-Box Pulse Secure Client User Interface](#)
- [Windows PowerShell Script Examples for Microsoft Windows In-Box Pulse Secure Client](#)

PART 5 Index

- [Index](#)

Index

▪
 .pulsepreconfig
 example..... 30, 41, 42, 89, 178, 180, 181, 192, 193, 194,
 197, 198, 212, 270

3

3G Wireless226

7

7Zip 202, 203

8

802.1X connections
 Pulse - OAC comparison.....46
 Pulse Policy Secure144

A

Access Manager.....169
 ACE server 45, 70, 71, 109, 110, 122, 174
 ActiveSync
 using authorization-only policy226
 using Secure Mail235
 ActiveX
 Web installation183
Adapter type.....46
 Admin privileges178
 Agentless
 sign-in notifications92
Allow saving logon information
 Pulse Connect Secure151
 Pulse Policy Secure53
Allow user connections
 Pulse Connect Secure122
 Pulse Policy Secure53
Allow user to override connection policy
 connection option145
 SRX connection type.....146
 Android.....258
 API
 Pulse of Android247
 Apple Help

 customizing.....210
 Application Acceleration
 Mobile VPN connection242
Attempt most recently connected URL first
 SecurID48
 Authentication methods172
 Authorization-Only Policy for ActiveSync238
 Automatic updates79
 Automatic Upgrades164

B

BlackBerry223
 BOM201
BrandingCredProv.png209
BrandingLogo.png209
BrandPackager
 Supports201
 Workflow202
 Browser
 Launching the Pulse Secure Client for Android258
 Browser Session Cookie.....102
 Byte Order Mark201

C

Captive portal detection27
 Certificate
 Downloading.....86
 Importing.....84
 Certificate authentication
 Android.....240
 Pulse Connect Secure103
 Certificates
 Understanding82
 CHAP inner authentication217
 Client Error
 Messages32
 OS-X.....32
 Command line launcher219
 ConfigMgr162
 ConfigureInstaller213
 Connection is Established Options49
 Connection properties.....181
 Connection set.....183
 Connection Set Options

Adapter type	45
allow saving logon information	45
Allow user connections	45
Allow user to override connection policy	45
Cookie	102
Credential provider	
Policy Secure	64
Smartcard or Password	68
Credential Provider	
Connect Secure	106
Customer Support	
contacting PSGSC	10
Customizing Pulse Secure Client	200

D

Default installer	41
Deleting	
Pulse client log files	35
Deployment options	
Pulse Policy Secure and Pulse Connect Secure	41
DNS	52
DNS lookups	72, 156
DNS server search	220
Dynamic certificate trust	
Pulse Connect Secure	123
Pulse Policy Secure	45
Dynamic connections	
Pulse Connect Secure	123
Pulse Policy Secure	45
Dynamic VPN	
Configuration overview	170

E

EAP Methods	
EAP-GTC inner authentication	216
EAP-JUAC inner authentication	216
EAP-MD5 inner authentication	216
EAP-MSCHAPv2 inner authentication	216
EAP-PEAP outer authentication	216
EAP-TLS outer authentication	216
EAP-TTLS outer authentication	216
Error codes	
Installing	214
Error Messages	
OS X	32
ESAP	261
Event Log	
Windows client	33
Extend session	218

G

GINA	63
Graphics	209

H

Hard token	27
Heartbeat Interval	43
Heartbeat Timeout	43
Host Checker	
for Android clients	243
for Android clients, configuring	230, 244
for Android clients, implementing	245
for iOS clients, overview	229
for iOS clients, implementing	231
for Pulse Connect Secure, configuring	102
for Windows in-box clients, configuring	278
for Windows in-box clients, implementing	279
for Windows in-box clients, overview	277
for Windows Phone clients, configuring	265
for Windows Phone clients, implementing	266
for Windows Phone, overview	261
Host Checker policy	
Statement of Health	280
Host Enforcer	218
Hostname Resolution	39
Hotspot	36

I

Icon	
Pulse Client Status	31
IF-MAP	
Configuring for session migration	175
IMCs and IMVs	
Pulse Connect Secure	159
Pulse Policy Secure	74
Installing	
OS X	194
Windows	192
Instant Virtual System	89
IPv6	39

J

Jail Breaking Detection	
for iOS clients	230
jamCommand	
Options	198

K

Kerberos	121
----------------	-----

L

Layer 2	
Connect Secure	104
Learned User Settings	122
Limit to subnet	44
List of Connection URLs	48
Local Computer	
Pulse Connect Secure	109
Pulse Policy Secure	55
Localization	268
Location Awareness	
Configuring	71
Location Awareness Rules	149
Overview	24
Log Files	
iOS devices	233
Pulse Client, Deleting	35
Log viewer	218
Logging	
Windows Event Log	281

M

Machine authentication	
Pulse Connect Secure	103
Machine Authentication	
Pulse Policy Secure	70
Remote Desktop Protocol	63
Machine settings	179
Machine-only	
Authentication	63
Max. Session Length	101
MDM	
Pulse for Windows Phone	260
Message text	
Customizing the Pulse client	200
Microsoft System Center Configuration Manager ...	162
Microsoft Tech Net	268
MSCHAP inner authentication	217
MSCHAPv2 inner authentication	217

N

NAT-T	218
Navigation	209
NetBIOS name lookups	
SAM	120

Netmask	121
Network Access Protection	268, 281
Network Connect	38
Network Option	268

O

Odyssey Access Client	
Comparison with Pulse	216
Online Help	
Pulse Online Help	209
OPSWAT IMV	218
OS X	
Installing	194
Outer username	
Pulse Connect Secure	144
Pulse Policy Secure	46, 54

P

PAP inner authentication	217
Password	
saving web mail password for Android	243
saving web mail password for iOS	229
Windows in-box connection	272
Patch assessment, in Host Checker policy	
Pulse Connect Secure	162
Pulse Policy Secure	77
Patch management	27
PowerShell	275
Preconfigured installer	42
Preconfigured Pulse connections	213
Preferred Machine Realm	52
Preferred Machine Role Set	52
Preferred Password Logon Realm	52
Preferred Smartcard Logon Realm	52
Preferred User Role Set	53
Privileges	80
Proxy Automatic Configuration	226
<i>Proxy Server Settings</i>	
iOS Connection Profile	229
Pulse Collaboration Suite	
Overview	165
Pulse config file	
Example	213
Pulse connection	
Window Phone	263
Pulse Connection	
for Windows Phone	260
Pulse connection prompt	273
Pulse for Android	240
Split tunneling	241

Pulse for iOS	
Split tunneling	226
Pulse for Windows	
Split tunneling	261
Pulse Launcher	195
Pulse messages	32
Pulse user interface	223
pulselauncher	
command line launcher	195
Push Configuration.....	79

R

Radius server	44
Randomize URL list order	48
Realm	
Android devices.....	241
iOS device	227
Remote Desktop Protocol	
Pulse 802.1X connection.....	63
Remove connection	183
resource ID	206
Resource profile	
Secure mail	236
Riverbed 'Steelhead	
Android and iOS	242
Roaming.....	44
Roaming session	44
Roles	
Android.....	241
iOS.....	227
Pulse Connect Secure	88
Pulse Policy Secure.....	41
SAM.....	118
Rooting detection	
for Android clients.....	245
Route Monitor	105
Route monitoring	95
RSA SecurID	27
RSA token code	261
Rsource profiles	
SAM.....	119

S

S/MIME certificate	
Secure Mail	237
Scan list	
Pulse Connect Secure	144
Pulse Policy Secure	46
SCCM.....	77

Secure Mail	
Defining the resource profile	235
Enabling in user role	235
importing an S/MIME certificate	237
Secure Meeting	165
Secure Virtual Workspace	38
Security Assertion Markup Language	
and Brandpackager	205
Security Certificates	
with Pulse	26
Server certificate	
802.1X connection, Pulse Connect Secure	144
802.1X connection, Pulse Policy Secure.....	47
Server certificate is untrusted	
Windows in-box connection	269
Windows Phone	260
server ID	182
Session cookie	102
Session lifetime	43
Session migration	
and IF-MAP	173
and Timeout.....	173
Authentication server support	174
Load Balancers	174
Overview.....	172
Session time-outs.....	166
Shortcut key.....	206
Sign-in notifications	
Agentless	92
Pulse Client	92
Single Sign-on	205
Smart card	68
Smart card authentication.....	68
SMS	76
Soft token.....	27
Software package.....	81
Software requirements.....	77
Software upgrades	81
SoH	
in Host Checker Policy	280
Split tunneling	
iOS VPN.....	226
Overview.....	94
Split Tunneling Options.....	105
SRX Series gateways	
Mac Connection Requirements.....	169
SSL fallback.....	16
Statement of Health	268
Status Icons	
Pulse Client	31
Steelhead	

Android and OS	242
StringReference	203
Support	
Technical support	10
Supported features	
for Windows In-box	269
Windows Phone	261
Supported Platforms	
Windows In-Box	269
Windows Phone	260
Suspend, a connection	58
SVW	38
System Center Configuration Manager.....	27

T

Technical Support	10
Testing the Pulse Package	211
time-to-live	72
TKIP	47
Token code	264
Traffic enforcement	89
Type of sign-in info	
Windows in-box connection	272

U

unbrand	199
Upgrade	
Client Software.....	164
Upgrading	
OS X	212
Windows.....	212
User agent string	
Mobile	223
Windows Phone	263
Windows-In-box	224
User interface	
Pulse Client	205

User name

Windows in-box connection	272
User roles	
Configuration for Pulse Client	31
User-after-desktop	
Authentication	60

V

Validation	160
Virtual Hostname	
for Secure Mail.....	236
Virtual smart card	269
Virus signature monitoring.....	27
VPN on Demand	
iOS	225
VPN Provider	268

W

Web install	177
Webmail	
Android.....	243
iOS	229
WEP	56
Windows 7	
Pulse support FIPS.....	56
Windows Event Log	33
Windows Event Viewer	33
Windows Phone	260
Windows PowerShell	275
Windows RT	
Hostchecker	74
Windows Update	269
Wireless suppression	
Pulse Connect Secure	152
Pulse Policy Secure	46
WPA/WPA2	216
WSAM Destinations	121