Web Rewriting Configuration Guide

Published August 2020

Document Version 1.0

Pulse Secure, LLC
2700 Zanker Road,
Suite 200 San Jose
CA 95134

www.pulsesecure.net

Web Rewriting Configuration 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.

Contents

Web Rewriting

The Web rewriting feature enables you to intermediate Web URLs through the Content Intermediation Engine. You can intermediate URLs on the World Wide Web or on your corporate Intranet. Web rewriting also supports SNI TLS Extension.

When you intermediate standard Web content, you can create supplemental policies that "fine-tune" the access requirements and processing instructions for the intermediated content. You can create these supplemental policies through resourcre profiles (recommended) or resource policies.

Standard Web rewriting policy types include:

- **Web access control** - Web access policies control which Web resources users can access in order to connect to the Internet, intranet, or extranet.
- **Single sign-on** - Single sign-on policies enable you to automatically pass user credentials to a Web application.
- **Caching** - Caching policies control which Web content the system caches on a user's machine.
- **Java** - Java policies control to which servers and ports Java applets can connect. These policies also specify trusted servers for which the system resigns content.
- **Rewriting** - Rewriting policies specify resources that the system should not intermediate, minimally intermediation, or only intermediate selectively.
- **Web compression** - Web compression policies specify which types of Web data the system should and should not compress.
- **Web proxy** - (Resource policies only) Web proxy resource policies specify Web proxy servers for which the system should intermediate content. Note that the system intermediates both forward and backwards proxies, but only enables single sign-on to trusted proxies.
- **Launch JSAM** - (Resource policies only) Launch JSAM policies specify URLs for which the system automatically launches J-SAM on the client.
- **Protocol** - (Resource policies only) Protocol resource policies enable or disable HTTP 1.1 protocol support on the system.
- **Options** - (Resource policies only) You can enable IP based matching for hostnames as well as case-sensitive matching for path and query strings in Web resources through resource policy options.

Web rewriting is a standard feature on Connect Secure devices.

Remote SSO Overview

The Remote Single Sign-On (SSO) feature enables the admin to specify the URL sign-in page of an application to which you want the system to post a user's credentials, minimizing the need for users to re-enter their credentials when accessing multiple back-end applications. You may also specify additional forms values and custom headers (including cookies) to post to an application's sign-in form.

Remote SSO configuration consists of specifying Web resource policies:

- **Form POST policy** - This type of Remote SSO policy specifies the sign-in page URL of an application to which you want to post system data and the data to post. This data can include the user's primary or secondary username and password as well as system data stored by system variables. You can also specify whether or not users can modify this information.
- **Headers/Cookies policy** - This type of Remote SSO policy specifies resources, such as customized applications, to which you can send custom headers and cookies.

  If a user's system credentials differ from those required by the back-end application, the user can alternatively access the application:

  - **By signing in manually** - The user can quickly access the back-end application by entering his credentials manually into the application's sign-in page. The user may also permanently store his credentials and other required information in the system through the Preferences page as described below, but is not required to enter information in this page.
  - **Specifying the required credentials on Connect Secure** - The user must provide the system with his correct application credentials by setting them through the Preferences page. Once set, the user must sign out and sign back in to save his credentials. Then, the next time the user clicks the Remote SSO bookmark to sign in to the application, the system sends the updated credentials.

Note: Use the Remote SSO feature to pass data to applications with static POST actions in their HTML forms. It is not practical to use Remote SSO with applications that employ frequently changing URL POST actions, time-based expirations, or POST actions that are generated at the time the form is generated.

Passthrough Proxy Overview

The passthrough proxy feature enables the admin to specify Web applications for which the system performs minimal intermediation. Unlike traditional reverse proxy functionality, which also rewrites only selective parts of a server response but requires network changes as well as complex configuration, this feature only requires that you specify application servers and the way in which the system receives client requests to those application servers. Passthrough proxy also supports "SNI TLS Extension" on page 61:

- **Via a Connect Secure port** - When specifying an application for the passthrough proxy to intermediate, the admin specifies a port on which the system listens for client requests to the application server. When the system receives a client request for the application server, it forwards the request to the specified application server port. When you choose this option, you must open traffic to the specified system port on your corporate firewall.
- **Via virtual hostname** - When specifying an application for the passthrough proxy to intermediate, the admin specifies an alias for the application server hostname. You need to add an entry for this alias in your external DNS server that resolves to the system. When the system receives a client request for the alias, it forwards the request to the port you specify for the application server.

  This option is useful if your company has restrictive policies about opening firewall ports to either internal servers or servers in the DMZ. When using this option, we recommend that each hostname alias contains the same domain substring as your hostname and that you upload a wild card server certificate to the system in the format: *.domain.com.

For example, if your system is iveserver.yourcompany.com, then a hostname alias should be in the format appserver.yourcompany.com and the wild card certificate format would be *.yourcompany.com. If you do not use a wild card certificate, then a client's browser issues a certificate name check warning when a user browses to an application server, because the application server hostname alias does not match the certificate domain name. However, this behavior does not prevent a user from accessing the application server.

Note: When you configure passthrough proxy to work in virtual hostname mode, users must use the hostname that you specify through the System > Network > Overview page of the admin console when signing into the device. They cannot access the use passthrough proxy feature if they sign into the device using its IP address.

Just as with the Content Intermediation Engine, the passthrough proxy option offers increased security relative to the Secure Application Manager, because when enabled for an application, the system allows the client to send only Layer 7 traffic directed to fixed application ports to the enterprise network. Use this option to enable the system to support applications with components that are incompatible with the Content Intermediation Engine, such as Java applets in Oracle e-business suite applications or applets that run in an unsupported Java Virtual Machine (JVM).

Note the following:

- Passthrough proxy URLs must be hostnames. Paths of hostnames are not supported.
- Pulse Secure strongly recommends that you not mix passthrough proxy Port mode and passthrough proxy Host mode.
- The passthrough proxy option works only for applications that listen on fixed ports and where the client does not make direct socket connections.
- To use passthrough proxy with Oracle E-Business applications, you must install a real certificate on the system and you must configure Oracle Forms to use the Forms Listener Servlet mode.
- The following advanced features of the framed toolbar are not available in passthrough proxy: bookmark current page, display the original URL, display the favorite bookmarks.

Task Summary: Configuring the Web Rewriting Feature

Note: When intermediating content through the content intermediation engine, it is recommended that the GMT time on both Pulse Connect Secure and the backend Web application server be the same. This prevents any premature expiration of cookies if the Connect Secure system time is later than the Web application server time.

To configure the Web rewriting feature:

1. Create resource profiles that enable access to web sites, create supporting autopolicies (such as single sign-on and Java access control policies) as necessary, include bookmarks that link to the web sites, and assign the policies and bookmarks to user roles using settings in the Web Applications Resource Profiles page (Users > Resource Profiles > Web) of the admin console.

   We recommend that the admin use resource profiles to configure Web rewriting (as described above). However, if the admin does not want to use resource profiles, the admin can configure Web rewriting using role and resource policy settings in the following pages of the admin console instead:

a. Create resource policies that enable access to web sites using settings in the Users > Resource Policies> Web > Web ACL page of the admin console.

b. As necessary, create supporting resource policies (such as single sign-on and Java access control policies) using settings in the Users > Resource Policies> Select Policy Type pages of the admin console.

c. Determine which user roles may access the web sites that you want to intermediate, and then enable Web access for those roles through the Users > User Roles > Select Role > General > Overview page of the admin console.

d. Create bookmarks to your web sites using settings in the Users > User Roles > Select Role > Web > Bookmarks page of the admin console.

e. As necessary, enable Web general options that correspond to the types of Web content you are intermediating (such as Java) using settings in the Users > User Roles > Select Role > Web > Options page of the admin console.

2. After enabling access to Web applications or sites using Web rewriting resource profiles or roles and resource policies, you can modify general role and resource options in the following pages of the admin console:

a. (Optional) Set additional Web browsing options (such as allowing users to create their own bookmarks or enabling hostname masking) Users > User Roles > Select Role > Web > Options page of the admin console.

Note: Even if you enable hostname masking, links corresponding to protocols not rewritten by Web rewriting are not obfuscated. For example, ftp://xyz.pulsesecure.net and file://fileshare.pulsesecure.net/filename are not obfuscated. By not obfuscating the hostname, users can still access these resources.

b. (Optional) Set additional Web options for individual resources (such as enabling Web rewriting to match IP addresses to hostnames) using settings in the Users > Resource Policies> Web > Options page of the admin console.

Note: Certain Web rewriting features (such as passthrough proxy and SSO to NTLM resources) require additional configuration. For more information, see the appropriate configuration instructions.

Note: If rewriter or passthrough proxy initiates the SSL handshake to the IP instead of hostname of the backend server, then the SNI extension cannot be added to the handshake.

Creating a Custom Web Application Resource Profile

A custom Web application resource profile is a resource profile that controls access to a Web application, Web server, or HTML page.

To create a custom Web application resource profile:

1. In the admin console, select **Users > Resource Profiles > Web.**
2. Click **New Profile.**
3. From the **Type** list, choose **Custom**.
4. Enter a unique name and optionally a description for the resource profile.

5. In the **Base URL** field, enter the URL of the Web application or page for which you want to control access using the format: [protocol://]host[:port][/path]. (The system uses the specified URL to define the default bookmark for the resource profile.)
6. In the Autopolicy: Web Access Control section, create a policy that allows or denies users access to the resource specified in the Base URL field. (By default, the system automatically creates a policy for you that enables access to the Web resource and all of its sub-directories.)
7. (Optional) Click **Show ALL autopolicy types** to create additional autopolicies that fine-tune access to the resource. Then, create the autopolicies using instructions in the following sections:
8. Click **Save** and **Continue**.
9. In the **Roles** tab, select the roles to which the resource profile applies and click **Add**.

   The selected roles inherit the autopolicies and bookmarks created by the resource profile. If it is not already enabled, the system also automatically enables the Web option in the Users > User Roles > Select Role > General > Overview page of the admin console for all of the roles you select.
10. Click **Save Changes**.
11. (Optional) In the **Bookmarks** tab, modify the default bookmark created by the system and/or create new ones. (By default, the system creates a bookmark to the base URL defined in the **Base URL** field and displays it to all users assigned to the role specified in the **Roles** tab.)

Defining Base URLs

When creating a Web resource profile, you must use the following format when defining base URLs:

[protocol://]host[:port][/path]

Within this format, the components are:

- **Protocol** (required) - Possible values: http:// and https://. Note that you cannot use special characters within the protocol.
- **Host** (required) - Possible values:
  - **DNS Hostname** - For example: www.pulsesecure.net
  - **IP address** - You must enter the IP address in the format: a.b.c.d.

  For example: IPv4 format: 10.11.149.2. IPv6 format: [2001:db8:a0b:12f0::1/64]:80,443/public/*

  [2001:db8:a0b:12f0::1/64]:8000-9000/*. You cannot use special characters in the IP address.
- **Ports** (optional) - You must use the delimiter ":" when specifying a port. For example: 10.11.149.2/255.255.255.0:*
- **Path** (optional) - When specifying a path for a base URL, the system does not allow special characters. If you specify a path, you must use the "/" delimiter. For example**, http://www.pulsesecure.net/sales.**

Defining Web Resources

When creating a Web resource profile, you must use the following format when defining resources for autopolicies:

[protocol://]host[:ports][/path]

Within this format, the four components are:

- **Protocol** (required) - possible values: http:// and https://. Note that you cannot use special characters within the protocol.
- **Host** (required) - possible values:
  - **DNS Hostname** - For example: www.pulsesecure.net

Table 1 lists the special characters allowed in the hostname.

Table 1 DNS Hostname Special Characters

*	Matches ALL characters.
%	Matches any character except dot (.)
?	Matches exactly one character

  - **IP address/Netmask** - You must enter the IP address in the format: a.b.c.d

    You may use one of two formats for the netmask:

    - **Prefix**: High order bits
    - **IP**: a.b.c.d

    For example: IPv4 format: 10.11.149.2. IPv6 format: [2001:db8:a0b:12f0::1/64]:80,443/public/*

    [2001:db8:a0b:12f0::1/64]:8000-9000/*. You cannot use special characters in the IP address. You cannot use special characters in the IP address or netmask.
- **Ports** (optional) - You must use the delimiter ":" when specifying a port. For example: 10.11.149.2/255.255.255.0:*

Table 2lists the possible port values.

Table 2 Possible Port Values

*	Matches ALL ports; you cannot use any other special characters
port[,port]*	A comma-delimited list of single ports. Valid port numbers are [1-65535].
[port1]-[port2]	A range of ports, from port1 to port2, inclusive.

Note: You can mix port lists and port ranges, such as: 80,443,8080-8090

If the port is missing, then the default port 80 is assigned for http, 443 for https.

- **Path** (optional) - When specifying a path for a Web access control autopolicy, you may use a * character, meaning ALL paths match. (The system does not support any other special characters.) If you specify a path, you must use the "/" delimiter. For example:
  - http://www.pulsesecure.net/sales
  - http://www.pulsesecure.net:80/*

- https://www.pulsesecure.net:443/intranet/*

Defining a Web Access Control Autopolicy

Web access policies control which Web resources users can access in order to connect to the Internet, intranet, or extranet. When defining a custom Web resource profile, you must enable a corresponding Web access control autopolicy that enables access to the profile's primary resource. The system simplifies the process for you by automatically creating an autopolicy that allows access to the Web resource and all of its sub-directories.

If necessary, you may choose to modify this default autopolicy or create supplementary Web access control autopolicies that control access to additional resources. For instance, your IT department may use one server to store Web pages for your company intranet (http://intranetserver.com) and another server to store the images that the Web pages reference (http://imagesserver.com). In this case, you can create two Web access control autopolicies that enable access to both servers so that your users can access both your Web pages and the corresponding images.

To create a new Web access control autopolicy:

1. Create a custom Web application resource profile.
2. If available, click the **Show ALL autopolicy types** button to display the autopolicy configuration options.
3. If it is not already enabled, select the **Autopolicy: Web Access Control** check box.
4. In the **Resource** field, specify the Web server or HTML page to which you want to control access using the format: [protocol://]host[:ports][/path].
5. From the **Action** list, choose **Allow** to enable access to the specified resource or **Deny** to block access to the specified resource.
6. Click **Add**.
7. Click **Save Changes.**

Defining a Single Sign-On Autopolicy

Single sign-on policies enable you to automatically pass user credentials to the Web application specified in your policy. Single sign-on autopolicies also intermediate the data that you pass.

To create a single sign-on (SSO) autopolicy:

1. Create a Web resource profile.
2. If available, click the **Show ALL autopolicy types** button to display the autopolicy configuration options.
3. Select the **Autopolicy: Single Sign-On** check box.
4. Select a single sign-on method and configure the corresponding **SSO** options:

Note: SSO options require you to select credentials. If you have not already done so, define the credentials using the Resource Policies > Web > General page prior to defining your SSO autopolicy.

- **Disable SSO** - Disables single sign-on.
- **Basic Auth** - Enables the system to intermediate the challenge/response sequence during basic authentication and use the credentials it collects to sign into a protected resource within the same Intranet zone. This option does not apply to Citrix resource profiles.
- **NTLM -** Enables the system to intermediate the challenge/response sequence during NTLM authentication and use the credentials it collects to sign into a protected resource within the same Intranet zone. This option does not apply to Citrix resource profiles.

Note: Web rewriting and file browsing both support **NTLM v1 and NTLM v2.**

- **Kerberos** - Enables the system to intermediate the challenge/response sequence during Kerberos authentication and use the credentials it collects to sign into a protected resource within the same Intranet zone.
- **Constrained Delegation** - Enables authentication of users by Kerberos after their identity has been verified using a non-Kerberos authentication method. For example, suppose a user authenticates with RADIUS and enters their passcode (typically PIN and tokencode). When accessing a service, the user may be challenged again because the PIN is not recognized. With constrained delegation, the administrator sets up passwords for constrained delegation users. The users do not need to know this password. When accessing the same HTTP service, the system now fetches the ticket on behalf of the user without challenging the user.
- **Remote SSO** - Enables the system to post the data that you specify (including usernames, passwords, and system data stored by variables) to Web applications. This option also enables you specify custom headers and cookies to post to Web applications.

5. Click **Save Changes**.

Specifying Basic Authentication, NTLM or Kerberos SSO Autopolicy Options

To configure basic authentication, NTLM or Kerberos SSO autopolicy options:

1. Create an SSO autopolicy and choose Basic Auth, NTLM or Kerberos.
2. In the **Resource** field, specify the resources to which this policy applies.

   When entering a resource in this field, note that:

   - If you want to automatically post values to a specific URL when an end-user clicks on a system bookmark, the resource that you enter here must exactly match the URL that you specify in the Base URL field of the resource profile.
   - If you want to automatically submit user credentials to other web sites within the same Intranet zone, the hostname that you enter here must end in the DNS suffix configured in the System > Network > Overview page of the admin console.

3. Select the credentials to use. If this pull-down menu is blank, no credentials are defined in the **SSO General** tab.
4. (NTLM only) Select the Fallback to **NTLM V1** option to fallback to NTLM V1 if **NTLM V2** fails. If you do not select this option, the system falls back only to **NTLM V2**. An intermediation page appears if SSO fails.

5. (Kerberos only) Select the Fallback to **NTLM V2 only** option to fallback only to NTLM V2 if kerberos fails. If you do not select this option, a Kerberos intermediation page appears if Kerberos SSO fails.
6. (Constrained delegation only) Select the Fallback to **Kerberos** option fallback to Kerberos if constrained delegation fails. If you do not select this option, an error page appears if SSO fails.

Specifying Remote SSO Autopolicy Options

To configure remote SSO autopolicy options:

1. Create an SSO autopolicy through a custom Web resource profile and choose Remote SSO.
2. If you want to perform a form POST when a user makes a request to the resource specified in the Resource field, select the POST the following data check box. Then:
    1. In the **Resource** field, specify the application's sign-in page, such as: http://my.domain.com/public/login.cgi. Widlcard characters are not supported in this field.

        If you want to automatically post values to a specific URL when an end user clicks on a system bookmark, the resource that you enter here must exactly match the URL that you specify in the Base URL or Web Interface (NFuse) URL field of the resource profile.
    2. In the **Post URL** field, specify the absolute URL where the application posts the user's credentials, such as: http://yourcompany.com/login.cgi. You can determine the appropriate URL using a TCP dump or by viewing the application's sign-in page source and searching for the POST parameter in the FORM tag.
    3. Optionally specify the user data you want to post and user modification permissions.
    4. To specify user data to post, enter data in the following fields and click Add:
        - **Name** - The name to identify the data of the **Value** field. (The back-end application should expect this name.)
        - **Value** - The value to post to the form for the specified Name. You can enter static data, a system variable, or system session variables containing username and password values.
        - **User modifiable?** setting - Set to **Not modifiable** if you do not want the user to be able to change the information in the Value field. Set to User **CAN** change value if you want the user to have the option of specifying data for a back-end application. Set to User **MUST** change value if users must enter additional data in order to access a back-end application. If you choose either of the latter settings, a field for data entry appears on the user's Advanced Preferences page. This field is labeled using the data you enter in the User label field. If you enter a value in the Value field, this data appears in the field but is editable.
    5. Select the **Deny direct login for this resource** check box if you do not want to allow users to manually enter their credentials in a sign-in page. (Users may see a sign-in page if the form POST fails.)
    6. Select the **Allow multiple POSTs to this resource** check box if you want to send POST and cookie values to the resource multiple times if required. If you do not select this option, the system does not attempt single sign-on when a user requests the same resource more than once during the same session.

3. If you want to post header data to the specified URL when a user makes a request to a resource specified in the Resource field, select the **Send the following data as request headers** check box. Then:
    1. In the Resource section, specify the resources to which this policy applies.
    2. Optionally specify the header data to post by entering data in the following fields and clicking Add:
        - **Header name** - The text to send as header data.
        - **Value** - The value for the specified header.
4. Click **Save Changes**.

Defining a Caching Autopolicy

Caching policies control which Web content the system caches on a user's machine.

To create a Web caching autopolicy:

1. Create a custom Web application resource profile.
2. If available, click the **Show ALL autopolicy types** to display the autopolicy configuration options.
3. Select the **Autopolicy: Caching** check box.
4. In the **Resource** field, specify the resources to which this policy applies.
5. In the **Action** field, select one of the following options:
    - **Smart** - Select this option to send a cache-control:no-store header or a cache-control:no-cache header based on the user's Web browser and content type.

      When you select this option, the system makes media files and zip files work properly by removing their origin server's cache-control headers. For example, the following logic searches for "msie" or "windows-media-player" in user-agent headers in order to remove cache or cache-control:no-store response headers and make the files cacheable:

      ```
      (if content type has "audio/x-pn-realaudio" OR
       if content type begins with "video/" OR
       if content type begins with "audio/" OR
       if content type is "application/octet-stream" and the file extension
      begins with "rm" or "ram"
      )
      ```

      If the system finds "msie" or "windows-media-player" in the user-agent header and any of the following apply:
        - Request is for Flash, .xls, .pps, .ppt files
        - Content-type is application/, text/rtf, text/xml, model/
        - Origin server sends a content-disposition header

then the system sends the cache-control:no-store header and removes the origin server's cache-control header.

In all other cases, the system adds the pragma:no-cache or cache-control:no-store response headers.

Note: Citrix .ica and QuickPlace files get some special treatment. Citrix .ica files get cache-control:private only when smart caching is enabled. QuickPlace files that do not match a specified rule files (which takes precedence) get CCNS and cache-control:private.

Also note that if you select this option, enable GZIP compression, and try to access a text file attachment using Domino Web Access 6.5 through Internet Explorer, you cannot open the attachment. To enable text attachments, you must either install the Internet Explorer 323308 patch or enable the No Store option.

- **No-Store -** Select this option to deliver attachments to Internet Explorer without saving them to the disk. (The browser temporarily writes files to the disk, but immediately removes them once it has opened the file in the browser.) When you select this option, the system removes the origin server's cache-control header and adds a cache-control:no-store response header if the user-agent string sent by the browser contains "msie" or "windows-media-player."

  This option might slow browsing by causing repeated content fetches, which can cause performance issues on very slow connections.
- **No-Cache -** Select this option to prevent the user's browser from caching files to the disk. When you select this option, the system adds the standard HTTP pragma:no-cache header and cache-control:no-cache (CCNC) header (HTTP 1.1) to response files. Also, the system does not forward the origin server's caching headers, such as age, date, etag, last-modified, expires.

  When no-cache headers are present on certain types of attachments (PDF, PPT, streaming files), Internet Explorer does not properly render the documents because the rendering process requires the browser to temporarily writes these files to cache.
- **Unchanged -** Select this option to forward the origin server's caching headers as is.

  When using Citrix published applications through the Web interface, the Web interface server may send a Cache-Control:no-cache in the response header of the .ica file. Because the caching header is not removed when using the Unchanged setting, .ica files are not downloaded to the client PC. To resolve this, use the Smart caching option.

6. Click **Add**.
7. Click **Save Changes.**

Defining a Java Access Control Autopolicy

A Java access control autopolicy defines the list of servers and ports to which Java applets can connect. This autopolicy also specifies which resources the system signs using the code-signing certificate that you upload.

When you enable Java access control using this autopolicy, the system automatically enables the Allow Java applets option on the Users > User Roles > Select Role > Web > Options page of the admin console.

To create a Java access control autopolicy:

1. Create a custom Web application resource profile.
2. Click **Show ALL autopolicy types.**
3. Select the **Autopolicy: Java Access Control** check box.
4. In the **Resource** field, specify the server resources to which this policy applies using the format: host:[ports]. (By default, the system populates this field with the server specified in your resource profile's base URL.)
5. Select one of the following options from the Action list:
   - **Allow socket access** - To enable Java applets to connect to the servers (and optionally ports) in the Resource list.
   - **Deny socket access** - To prevent Java applets from connecting to the servers (and optionally ports) in the Resource list.
6. Click **Add**.
7. Select the **Sign applets with code-signing certificate** check box to resign the specified resources using the certificate uploaded through the System > Configuration > Certificates > Code-signing Certificates page of the admin console. (The system uses the imported certificate to sign the server resources that you specify in the **Resources** field.)
8. Click **Save Changes**.

Defining a Server to Which Java Applets Can Connect

When defining servers to which Java applets can connect, you must use the following format:

host[:ports]

Within this format, the two components are:

- **Host** (required) - Possible values:
  - **DNS Hostname** - For example: www.pulsesecure.net

You may use the following special characters allowed in the hostname:

*	Matches ALL characters.
%	Matches any character except dot (.)
?	Matches exactly one character

- **IP address/Netmask** - You must enter the IP address in the format: a.b.c.d.

  You may use one of two formats for the netmask:

  Prefix: High order bits

  IP: a.b.c.d

  For example: 10.11.149.2/24 or 10.11.149.2/255.255.255.0 You cannot use special characters in the IP address or netmask.

- **Ports** - You must use the delimiter ":" when specifying a port. For example: 10.11.149.2/255.255.255.0:* Table 3 lists the possible port values.

Table 3 Possible Port Values

*	Matches ALL ports; you cannot use any other special characters
port[,port]*	A comma-delimited list of single ports. Valid port numbers are [1-65535].
[port1]-[port2]	A range of ports, from port1 to port2, inclusive.

Note: You can mix port lists and port ranges, such as: 80,443,8080-8090.

Defining a Rewriting Autopolicy

By default, the system intermediates all user requests to Web hosts-unless you have configured it to serve requests to certain hosts using a different mechanism, such as the Secure Application Manager. Rewriting autopolicies enable you to "fine-tune" the default options by changing which mechanisms to rewrite Web data and defining resources that you want to minimally rewrite or not rewrite at all.

To create a rewriting autopolicy:

1. Create a custom Web application resource profile.
2. Click **Show ALL autopolicy types.**
3. Select the **Autopolicy: Rewriting Options** check box.
4. Select one of the following options:

- **Passthrough Proxy** - Select this option to specify Web applications for which the Content Intermediation Engine performs minimal intermediation.
- **No rewriting (use WSAM)** - Select this option to intermediate content using PSAM instead of the Content Intermediation Engine. Then, specify the application server for which you want to intermediate content. (At minimum, you need to click Add in order to intermediate content to and from the server that the system extracts from the Web access control policy).
- **No rewriting (use JSAM)** - Select this option to intermediate content using JSAM instead of the Content Intermediation Engine. Then, specify the application server for which you want to intermediate content. (At minimum, you need to click Add in order to intermediate content to and from the server that the system extracts from the Web access control policy).
- **No rewriting** - Select this option to automatically create a selective rewriting policy for the autopolicy's URL, thereby configuring the system to not intermediate any content to and from the resource. For example, you may choose this option if you do not want the system to intermediate traffic from web sites that reside outside of the corporate network, such as yahoo.com. If you select this option, you do not have to configure any additional rewriting settings.

Specifying Passthrough Proxy Autopolicy Options

To configure passthrough proxy autopolicy options:

1. Create a rewriting autopolicy and select **Passthrough Proxy**.
2. Choose the way in which you want to enable the passthrough proxy feature:

- **Use virtual hostname** - If you choose this option, specify a hostname alias for the application server. When the system receives a client request for the application server hostname alias, it forwards the request to the specified application server port in the Base URL field.
- **Use IVE port** - If you choose this option, specify a unique port in the range 11000-11099. The system listens for client requests to the application server on the specified port and forwards any requests to the application server port specified in the Base URL field.

  The corresponding URL for the resource profile must specify the application server hostname and the port used to access the application internally. You cannot enter a path for the base URL.

  In order to make Sharepoint work successfully through the system, you must select the Override automatic cookie handling check box in Internet Explorer under Tools Internet options > Privacy > Advanced Privacy Settings if the following conditions true:

  - You select the U**se virtual hostname** option during Pass Through Proxy configuration.
  - The virtual hostname that you specify in your Sharepoint configuration is different from the hostname that you configure through the system setup (that is, if the domains are different).
  - You enable persistent cookies through the Users > User Roles > Select Role > General > Session Options page of the admin console.

3. Select the **Rewrite XML** check box if you want to rewrite URLs contained within XML content. If this option is disabled, the system passes the XML content "as is" to the server.
4. Select the **Rewrite external links** check box if you want to rewrite all the URLs presented to the proxy. If this option is disabled, the system rewrites only those URLs where the hostname is configured as part of the passthrough proxy policy.
5. Select the **Block cookies from being sent to the browser** check box if you want to block cookies destined for the client's browser. The system stores the cookies locally and sends them to applications whenever they are requested.
6. Select the **Host-Header forwarding** check box if you want to pass the hostname as part of the host header instead of the actual host identifier.

   The Host-Header forwarding option is only valid in passthrough proxy Virtual hostname mode.
7. Click **Save Changes**.
8. If you select:

   - Use virtual hostname, you must also:
     - Add an entry for each application server hostname alias in your external DNS that resolves to the system.
     - Upload a wildcard server certificate to the system (recommended).
     - Define the system name and hostname in the Network Identity section of the System > Network > Internal Port tab.

- To use the system port, you must also open traffic to port you specified for the application server in your corporate firewall.

If your application listens on multiple ports, configure each application port as a separate passthrough proxy entry with a separate port. If you intend to access the server using different hostnames or IP addresses, configure each of those options separately; in this case, you can use the same port.

Specifying PSAM Rewriting Autopolicy Options

To configure PSAM rewriting autopolicy options:

1. Create a rewriting autopolicy and select **No rewriting (use WSAM)**.
2. In the **Destination** field, specify resources for which PSAM secures client/server traffic between the client and the system. By default, the system extracts the correct server from the Web access control policy. You may choose to use this server as-is, modify it, and/or add new servers to the list.

   When specifying a server, specify the hostname (the wild cards '*' or '?' are accepted) or an IP/netmask pair. Specify multiple ports for a host as separate entries.
3. Click **Add**.
4. Click **Save Changes**.

   When you intermediate through PSAM using this autopolicy, the system automatically enables the Secure Application Manager option on the Users > User Roles > Select Role > General > Overview page of the admin console.

Specifying JSAM Rewriting Autopolicy Options

To configure JSAM rewriting autopolicy options:

1. Create a rewriting autopolicy and select **No rewriting (use JSAM).**
2. In the **Server Name** field, enter the **DNS name** of the application server or the server IP address.
3. In the Server Port field, enter the port on which the remote server listens for client connections.

   For example, to forward Telnet traffic from a remote machine, specify port 23 for both the client port (on which JSAM listens) and the server port (on which the Telnet server listens).

Note: To enable drive mapping to this resource, enter 139 as the server port.

4. In the **Client Loopback IP** field, provide a static loopback address. If you do not provide a static IP loopback address, the system assigns an IP loopback address dynamically.
5. In the **Client Port** field, enter the port on which JSAM should listen for client application connections.

   Typically, the local port value is the same value as the server port; the local port value usually only differs for Linux or Macintosh users who want to add applications for port forwarding that use ports under 1024.

Note: To enable drive mapping to this resource, enter 139 as the server port.

You may configure more than one application on a single port, such as app1.mycompany.com, app2.mycompany.com, app3.mycompany.com. Either you assign a static loopback address or the system assigns a dynamic loopback address (127.0.1.10, 127.0.1.11, 127.0.1.12) to each application. JSAM then listens on these multiple loopback addresses on the specified port. For example, when there is traffic on 127.0.1.12 on the specified port, the system forwards the traffic to the app3.mycompany.com destination host.

6. Select **Launch JSAM** to automatically start JSAM when the system encounters the Base URL.
7. Click **Add**.
8. Click **Save Application** or **Save + New**.

Defining a Web Compression Autopolicy

Web compression autopolicies specify which types of Web data the system should and should not compress. For example, since javascript does not work when compressed, you might use this feature to specify that the system should not compress javascript data going to and from an e-mail server by entering the following resource: http://owa. pulsesecure.net.net/*.js.

Note: In order to properly compress data, you must enable compression at the system level as well as creating compression autopolicies. To enable compression, use settings in the Maintenance > System > Options page of the admin console.

To create a Web compression autopolicy:

1. Create a custom Web application resource profile.
2. If available, click the **Show ALL autopolicy types** button to display the autopolicy configuration options.
3. Select the **Autopolicy: Web compression** check box.
4. In the **Resource** field, specify the resources to which this policy applies.
5. Select one of the following options from the Action list:
   - **Compress** - Compress the supported content types from the specified resource.
   - **Do not compress** - Do not compress the supported content types from the specified resource.
6. Click **Add**.
7. Click **Save Changes.**

Defining Web Resource Profile Bookmarks

When you create a Web resource profile, the system automatically creates a bookmark that links to the primary URL or domain that you specified in the resource profile. The system enables you to modify this bookmark as well as create additional bookmarks within the same domain.

For example, you may create a resource profile that controls access to your company intranet. Within the profile, you may specify:

- **Resource profile name**: Your Intranet
- **Primary resource**: http://intranet.com
- **Web access control autopolicy**: Allow access to http://intranet.com:80/*
- **Roles**: Sales, Engineering

When you create this policy, the system automatically creates a bookmark called "Your Intranet" enabling access to http://intranet.com and displays the bookmark to members of the Sales and Engineering roles.

You may then choose to create the following additional bookmarks to associate with the resource profile:

- **"Sales Intranet" bookmark**: Creates a link to the http://intranet.com/sales page and displays the link to members of the Sales role.
- **"Engineering Intranet" bookmark**: Creates a link to the http://intranet.com/engineering page and displays the link to members of the Engineering role.

When configuring bookmarks, note that:

- You can only assign bookmarks to roles that you have already associated with the resource profile-not all of the roles defined on the system. To change the list of roles associated with the resource profile, use settings in its Roles tab.
- Bookmarks simply control which links to display to users-not which resources the users can access. For instance, in the example used above, a member of the Sales role would not see a link to the Engineering Intranet page, but he could access it by entering http://intranet.com/engineering his Web browser's address bar.
- You cannot create bookmarks that link to additional URLs and domains defined through Web access control autopolicies.

You can use two different methods to create Web bookmarks:

- **Create bookmarks through existing resource profiles** (recommended) - When you select this method, the system automatically populates the bookmark with key parameters (such as the Web interface (NFuse) URL) using settings from the resource profile. Additionally, while you are creating the associated resource profile, the system guides you through the process of creating any required policies to enable access to the bookmark.
- **Create standard bookmarks** - When you select this option, you must manually enter all bookmark parameters during configuration. Additionally, you must enable access to the Web feature and create resource policies that enable access to the web sites defined in the bookmark.

Creating Bookmarks Through Existing Resource Profiles

To configure Web resource profile bookmarks:

1. If you want to create a resource profile bookmark through the standard resource profiles page:
    1. In the admin console, select **Users > Resource Profiles > Web > Resource Profile Name > Bookmarks**.
    2. Click the appropriate link in the **Bookmark** column if you want to modify an existing bookmark. Or, click **New Bookmark** to create an additional bookmark.

    Alternatively, if you want to create a resource profile bookmark through the user roles page:

2. In the admin console, select **Users > User Roles > Role Name > Web > Bookmarks.**
3. Click **New Bookmark**.
4. From the **Type** list, choose **Pick a Web Resource Profile.** (The system does not display this option if you have not already created a Web resource profile.)
5. Select an existing resource profile.
6. Click **OK**. (If you have not already associated the selected role with the resource profile, the system automatically makes the association for you. The system also enables any access control policies for the role that are required by the resource profile.)
7. If this role is not already associated with the selected resource profile, the system displays an informational message. If you see this message, click **Save Changes** to add this role to the resource profile's list of roles and to update the profile's autopolicies as required. Then, repeat the previous steps to create the bookmark.

   When you create a resource profile bookmark through the user roles page (instead of the standard resource profiles page), the system only associates the generated bookmark with the selected role. The system does not assign the bookmark to all of the roles associated with the selected resource profile.
8. Optionally change the name and description of the bookmark. (By default, the system populates names the bookmark using the resource profile name.)
9. In the **URL** field, add a suffix to the URL if you want to create links to sub-sections of the domain defined in the primary resource profile.

   Make sure to enter a unique URL in this field. If you create two bookmarks with the same URL, the system deletes one of the bookmarks from the end-user view. You will still be able to see both bookmarks, however, in the administrator console.
10. Under Options, select the **Bookmark opens in new window** check box if want to enable the system to automatically open the Web resource in a new browser window. Next, select:
    - **Do not display browser address bar** - Select this option to remove the address bar from the browser window. This feature forces all Web traffic through the system by precluding users in the specified role from typing a new URL in the address bar, which circumvents the system.
    - **Do not display browser toolbar** - Select this option to remove the menu and toolbar from the browser. This feature removes all menus, browsing buttons, and bookmarks from the browser window so that the user browses only through the system.
11. If you are configuring the bookmark through the resource profile pages, under Roles, specify the roles to which you want to display the bookmark:
    - **ALL selected roles** - Select this option to display the bookmark to all of the roles associated with the resource profile.
    - **Subset of selected roles** - Select this option to display the bookmark to a subset of the roles associated with the resource profile. Then select roles from the **ALL Selected Roles** list and click **Add** to move them to the **Subset of selected roles** list.
12. Click **Save Changes.**

Creating Standard Web Bookmarks

Information in this section is provided for backwards compatibility. We recommend that you configure access to Web URLs and servers through resource profiles instead, since they provide a simpler, more unified configuration method.

Use the Bookmarks tab to create bookmarks that appear on the welcome page for users mapped to this role. You can create two types of bookmarks through this page:

- Web URL bookmarks - These bookmarks link the user to Web URLs on the World Wide Web or on your corporate Intranet. When you create Web bookmarks, you can insert the user's username in the URL path to provide single sign-on access to back-end Web applications. For Web bookmark configuration instructions, see the instructions that follow.
- **Java applet bookmarks** - These bookmarks link the user to a Java applets that you upload through the Users > Resource Profiles > Web > Hosted Java Applets page of the admin console.

When you create either of these bookmark types, the corresponding links appear on the welcome page for users mapped to this role.

To create a bookmark to a Web resource:

1. In the admin console, choose **Users > User Roles > Role > Web > Bookmarks.**
2. Click **New Bookmark**.
3. Select **Standard**.
4. Enter a name and description for the bookmark (optional). This information displays on the home page instead of the URL.
5. Enter a **Category** for the URL. See Figure 1.
6. Enter the URL to bookmark. If you want to insert the user's username, enter <username> at the appropriate place in the URL.

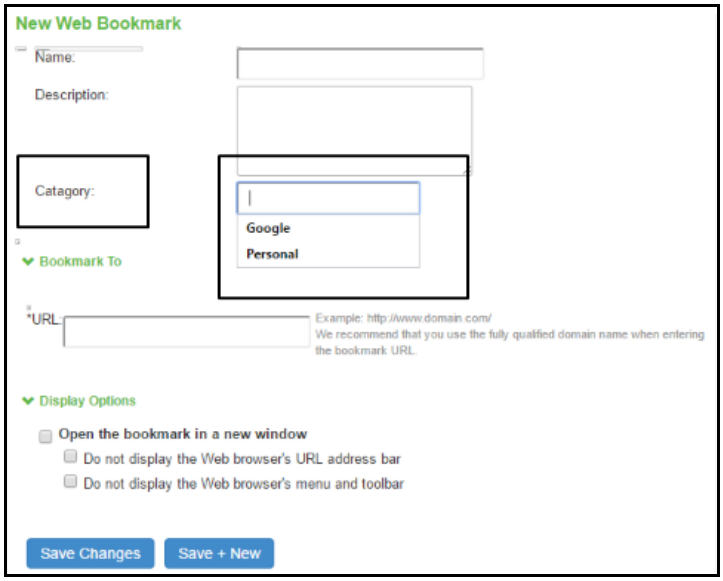
   Make sure to enter a unique URL in this field. If you create two bookmarks with the same URL, the system deletes one of the bookmarks from the end-user view. You will still be able to see both bookmarks, however, in the administrator console.
7. Under Auto-allow, click **Auto-allow Bookmark** to automatically create a corresponding Web access resource policy. Note that this functionality applies only to role bookmarks and not bookmarks created by users. Next, select:
   - **Only this URL** to allow users to access only the URL.
   - **Everything under this UR**L to allow the user to access any path under the URL.

     You may not see the Auto-allow option if you are using a new installation or if an administrator hides the option.
8. Under Display options, click **Open bookmark in a new window** to automatically open the Web resource in a new browser window. Note that this functionality applies only to role bookmarks and not bookmarks created by users. Next, select:

- **Do not display the URL address bar** if you want to remove the address bar from the browser window. This feature forces all Web traffic through the system by precluding users in the specified role from typing a new URL in the address bar, which circumvents the system.
- **Do not display the menu and the toolbar** to remove the menu and toolbar from the browser. This feature removes all menus, browsing buttons, and bookmarks from the browser window so that the user browses only through the system.

9. Click **Save Changes** or **Save + New** to add another.

Figure 1 Categorize Bookmarks



Specifying Web Browsing Options

The system enables you to configure a wide-variety of Web browsing options for a user role.

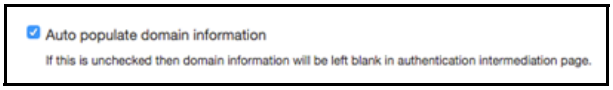
To configure the Web browsing options for a role:

1. Select **Users > User Roles > RoleName > Web > Options.** Complete the configuration as described in Table 4.
2. Click **Save Changes.**

Table 4 Web Browsing Options for a Role

Settings	Guidelines
User can type URLs in the browse bar	(Default) Select this option to enable users to enter URLs on the welcome page and browse to Internet sites.
User can add bookmarks	(Default) Select this option to enable users to create personal web bookmarks on the system welcome page.
Mask hostnames while browsing	Select this option to obscure the target resources in the URLs to which the users browse. When you select this option, the system masks IP addresses and hostnames in the user's: • Web browser address bar (when the user navigates to a page) • Web browser status bar (when a user hovers over a hyperlink) • HTML source files (when the user chooses to View Source) The hostname encoding feature (also called hostname obfuscation or URL obfuscation) prevents casual observers from noting the URL of an internal resource by obscuring the target server within the URL without masking the full path name, target file, or port number. For example, if a user navigates to www.msn.com without selective rewriting or hostname encoding enabled, the system displays an unobscured URL in his Web browser's address bar: http://www.msn.com/ If you then enable selective rewriting, the system might display the following URL: https://mycompanyserver.com/,DanaInfo=www.msn.com,SSO=U+ If you then enable hostname encoding, and the same user navigates to the same site, he sees a URL in which the hostname (www.msn.com) is obscured: https://i5.asglab.pulsesecure.net/,DanaInfo=.awxyCqxtGkxw,SSO=U+ Hostname encoding uses a lightweight reversible algorithm so that users can bookmark encoded URLs. (The system can translate the encoded URL and resolve it back to the original URL.) For compatibility, previously created bookmarks to unmasked URLs continue to work when hostname encoding is enabled. **Note:** • If you enable selective rewriting and hostname encoding, the system only obscures the hostnames and IP addresses of those servers that you have chosen to rewrite using the selective rewrite feature. • Links not rewritten by the system are not obscured. For example, the rewriter does not intermediate ftp, rtsp, mms and mailto links and therefore the hostnames in these links are not masked. This is required to pass security audits. • If you enable the framed toolbar and hostname encoding, the system does not obscure hostnames that the user enters in the framed toolbar's browse field. • The system does not obscure hostnames and IP addresses in log entries, including hostname encoding log entries.
Advanced options	

Settings	Guidelines
Allow Java applets	(Default) Select this option to enable users to browse to Web pages containing client-side Java applets. The system appears to the application server as a browser over SSL. The system transparently handles any HTTP requests and TCP connections initiated by a Java applet and handles signed Java applets. If you enable this feature, users can launch Java applets and run applications that are implemented as client-side Java applets, such as the Virtual Computing (VNC) Java client, Citrix NFuse Java client, WRQ Reflections Web client, and Lotus WebMail.
Allow Flash content	(Default) Select this option to enable the system to intermediate Flash content through its Content Intermediation Engine. Note that the system provides limited support for ActionScript 2.0 and Flash Remoting, and does not support XMLSocket connections. The Content Intermediation Engine supports Flash versions 5, 6, 7 and 8, including dynamic rewriting of internal Web links during an access request. We support the rewriting of Actionscript in Flash. The calls in Actionscript that are supported are: load, send, sendAndLoad, loadVariables, loadMovie, loadVariablesNum, loadMovieNum, loadClip, loadSound, apply, connect on classes of XML, Sound, MovieClip, NetConnection, and MovieClipLoader. The eval equivalent of Actionscript is not supported. Therefore, we recommend that the above function calls not be embedded in an Actionscript string object. Note, Flash applications that use the XMLSocket object or Flash Remoting are not supported. For more information, see the Content Intermediation Engine Best Practices Guide.
Persistent cookies	(Default) Select this option to enable users to customize their browsing experiences by enabling them to keep persistent cookies. By default, the system flushes Web cookies that are stored during a user session. A user can delete cookies through the Advanced Preferences page if you enable this option.
Unrewritten pages open in new window	Select this option to configure the system to open content in a new browser window when a user access an unrewritten Web page. Opening content in a new window can help remind users that they still have a secure session. When a user request is made to a resource to which this option applies, the system displays a page that contains a link to the requested resource and directs the users to click on the link. This link opens the resource in a new browser window and the page from which the request originates continues to display in the system. If you uncheck this box, users might not realize that their session is still active and that to return to the system, they need to use the browser's Back button. Users must return to the system to sign out. If they simply close the browser window, their sessions remain active until the session time limit expires.

Settings	Guidelines
Allow browsing untrusted SSL Web servers	(Default) Select this option to allow access to untrusted web sites through the system. Untrusted web sites are those whose server certificates are not installed, expired, or revoked through the System > Configuration > Certificates > Trusted Servers CAs tab of the admin console. **Note:** If a web page has internal references to files within a SCRIPT tag and these files are hosted on different HTTPS servers that have SSL certificates not trusted by the system, the web page does not render correctly. In these cases, the Warn users about the certificate problems option must be disabled. Warn users about the certificate problems. (Default) Select this option to warn users about the certificate problems option and the user accesses non-HTML content (such as images, js, and css) served from a different SSL server than the HTML page, the page containing the links may not display correctly. You can avoid this problem either by deselecting this option or by uploading a valid production SSL certificate on the servers that serve the non-HTML content. If enabled, display a warning to the user when he first accesses an untrusted web site telling him why the site's certificate is untrusted and allowing him to either continue or cancel. If the user chooses to continue after viewing the warning, the system does not display any more warnings for that site during the current session. **Note:** This option is not applicable for auth-only URLs (for example, ActiveSync) and Secure Mail URLs. • **Allow users to bypass warnings on a server-by-server basis**. Select this option to allow the user to suppress all further warnings for an untrusted web site. If a user chooses this option, he never sees a warning for this site again, provided that he accesses it from the current device or cluster. If you choose to allow users to access untrusted web sites without seeing a warning, the system still logs a message to the user access log whenever a user navigates to an untrusted site. Also note that if a user chooses to suppress warnings, he can clear the persistent settings of the untrusted web sites using the Delete Passwords option in the System > Preferences > Advanced tab in the end user console.
Rewrite file:// URLs	Select this option to rewrite file:// URLs so that they are routed through the system's file browsing CGI.
Rewrite links in PDF files	Select this option to rewrite hyperlinks in PDFs.
Auto populate domain information	Select this option to display the domain information in the end user authentication intermediate page that prompts for credentials. When this option is not selected, the domain text box will be blank. 
HTTP Connection Timeout	

Settings	Guidelines
HTTP Connection Timeout	Specify the duration to wait for a response from an HTTP server before timing out and closing the connection. Use values from 30 to 1800 seconds (default is 240). Higher timeout values might exhaust system resources if applications do not close connections properly or take too long to close the connections. Unless an application requires a higher timeout value, we recommend accepting the default value.
WebSocket Connection Timeout	Specify the duration to wait for data transfer between the client and server. Use values from 30 to 1800 seconds (default is 900). WebSocket is a web technology that provides bidirectional, full-duplex communication channels over a single TCP connection. This provides a mechanism for browser-based applications that need two-way communication with servers that do not rely on opening multiple HTTP connections. Communication is done over the regular TCP port numbers 80 or 443. Currently, only the following web resource policies support WebSocket: • Web ACL Access • Passthrough Proxy • Options The WebSocket URL that starts with ws:// or wss:// is not allowed in any of the web resource profile pages, web resource policies or web bookmark pages. The following Web options under Roles accept WebSocket requests: • Mask hostnames while browsing • Persistent cookies • Allow browsing untrusted SSL web sites
ActiveSyncLongLived Connection Timeout	Specify the duration of a long-lived request used to synchronize an iOS device with a Microsoft Exchange server (Secure Mail must be enabled for the role). When the request expires, the device issues a new request. Use values from 30 to 7200 seconds. Microsoft recommends using 1800 seconds (the default).

Resource Policy Overview

When you enable the Web access feature for a role, you need to create resource policies that specify which resources a user can access, whether or not to rewrite the content requested by the user, and caching, applet, or single sign-on requirements. For every Web request, the system first evaluates the rewriting policies you configure. If the user's request is to a resource specified as "don't rewrite" due to either a selective rewriting or passthrough proxy resource policy, then forward the user's request to the appropriate back-end resource. Otherwise, the system continues to evaluate those resource policies corresponding to the request, such as Java resource policies for a request to fetch a Java applet. After matching a user's request to a resource listed in a relevant policy, the system performs the action specified for the resource.

You can create resource policies through the standard interface (as described in this section) or through resource profiles (recommended method).

When writing a Web resource policy, you need to supply key information:

- **Resources** - A resource policy must specify one or more resources to which the policy applies. When writing a Web policy, you need to specify Web servers or specific URLs, as explained in the section that follows.

- **Roles** - A resource policy must specify the roles to which it applies. When a user makes a request, the system determines what policies apply to the role and then evaluates those policies that correspond to the request.
- **Actions** - Each type of resource policy performs a certain action, which is either to allow or deny a resource or to perform or not perform some function, such as rewrite content, re-sign an applet, or post Web data. You can also write detailed rules that apply more conditions to a user request.

The system platform's engine that evaluates resource policies requires that the resources listed in a policy's Resources list follow a canonical format.

Canonical Format

This section outlines special considerations you must consider when specifying a Web resource using the canonical format.

- [protocol://]host[:ports][/path]

The four components are:

- **Protocol** (optional) - Possible values: http and https (case-insensitive)

  If the protocol is missing, then both http and https are assumed. If a protocol is specified, then the delimiter "://" is required. No special characters are allowed.
- **Host** (required) - Possible values:
  - **DNS Hostname** - For example: www.pulsesecure.net

    Allowed special characters are described in Table 5.

Table 5 DNS Hostname Special Characters

*	Matches ALL characters
%	Matches any character except dot (.)
?	Matches exactly one character

  - **IP address/Netmask** - The IP address needs to be in the format: a.b.c.d

    The netmask can be in one of two formats:
    - Prefix: High order bits
    - IP: a.b.c.d
      For example: ÌPv4 format: 10.11.149.2/24 or 10.11.149.2/255.255.255.0;
      IPv6 format: [2001:db8:a0b:12f0::1/64]:80,443/public/*
      [2001:db8:a0b:12f0::1/64]:8000-9000/*
      No special characters are allowed.
- **Ports** - You must specify a port when specifying IP/netmask as a resource. The port is optional when specifying a DNS hostname. If a port is specified, then the delimiter ":" is required. For example: 10.11.149.2/255.255.255.0:* Table 6 lists the possible port values.

Table 6 Possible Port Values

*	Matches ALL ports; no other special characters are allowed
port[,port]*	A comma-delimited list of single ports. Valid port numbers are [1-65535].
[port1]-[port2]	A range of ports, from port1 to port2, inclusive.

Note: You can mix port lists and port ranges, such as: 80,443,8080-8090

If the port is missing, then the default port 80 is assigned for http, 443 for https.

- Path (optional)-If the path is missing, then star (*) is assumed, meaning ALL paths match. If a path is specified, then the delimiter "/" is required. No other special characters are supported. For example:
  - http://www.pulsesecure.net:80/*
  - https://www.pulsesecure.net:443/intranet/*
  - *.yahoo.com:80,443/*
  - %.danastreet.net:80/share/users/<username>/*

Writing a Web Access Resource Policy

Web access resource policies control which Web resources users can access in order to connect to the Internet, intranet, or extranet. You can deny or allow access to Web resources by URL or IP range. For URLs, you can use the "*" and "?" wildcards to efficiently specify multiple hostnames and paths. For resources that you specify by hostname, you can also choose either HTTP, HTTPS, or both protocols.

To write a Web Access resource policy:

1. In the admin console, choose **Users > Resource Policies > Web > Web ACL.**
2. On the Web Access Policies page, click **New Policy.**
3. On the **New Policy** page, enter a name to label this policy and optionally a description.
4. In the Resources section, specify the resources to which this policy applies.
5. In the Roles section, specify:
   - **Policy applies to ALL roles** - To apply this policy to all users.
   - **Policy applies to SELECTED roles** - To apply this policy only to users who are mapped to roles in the Selected roles list. Make sure to add roles to this list from the Available roles list.
   - **Policy applies to all roles OTHER THAN those selected below** - To apply this policy to all users except for those who map to the roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.
6. In the Action section, specify:
   - **Allow access** - To grant access to the resources specified in the Resources list.
   - **Deny access** - To deny access to the resources specified in the Resources list.

- **Use Detailed Rules** - To specify one or more detailed rules for this policy.

7. Click **Save Changes**.
8. On the Web Access Policies page, order the policies according to how you want to evaluate them. Keep in mind that once the system matches the resource requested by the user to a resource in a policy's (or a detailed rule's) **Resource** list, it performs the specified action and stops processing policies.

Defining Single Sign-On Policies

Single sign-on policies enable you to automatically pass user credentials to the Web application specified in your policy. You can configure single sign-on policies to intercept basic authentication, Kerberos and NTLM challenges and display an intermediate sign-in page to collect credentials for the Web resource. Or, you can post the credentials and headers that you specify to the Web application.

About Basic, NTLM and Kerberos Resources

Use the SSO > General tab to set up the basic, NTLM and Kerberos credentials. The credentials you define here are used when defining Web resource profiles with SSO autopolicies and Web resource policies.

The following outlines the basic ideas behind the handling of SSO:

- The system will do Kerberos if challenged with Negotiate header, NTLM if challenged with NTLM header and Basic Auth if challenged with Basic.
- If the system receives multiple challenges, the order of preference is:
  - Kerberos
  - NTLM
  - Basic
- The system will first try constrained delegation if the service is configured in a service list.
- Policy configurations override any settings in the SSO > General tab.
- Disabling SSO or disabling all sections in the General tab prevents single sign-on. However, the system will continue to intermediate and display an intermediation page to the end user.
- Basic authentication intermediation can be explicitly turned off in a policy. For kerberos and NTLM, the system will always intermediate.
- Depending on the SSO used, the intermediation page will show different fields for the end user to complete:
  - Basic authentication intermediation page displays username and password fields
  - NTLM intermediation page displays username, password and domain fields
  - Kerberos intermediation page displays username, password and realm fields
- For constrained delegation, you must define a policy and specify roles. Entering data in the General tab only is not sufficient.
- If no policies are configured for single sign-on, the system uses the default system credentials.

- If credentials are defined, the order of preference is:
  - System credentials
  - Variable credentials
  - Fixed or static credentials
- For fixed or static credentials, you must define a policy and specify roles. Entering data in the General tab only is not sufficient.
- If there is a policy match, the credential and protocol of the policy is used. If the policy fails to authenticate, the fallback mechanism defined in the policy is used. If the policy protocol does not match the protocol of the challenge, the logic defined in the General tab is used.
- When upgrading a device or performing a new install, the default SSO policy of BasicAuthNoSSO is preserved. Even if all sections of the General tab are enabled, SSO will not be enabled until the BasicAuthNoSSO policy is deleted.

Writing the Basic, NTLM and Kerberos Resources

To set up the basic, NTLM and Kerberos resources:

1. In the admin console, select **Users > Resource Policies > Web**.
2. If your administrator view is not already configured to show SSO policies, make the following modifications:
   1. Click the **Customize** button in the upper right corner of the page.
   2. Select the **SSO** check box.
   3. Select the **General** check box below the **SSO** check box.
   4. Click **OK**.
3. Select the **SSO > General** tab.
4. Select **Enable kerberos** to enable Kerberos SSO. You can then define the type of intermediation: constrained delegation or Connect Secure. If you do not define any intermediation types, the system attempts to figure out the realm from the hostname and performs SSO using the system credentials.

For realm intermediation, enter the following and click **Add**:

- **Realm** - Enter the Kerberos realm name. For example, KERBER.NET. The system uses KERBER.NET to obtain the list of Key Distribution Centers (KDCs).
- **Site Name** - (optional) Enter the Active Directory site names. Use this field to have the system contact the KDC at a specific site. For example, if site name is Sunnyvale and realm is KERBER.NET, then the system uses SunnyvaleoKERBER.NET to get a list of KDCs. Note that the Active Directory must have the sites defined and DNS should be configured to return the KDCs in the site.
- **Pattern List** - Enter the hostnames mapped to the Kerberos realm. You can enter wildcard characters, such as *.y.com, *.kerber.net, or *.*. Note the following:
  - Make sure that realms to not have hostnames matching a subset of the patterns defined for another realm.

    - You do not need to define a pattern if all servers follow the mirrored DNS namespace convention. The system determines the realm from the hostname.
    - All disjoined hostname patterns must be defined.
    - You can use * as the default realm. Do not list more than one * when defining multiple realms.
- **KDC** - Enter the hostname or IP address of the Key Distribution Centers if DNS is unavailable or if you want the system to contact a specific KDC for tickets. If you enter a KDC, the system does not use DNS to obtain the list of KDCs based on the values entered in the Site Name and Realm fields.

For constrained delegation intermediation, enter the following and click **Add**:

- **Label** - Enter a name to uniquely identify this row. No external mapping is made to the label value.
- **Realm** - Select the realm to use. The drop-down list is populated by values in the Realm Definition table.
- **Principal Account** - Enter the constrained delegation account to use to get constrained delegation tickets on behalf of the user.
- **Password** - Enter the constrained delegation account password.
- **Service List** - Select the service list to use. Click Edit to define and upload service lists. The list should be an exact match with the service list in Active Directory if you want to perform constrained delegation for all the services. Hostnames must be an exact match.

  For more information about constrained delegation, see http://msdn.microsoft.com/en-us/library/aa480585.aspx.

For system intermediation, enter the following and click **Add**:

- **Label** - Enter a name to uniquely identify this row. No external mapping is made to the label value.
- **Realm** - Select the realm to use. The drop-down list is populated by values in the Realm Definition table.
- **Credential Type** - Select one of the following credential types:
    - **System credentials** - Use the set of user credentials, such as primary and secondary authorization credentials, stored on the device. If you select this option, you do not need to enter values in the Username and Password fields.
    - **Variable** - Allow tokens such as <username> and <password> to be used in the username and Variable Password fields.
    - **Static** - Use the username and password exactly as they are entered in the username and password fields.
- **Username and Password** - Enter the account username and password. If you select Variable as the credential type, you can enter the username token here. For example, <username>.
- **Variable Password** - If you select Variable as the credential type, enter the password token here. For example, <password>.
- **Fallback to NTLM V2** - Select this option to fallback to NTLM V2 if Kerberos fails. If you do not select this option and Kerberos SSO fails, an intermediation page appears.

5. Select **Enable NTLM** to enable NTLM SSO. If you do not enter any configuration information, the system attempts to figure out the domain from the hostname and performs SSO using the system credentials.

Note: Do not edit or delete the default system credential.

- **Label** - Enter a name to uniquely identify this row. No external mapping is made to the label value.
- **Domain** - Enter the Active Directory domain name here.
- **Credential Type** - Select one of the following credential types:
  - **System credentials** - Use the set of user credentials, such as primary and secondary authorization credentials, stored on the device. If you select this option, you do not need to enter values in the Username and Password fields.
  - **Variable** - Allow tokens such as <username> and <password> to be used in the Username and Variable Password fields.
  - **Static** - Use the username and password exactly as they are entered in the username and password fields.
- **Username and Password** - Enter the account username and password. If you select Variable as the credential type, you can enter the username token here. For example, <username>.
- **Variable Password** - If you select Variable as the credential type, enter the password token here. For example, <password>.
- **Fallback to NTLM V1** - Select this option to fallback to NTLM V1 if SSO fails. If you do not select this option and SSO fails, only NTLM V2 is attempted. An intermediation page appears if NTLM V2 fails.

6. Select **Enable Basic Authentication** to enable basic authentication SSO. If you select this option but do not set up any configuration data, the system will attempt SSO using system credentials.

Note: Do not edit or delete the default system credential.

- **Label** - Enter a name to uniquely identify this row. No external mapping is made to the label value.
- **Credential Type** - Select one of the following credential types:
  - **System credentials** - Use the set of user credentials, such as primary and secondary authorization credentials, stored on the device. If you select this option, you do not need to enter values in the Username and Password fields.
  - **Variable** - Allow tokens such as <username> and <password> to be used in the Username and Variable Password fields.
  - **Static** - Use the username and password exactly as they are entered in the username and password fields.
- **Username and Password** - Enter the account username and password. If you select Variable as the credential type, you can enter the username token here. For example, <username>.
- **Variable Password** - If you select Variable as the credential type, enter the password token here. For example, <password>.
- **Pattern List** - Enter the hostnames mapped to the Kerberos realm. You can enter wildcard characters, such as *.y.com, *.kerber.net, or *.*. Note the following:

- Make sure that realms to not have hostnames matching a subset of the patterns defined for another realm.
- You do not need to define a patter if all servers follow the mirrored DNS namespace convention. The system determines the realm from the hostname.
- All disjoined hostname patterns must be defined.
- You can use * as the default realm. Do not list more than one * when defining multiple realms.
- You can use * as the default domain. Do not list more than one * when defining multiple domains.

Writing a Basic Authentication, NTLM or Kerberos Intermediation Resource Policy

Basic Authentication, NTLM or Kerberos Intermediation resource policies enable you to control NTLM and Kerberos intermediation on the system. If a user accesses a Web resource that sends a basic authentication challenge, the system can intercept the challenge, display an intermediate sign-in page to collect credentials for the Web resource, and then rewrite the credentials along with the entire challenge/response sequence.

The initial HTTP request generated for an NTLM protected server should be for a request that results in HTML content. If SSO is not enabled or if the SSO credentials fail, the system responds with an HTML page to gather user credentials. If the browser is expecting non-HTML content, the browser rejects the response and the navigation to the resource fails.

With the Kerberos Intermediation resource policy, backend web applications protected by Kerberos are accessible to end users. For example, a user logs in to a device using Active Directory as the authentication server and the authentication protocol is Kerberos. When the user browses to a Kerberos-protected server, the user is single-signed on to the backend server and is not prompted for credentials. Or, if a user logs in to a device using an authentication protocol other than Kerberos and then browses to a Kerberos-protected server. Depending on the settings in Kerberos Intermediation resource policy and the configured Kerberos authentication server, the user will either be authenticated by the rewriter or the user will be prompted to enter a username and password.

To write a Basic Authentication, NTLM or Kerberos Intermediation resource policy:

1. In the admin console, select **Users > Resource Policies > Web.**
2. If your administrator view is not already configured to show SSO policies, make the following modifications:
    1. Click the **Customize** button in the upper right corner of the page.
    2. Select the **SSO** check box.
    3. Select the **Kerberos/Basic Auth/NTLM** check box below the **SSO** check box.
    4. Click **OK**.
3. Select the **SSO > Kerberos/NTLM/BasicAuth** tab.
4. Click **New Policy.**
5. Enter a name to label this policy (required) and a description of the policy (optional).

6. In the Resources section, specify the resources to which this policy applies.

   If you want to automatically post values to a specific URL when an end user clicks on a bookmark, the resource that you enter here must exactly match the URL that you specify in the Users > User Roles > Role > Web > Bookmarks page of the admin console.

7. In the Roles section, specify:

   - **Policy applies to ALL roles** - To apply this policy to all users.
   - **Policy applies to SELECTED roles** - To apply this policy only to users who are mapped to roles in the Selected roles list. Make sure to add roles to this list from the Available roles list.
   - **Policy applies to all roles OTHER THAN those selected below** - To apply this policy to all users except for those who map to the roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.

8. In the Action section, specify:

   - **Disable SSO** - Disables automatic SSO authentication for this user role and, instead, prompts the user for sign-in credentials.
   - **Basic** - This option uses the Basic Authentication Intermediation method to control SSO behavior.
     - **Enable Intermediation** - Select the credentials to use. If this pull-down menu is blank, no basic authentication SSO settings are defined in the SSO General tab.
     - **Disable Intermediation** - When you select this option, the system does not intermediate the challenge/response sequence.

     The system always intermediates requests to Web proxies that require basic authentication, even if you select Disable Intermediation.

     Although you are given an option to disable basic authentication intermediation, we do not recommend this option, as it is a very insecure authentication method and, in some cases, can transmit user credentials over the network in clear (unencrypted) text.
   - **NTLM** - This option specifies that the system use the Microsoft NTLM Intermediation method to control SSO behavior.
     - Select the credentials to use. If this pull-down menu is blank, no NTLM SSO settings are defined in the SSO General tab.
     - Select the **Fallback to NTLM V1** option to try both NTLM V1 and NTLM V2. If you do not select this option, the system falls back only to NTLM V2. An intermediation page appears if SSO fails.
   - **Kerberos** - This option specifies that the system use the Kerberos Intermediation method to control SSO behavior.
     - Select the credentials to use. If this pull-down menu is blank, no kerberos SSO settings are defined in the SSO General tab
     - Select the **Fallback to NTLM V2** option to fallback only to NTLM V2 if kerberos fails. If you do not select this option, a Kerberos intermediation page appears if Kerberos SSO fails.
   - **Constrained Delegation** -This option specifies that the system use the constrained delegation intermediation method to control SSO behavior.

- Select the credentials to use. If this pull-down menu is blank, no constrained delegation SSO settings are defined in the SSO General tab.
- Select the **Fallback to Kerberos** option to fallback to Kerberos if constrained delegation fails. If you select this option, an intermediation page appears if constrained delegation fails. If you do not select this option and constrained delegation fails, an error page appears.

- **Use Detailed Rules** - To specify one or more detailed rules for this policy.

9. Click **Save Changes**.
10. On the Basic Auth, NTLM and Kerberos policies page, order the policies according to how you want to evaluate them. Keep in mind that once the system matches the resource requested by the user to a resource in a policy's (or a detailed rule's) Resource list, it performs the specified action and stops processing policies.

    Check the activity events listed in the user log if you encounter any problems.

Writing a Remote SSO Form POST Resource Policy

Remote SSO Form POST resource policies specify Web applications to which the system posts data. This data can include a user's username and password, as well as system data stored by system variables.

To write a remote SSO Form POST resource policy:

1. In the admin console, navigate to **Users > Resource Policies > Web**.
2. If your administrator view is not already configured to show SSO policies, make the following modifications:
    1. Click the **Customize** button in the upper right corner of the page.
    2. Select the **SSO** check box.
    3. Select the **Form Post** check box below the SSO check box.
    4. Click **OK**.
3. Select the **SSO> Form Post** tab.
4. On the Form POST Policies page, click **New Policy**.
5. Enter a name to label this policy (required) and a description of the policy (optional).
6. In the Resources section, specify the application's sign-in page, such as: http://yourcompany.com.

    If you want to automatically post values to a specific URL when an end user clicks on a bookmark, the resource that you enter here must exactly match the URL that you specify in the Users > User Roles > Role > Web > Bookmarks page of the admin console.
7. In the Roles section, specify:
    - **Policy applies to ALL roles** - To apply this policy to all users.
    - **Policy applies to SELECTED roles** - To apply this policy only to users who are mapped to roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.

- **Policy applies to all roles OTHER THAN those selected below** - To apply this policy to all users except for those who map to the roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.

8. In the Action section, specify:
   - **Perform the POST defined below** - Perform a form POST with the user data specified in the POST details section to the specified URL when a user makes a request to a resource specified in the Resources list.
   - **Do NOT perform the POST defined below** - Do not perform a form POST with the user data specified in the POST details section.
   - **Use Detailed Rules** - Select this option to specify one or more detailed rules for this policy.
9. In the POST details section:
   - In the **POST to URL** field, specify the absolute URL where the application posts the user's credentials, such as: http://yourcompany.com/login.cgi. The admin can determine the appropriate URL using a TCP dump or by viewing the application's sign-in page source and searching for the POST parameter in the FORM tag. (Wildcard characters are not supported in this field.)
   - Check **Deny direct login** for this resource if you do not want users to be able to access the URL directly.
   - Select the **Allow multiple POSTs to this resource** check box if you want to send POST and cookie values to the resource multiple times if required. If you do not select this option, the system does not attempt single sign-on when a user requests the same resource more than once during the same session.
   - Specify the user data to post and user modification permission:
     - **User label** - The label that appears on a user's Preferences page. This field is required if you either enable or require users to modify data to post to back-end applications.
     - **Name** - The name to identify the data of the Value field. (The back-end application should expect this name.)
     - **Value** - The value to post to the form for the specified Name. You can enter static data, a system variable, or system session variables containing username and password values.
     - **User modifiable**? setting - Set to **Not modifiable** if you do not want the user to be able to change the information in the Value field. Set to User CAN change value if you want the user to have the option of specifying data for a back-end application. Set to User MUST change value if users must enter additional data in order to access a back-end application. If you choose either of the latter settings, a field for data entry appears on the user's Advanced Preferences page. This field is labeled using the data you enter in the User label field. If you enter a value in the Value field, this data appears in the field but is editable.
10. Click **Save Changes.**
11. On the Form POST Policies page, order the policies according to how you want to evaluate them. Keep in mind that once the system matches the resource requested by the user to a resource in a policy's (or a detailed rule's) Resource list, it performs the specified action and stops processing policies.

Writing a Remote SSO Headers/Cookies Resource Policy

Remote SSO Headers/Cookies resource policies specify customized Web applications to which the system posts custom headers and cookies.

When creating a Headers/Cookies policy, note that the system does not parse or "understand" the headers that you enter in this section. For instance, if you add an Accept-Encoding: gzip or Accept-Encoding:deflate header, it does not mean that the system can handle gzip content or deflated content.

To write a remote SSO Headers/Cookies resource policy:

1. In the admin console, select **Users > Resource Policies > Web.**
2. If your administrator view is not already configured to show SSO policies, make the following modifications:
    1. Click the **Customize** button in the upper right corner of the page.
    2. Select the **SSO** check box.
    3. Select the **Headers/Cookies** check box below the **SSO** check box.
    4. Click **OK**.
3. Select the S**SO > Headers/Cookies** tab.
4. On the Headers/Cookies Policies page, click **New Policy**.
5. Enter a name to label this policy (required) and a description of the policy (optional).
6. In the Resources section, specify the resources to which this policy applies.
7. In the Roles section, specify:
    - **Policy applies to ALL roles** - To apply this policy to all users.
    - **Policy applies to SELECTED roles** - To apply this policy only to users who are mapped to roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.
    - **Policy applies to all roles OTHER THAN those selected below** - To apply this policy to all users except for those who map to the roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.
8. In the Action section, specify:
    - **Append headers as defined below** - Post the user data specified in the POST details section to the specified URL when a user makes a request to a resource specified in the Resources list.
    - **Do NOT append headers as defined below** - Do not post the user data specified in the POST details section to the specified URL when a user makes a request to a resource specified in the **Resources** list.
    - **Use Detailed Rules** - Select this option to specify one or more detailed rules for this policy.
9. In the Headers and values section, specify the:
    - **Header name** - The text for to send as header data.

- **Value** - The value for the specified header.

Note: If you need to forward a cookie to a backend server, you must set the Header Name field to "Cookie" and the Value field to "CookieName=CookieValue".

10. Click **Save Changes**.
11. On the Headers/Cookies Policies page, order the policies according to how you want to evaluate them. Keep in mind that once the system matches the resource requested by the user to a resource in a policy's (or a detailed rule's) Resource list, it performs the specified action and stops processing policies.

Writing a Web Caching Resource Policy

To write a Web Caching resource policy:

1. In the admin console, select **Users > Resource Policies > Web**.
2. If your administrator view is not already configured to show caching policies, make the following modifications:
    1. Click the **Customize** button in the upper right corner of the page.
    2. Select the **Caching** check box.
    3. Select the **Policies** check box below the **Caching** check box.
    4. Click **OK**.
3. Select the **Caching > Policies** tab.
4. On the Web Caching Policies page, click **New Polic**y.
5. Enter a name to label this policy (required) and a description of the policy (optional).
6. In the Resources section, specify the resources to which this policy applies.
7. In the Roles section, specify:
    - **Policy applies to ALL roles** - To apply this policy to all users.
    - **Policy applies to SELECTED roles** - To apply this policy only to users who are mapped to roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.
    - **Policy applies to all roles OTHER THAN those selected below** - To apply this policy to all users except for those who map to the roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.
8. In the Action section, select one of the following options:
    - **Smart Caching (send headers appropriate for content and browser**) - Select this option to send a cache-control:no-store header or a cache-control:no-cache header based on the user's Web browser and content type.

When you select this option, the system makes media files and zip files work properly by removing their origin server's cache-control headers. For example, the following logic searches for "msie" or "windows-media-player" in user-agent headers in order to remove cache or cache-control:no-store response headers and make the files cacheable:

(if content type has "audio/x-pn-realaudio" OR

if content type begins with "video/" OR

if content type begins with "audio/" OR

if content type is "application/octet-stream" and the file extension begins with "rm" or "ram"

)

If the system finds "msie" or "windows-media-player" in the user-agent header and any of the following apply:

- Request is for Flash, .xls, .pps, .ppt files
- Content-type is application/, text/rtf, text/xml, model/
- Origin server sends a content-disposition header

then the system sends the cache-control:no-store header and removes the origin server's cache-control header.

In all other cases, the system adds the pragma:no-cache or cache-control:no-store response headers.

Citrix .ica and QuickPlace files get some special treatment. Citrix .ica files are always cacheable and get cache-control-private as well. QuickPlace files that do not match a specified rule files (which takes precedence) get CCNS and cache-control:private.

Also note that if you select this option, enable GZIP compression, and try to access a text file attachment using Domino Web Access 6.5 through Internet Explorer, you cannot open the attachment. To enable text attachments, you must either install the Internet Explorer 323308 patch or enable the Don't Cache (send "Cache Control: No Store") option.

- **Don't Cache (send "Cache Control: No Store")** - Select this option to deliver attachments to Internet Explorer without saving them to the disk. (The browser temporarily writes files to the disk, but immediately removes them once it has opened the file in the browser.) When you select this option, the system removes the origin server's cache-control header and adds a cache-control:no-store response header if the user-agent string sent by the browser contains "msie" or "windows-media-player."

  This option might slow browsing by causing repeated content fetches, which can cause performance issues on very slow connections. Alternatively, you can specify a policy that allows certain kinds of content to be cached, such as images that do not exceed a specified size limit.

- **Don't Cache (send "Pragma: No Cache")** - Select this option to prevent the user's browser from caching files to the disk. When you select this option, the system adds the standard HTTP pragma:no-cache header and cache-control:no-cache (CCNC) header (HTTP 1.1) to response files. Also, the system does not forward the origin server's caching headers, such as age, date, etag, last-modified, expires.

  When no-cache headers are present on certain types of attachments (PDF, PPT, streaming files), Internet Explorer does not properly render the documents because the rendering process requires the browser to temporarily writes these files to cache.

- **Unchanged (do not add/modify caching headers)** - Select this option to not add the pragma:no-cache or cache-control:no-store response headers and forwards the origin server's caching headers.
- **Remove Cache-Control: No-Cache|No** - Store-Select this option to help "cache" files sent by web applications in an HTTPS environment. This option removes the Cache Control:No Cache and Pragma:no-cache headers. Removing these headers is necessary to allow the successful download of certain file types. These headers work fine in an HTTP environment, but fail in an HTTPS environment where the associated pages become uncacheable, preventing the user's web browser from downloading the pages.

  Use this option when you want the end user to have the ability to download and open a file that will be opened by another third-party application. For example, zip files and wav files are stored on disk and opened by another application.
- **Use Detailed Rules** - To specify one or more detailed rules for this policy.

9. Click **Save Changes**.
10. On the Web Caching Policies page, order the policies according to how you want to evaluate them. Keep in mind that once the system matches the resource requested by the user to a resource in a policy's (or a detailed rule's) Resource list, it performs the specified action and stops processing policies.

About OWA and Lotus Notes Caching Resource Policies

Table 7 and Table 8 include examples of some of the content types that the system supports with the Outlook Web Access (OWA) and Lotus iNotes applications. Additionally, it specifies the cache control directives that you must implement in Microsoft Internet Explorer in order to support opening and saving the specified content types.

Note that for performance reasons, we recommend creating caching policies for everything in the iNotes directory.

Table 7 OWA Caching Resource Policies

Attachment type	To open the attachment, use:	To save the attachment, use:
zip	Cache	Smart caching
ppt	Smart caching	Smart caching
doc	Smart caching	Smart caching
xls	Smart caching	Smart caching
pdf	Smart caching	Smart caching
txt	Cache	Cache control: No store
html	Smart caching	Cache control: No store

Table 8 iNotes Caching Resource Policies

Attachment type	To open the attachment, use:	To save the attachment, use:
zip	Cache control: No store	Cache control: No store
ppt	Cache control: No store	Cache control: No store
doc	Smart caching	Smart caching
xls	Cache control: No store	Cache control: No store
pdf	Cache control: No store	Cache control: No store
txt	Cache control: No store	Cache control: No store
html	Cache control: No store	Cache control: No store
other file types	Cache control: No store	Cache control: No store

Specifying General Caching Options

You can use caching options to specify the maximum image file size that is cached on a client. If the content-type header from the origin server begins with "image/" and the content-length header specifies a size less than the maximum size configured for this option, then the system passes along the origin server's caching headers. Otherwise, the system treats the request as though caching is disabled.

To specify caching options:

1. In the admin console, choose **Users > Resource Policies > Web**.
2. If your administrator view is not already configured to show caching policies, make the following modifications:
    1. Click the **Customize** button in the upper right corner of the page.
    2. Select the **Caching** check box.
    3. Select the **Options** check box below the **Caching** check box.

4. Click **OK**.

3. Select the C**aching > Option**s tab.
4. On the Caching Options page, specify a maximum allowable image size in the Clients should cache all images less than field.
5. On the Caching Options page, specify a maximum allowable image size in the Clients should cache all images less than field.

Writing a Java Access Control Resource Policy

Java access control resource policies control to which servers and ports Java applets can connect.

To write a Java access control resource policy:

1. In the admin console, select **Users > Resource Policies > Web**.
2. If your administrator view is not already configured to show Java policies, make the following modifications:
    1. Click the **Customize** button in the upper right corner of the page.
    2. Select the **Java** check box.
    3. Select the **Access Control** check box below the **Java** check box.
    4. Click **OK**.
3. Select the **Java > Access Control t**ab.
4. On the Java Access Policies page, click **New Policy**.
5. Enter a name to label this policy (required) and a description of the policy (optional).
6. In the Resources section, specify the resources to which this policy applies.
7. In the Roles section, specify:
    - **Policy applies to ALL roles** - To apply this policy to all users.
    - **Policy applies to SELECTED roles** - To apply this policy only to users who are mapped to roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.
    - **Policy applies to all roles OTHER THAN those selected below** - To apply this policy to all users except for those who map to the roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.
8. In the Action section, specify:
    - **Allow socket access** - To enable Java applets to connect to the servers (and optionally ports) in the Resources list.
    - **Deny socket access** - To prevent Java applets from connecting to the servers (and optionally ports) in the Resources list.
    - **Use Detailed Rules** - To specify one or more detailed rules for this policy.

9. Click **Save Changes**.
10. On the Java Access Policies page, order the policies according to how you want to evaluate them. Keep in mind that once the system matches the resource requested by the user to a resource in a policy's (or a detailed rule's) Resource list, it performs the specified action and stops processing policies.
11. (Optional) To improve the performance of your Java applications:
    1. Select Enable Java instrumentation caching on the Maintenance > System > Options page of the admin console. This option can improve the performance of downloading Java applications.
    2. After you finish configuring the system, cache your Java applet and access it as end user. This action eliminates the performance hit that occurs through the intermediation engine when the first end user accesses the applet.

Writing a Java Code Signing Resource Policy

Java code signing resource policies specify how the system rewrites Java applets. By default, when the system intermediates a signed Java applet, it re-signs the applet with its own certificate, which is not chained to a standard root certificate. When a user requests an applet that performs potentially high-risk tasks, such as accessing network servers, the user's browser displays a security warning that the root is not a trusted root. To forestall this warning, you can import a code-signing certificate that the system uses to re-sign applets that it intermediates.

When configuring Java code signing resource policies, enter the servers from which you trust applets. You can enter a server IP address or domain name. The system only re-signs applets served by a trusted server. If a user requests an applet from server not on the list, the system does not use the imported production certificates to sign the applet, which means the user is prompted by the browser with a security warning. For Sun JVM users, the system additionally checks that the root CA of the original applet certificate is on its list of trusted root certificate authorities.

To write a Java code signing resource policy:

1. In the admin console, choose **Users > Resource Policies > Web**.
2. If your administrator view is not already configured to show java policies, make the following modifications:
    1. Click the **Customize** button in the upper right corner of the page.
    2. Select the **Java** check box.
    3. Select the **Code-Signing** check box below the **Java** check box.
    4. Click **OK**.
3. Select the **Java > Code-Signing** tab.
4. On the Java Signing Policies page, click **New Policy**.
5. Enter a name to label this policy (required) and description of the policy (optional).
6. In the Resources section, specify the resources to which this policy applies.
7. In the Roles section, specify:

- **Policy applies to ALL roles** - To apply this policy to all users.
- **Policy applies to SELECTED roles** - To apply this policy only to users who are mapped to roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.
- **Policy applies to all roles OTHER THAN those selected below** - To apply this policy to all users except for those who map to the roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.

8. In the Action section, specify:
   - **Resign applets using Code-Signing Certificate** - The uploaded code-signing certificate will be used to sign the Java applets intermediated by the system.
   - **Resign applets using default certificate** - The system re-signs the applet with its own self-signed code signing certificate that is not chained to a standard root certificate.
   - **Use Detailed Rules** - To specify one or more detailed rules for this policy.
9. Click **Save Changes**.
10. On the Java Signing Policies page, order the policies according to how you want to evaluate them. Keep in mind that once the system matches the resource requested by the user to a resource in a policy's (or a detailed rule's) Resource list, it performs the specified action and stops processing policies.

Creating a Selective Rewriting Resource Policy

Selective rewriting resource policies enable you to define a list of hosts for which you want to intermediate content as well as exceptions to this list. By default, the system intermediates all user requests to Web hosts-unless you have configured the system to serve requests to certain hosts using a different mechanism, such as the Secure Application Manager.

Create a selective rewriting policy if you do not want the system to intermediate traffic from web sites that reside outside of the corporate network, such as yahoo.com, or if you do not want the system to intermediate traffic for client/server applications you have deployed as Web resources, such as Microsoft OWA (Outlook Web Access).

To write a selective rewriting resource policy:

1. In the admin console, choose **Users > Resource Policies > Web.**
2. If your administrator view is not already configured to show rewriting policies, make the following modifications:
   1. Click the **Customize** button in the upper right corner of the page.
   2. Select the **Rewriting** check box.
   3. Select the **Selective Rewriting** check box below the **Rewriting** check box.
   4. Click **OK**.
3. Select the **Rewriting > Selective Rewriting** tab.
4. On the Web Rewriting Policies page, click **New Policy**.

5. Enter a name to label this policy (required) and a description of the policy (optional).
6. In the Resources section, specify the resources to which this policy applies.
7. In the Roles section, specify:
   - **Policy applies to ALL roles** - To apply this policy to all users.
   - **Policy applies to SELECTED roles** - To apply this policy only to users who are mapped to roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.
   - **Policy applies to all roles OTHER THAN those selected below** - To apply this policy to all users except for those who map to the roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.
8. In the Action section, specify:
   - **Rewrite content** - The system intermediates all Web content from the resources specified in the Resources list.
   - **Rewrite content as** - The system intermediates all Web content from the resources specified in the Resources list and rewrites the content as if it were the file type specified in the drop-down list. The available options are:
     - **HTML** - Rewrite content as Hypertext Markup Language (HTML)
     - **XML** - Rewrite content as Extensible Markup Language (XML)
     - **Javascript** - Rewrite content as Java scripting language
     - **VBScript** - Rewrite content as Virtual Basic scripting language
     - **CSS** - Rewrite content as Cascading Style Sheets
     - **XSLT** - Rewrite content as XML Style Sheets
     - **Flash** - Rewrite content as Shockwave Flash
     - **DTD** - Rewrite content as Document Type Definitions (DTD)
     - **HTC** - Rewrite content as HTML component

Table 9 summarizes the existing contents that are rewritten for IPv4 and IPv6.

Table 9 Contents Rewritten for IPv4 and IPv6.

Content Type	IPv6 Supported	Source Class
HTML	Yes	DSContentHtmlRewriter/ DSContentHTMLHelpHHCRewriter
JavaScript	Yes	DSContentScriptRewriter/ DSContentScriptRewriter
CSS	Yes	DSContentCssRewriter
XML		DSContentXMLRewriter
MSP		DSContentMSPRewriter
Flash		DSContentSWFRewriter
DTD		DSContentDTDRewriter
Siebel		DSContentSiebelRewriter
PDF		DSContentPDFRewriter
XSL	Yes	DSContentXSLPartialRewriter
Manifest		DSContentManifestRewriter
Java		DSContentJavaRewriter

- **Don't rewrite content: Redirect to target Web server** - The system does not intermediate Web content from the resources specified in the Resources list and automatically redirects the request to the target Web server. This is the default option for all rewrite resource policies that you create. If you select this option, you might want to specify that the system open the unrewritten pages in a new window.

Note: Do not select this option if the specified content needs to access resources inside your corporate network. For instance, if you specify that the system should not rewrite a particular file, and that file calls another file within your network, the user will see an error.

- **Don't rewrite content: Do not redirect to target Web server** - The system retrieves the content from the original Web server, but does not modify it. This is useful in cases where users may not be able to reach the original server, thus disabling redirection. (For example, if the Web server is not accessible from the public internet because it resides behind a firewall.)

  The Don't rewrite content: Do not redirect to target Web server option allows users to download data from network resources via the system, but bypasses the rewriting engine in the process. We recommend you use this feature only when rewriting signed Java applets-not other content types. For other content types such as HTML and Javascript, use the Don't rewrite content: Redirect to target Web server option to download an applet via the system, thus enabling direct connections to network resources.
- **Optimize as long lived resource (no rewrite)** - Some http(s) resources which are long lived, are known to cause high CPU usage. Examples of this kind of resources are:

1. Outlook web access PendingNotificationRequest identified by pattern

   ":/ns=PendingRequest&ev=PendingNotificationRequest"

2. VMware horizon view HTML5 feature's heartbeat request identified by pattern

   ":/system/wts,system/heartbeat"

   These resources can be optimized to use less resources by enabling this option. This option does not work if the resource which is optimized is:

   - Kerberos protected resource
   - Has Web proxy policy configured
   - Resource is accessed through HTTP POST method and SSO is configured.

9. Click **Save Changes**.

On the Web Rewriting Policies page, order the policies according to how you want to evaluate them. Keep in mind that once the system matches the resource requested by the user to a resource in a policy's (or a detailed rule's) Resource list, it performs the specified action and stops processing policies.

Creating a Passthrough Proxy Resource Policy

Passthrough proxy resource policies specify Web applications for which the system performs minimal intermediation. To create a passthrough proxy resource policy, you need to specify two things:

- Which Web application to intermediate with the passthrough proxy
- How the system listens for client requests to the application server

To write a passthrough proxy resource policy:

1. In the admin console, choose **Users > Resource Policies > Web.**
2. If your administrator view is not already configured to show rewriting policies, make the following modifications:
   1. Click the **Customize** button in the upper right corner of the page.
   2. Select the **Rewriting** check box.
   3. Select the **Passthrough Proxy** check box below the **Rewriting** check box.
   4. Click **OK**.
3. Select the **Rewriting > Passthrough Proxy** tab.
4. On the Passthrough Proxy Policies page, click **New Application**.
5. Enter a name to label this policy (required) and a description of the policy (optional).
6. In the **URL** field, specify the application server hostname and the port used to access the application internally. Note that you cannot enter a path in this field.
7. Choose the way in which you want to enable the passthrough proxy feature:

- **Use virtual hostname** - If you choose this option, specify a hostname alias for the application server. When the system receives a client request for the application server hostname alias, it forwards the request to the specified application server port in the URL field.

  If you choose this option, you must also define the name and hostname in the Network Identity section of the System > Network > Internal Port tab. In order to make Sharepoint work successfully through the system, you must select the Override automatic cookie handling check box in Internet Explorer under Tools Internet options > Privacy > Advanced Privacy Settings if the following conditions true:

  - You select the **Use virtual hostname** option during Pass Through Proxy configuration.
  - The virtual hostname that you specify in your Sharepoint configuration is different from the hostname that you configure through the system setup (that is, if the domains are different).

  You enable persistent cookies through the Users > User Roles > Select Role > General > Session Options page of the admin console.
- **Use IVE port** - If you choose this option, specify a unique port in the range 11000-11099. The system listens for client requests to the application server on the specified port and forwards any requests to the application server port specified in the URL field.

8. In the Action section, specify the method to use to intermediate traffic:

   - **Rewrite XML** - If you select this option, the system rewrites URLs contained within XML content. If you disable this option, the system passes the XML content "as is" to the server.
   - **Rewrite external links** - If you select this option, the system rewrites all URLs. If you disable this option, the system rewrites only those URLs that contain a hostname specified in the passthrough proxy policy.
   - **Block cookies from being sent to the browser** - If you select this option, the system blocks cookies destined for the client's browser. The system stores the cookies locally and sends them to applications whenever they are requested.
   - **Host** - Header forwarding-If you select this option, the system passes the hostname as part of the host header instead of the actual host identifier.

     The Host-Header forwarding option is only valid in passthrough proxy Virtual Host mode.

9. Click **Save Changes.**

10. On the Pass-through Proxy Policies page, order the policies according to how you want to evaluate them. Keep in mind that once the system matches the application requested by the user to an application specified in a policy's (or a detailed rule's) Resource list, it performs the specified action and stops processing policies.

11. If you select:

    - **Use virtual hostname,** you must also:

    1. Add an entry for each application server hostname alias in your external DNS that resolves to the system.
    2. Upload a wildcard server certificate to the system (recommended).

- **Use IVE port,** open traffic to the port you specified for the application server in your corporate firewall.

  If your application listens on multiple ports, configure each application port as a separate passthrough proxy entry with a separate port. If you intend to access the server using different hostnames or IP addresses, configure each of those options separately; in this case, you can use the same port.

  External passthrough proxy links that are embedded in a passthrough proxy page may not work. For example, if the bar.company.com page contains a link to foo.company.com and foo.company.com is configured as a host-mode passthrough proxy application, the link to foo.company.com fails. To avoid this, use port-mode passthrough proxy for passthrough proxy links embedded in passthrough proxy applications.

Creating a Custom Header Resource Policy

By default, the rewriting engine only sends selected custom headers to browsers (clients) and backend servers. You can use custom header resource policies, however, to allow or deny custom headers for specific resources.

Note that custom header resource policies do not control standard HTTP headers such as Content-Type.

To write a custom header resource policy:

1. In the admin console, choose **Users > Resource Policies > Web.**
2. If your administrator view is not already configured to show rewriting policies, make the following modifications:
   1. Click the **Customize** button in the upper right corner of the page.
   2. Select the **Rewriting** check box.
   3. Select the **Custom Headers** check box below the **Rewriting** check box.
   4. Click **OK**.
3. Select the **Rewriting > Custom Headers** tab.
4. On the Custom Header Policies page, click **New Policy**.
5. Enter a name to label this policy (required) and a description of the policy (optional).
6. In the Resources section, specify the resources to which this policy applies.
7. In the Roles section, specify:
   - **Policy applies to ALL roles** - To apply this policy to all users.
   - **Policy applies to SELECTED roles** - To apply this policy only to users who are mapped to roles in the Selected roles list. Make sure to add roles to this list from the Available roles list.
   - **Policy applies to all roles OTHER THAN those selected below** - To apply this policy to all users except for those who map to the roles in the Selected roles list. Make sure to add roles to this list from the Available roles list.

8. In the Action section, specify:
   - **Allow Custom Headers** - Select this option to prevent the system from blocking the headers to browsers (clients) and backend servers.
   - **Deny Custom Headers** - Select this option to use the default custom header behavior on the system. When you select this option, the system blocks custom headers for added security.
   - **Use Detailed Rules** - To specify one or more detailed rules for this policy.
9. Click **Save Changes.**
10. On the Web Rewriting Policies page, order the policies according to how you want to evaluate them. Keep in mind that once the system matches the resource requested by the user to a resource in a policy's (or a detailed rule's) Resource list, it performs the specified action and stops processing policies.

Creating an ActiveX Parameter Resource Policy

When the system rewrites a Web page, it does not rewrite the ActiveX controls that are embedded in the Web page. However, you can create resource policies specifying that the system should rewrite the URL and hostname parameters that are passed by the Web page to the Active X controls. To configure these resource policies, you must obtain the following information:

- **Class ID** - Web pages generally use a class ID to embed an ActiveX control. A class ID is a unique, constant string that uniquely identifies an ActiveX control.

  You can determine what an ActiveX object's class ID is using Internet Explorer 6: Select T**ools > Internet Options**, click **Settings**, and then click **View Objects.** Select the ActiveX object, right-click, and select **S**. The ActiveX object's ID is highlighted.
- **Language** - Web pages can use either static or dynamic HTML (that is, by using JavaScript) to embed an Active X control. When a Web page uses static HTML, the system can rewrite the specified ActiveX parameters on the system itself while it intermediates traffic, since all of the required information passes between the user's browser and the application's Web server. When a Web page uses dynamic HTML to embed an ActiveX control, however, the page frequently pulls information from the client and then generates HTML to embed the ActiveX control. Therefore, the system needs to run script in the user's browser in order to obtain the information it needs to rewrite the specified ActiveX parameters.
- **Parameter type** - When configuring the system to rewrite a parameter, you must determine whether the parameter is a URL or hostname. The system does not support any other parameter types.
- **Parameter name** - You must specify the name of the parameter that you want to rewrite. You can find the parameters by searching for the param tag within an object tag. For example, you might find a flash movie embedded in a page using the following code:

  ```
  <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" > <param name="movie" value="mymovie.swf" />
  <param name="quality" value="high" />
  </object>
  ```

  When configuring the corresponding resource policy, you should enter movie in the Parameter name field because movie refers to the URL requires rewriting. Frequently, pages contain multiple param tags, but not all of them require rewriting. In this example, the quality parameter does not require rewriting.

To write an ActiveX parameter rewriting resource policy:

1. In the admin console, choose **Users > Resource Policies > Web.**
2. If your administrator view is not already configured to show rewriting policies, make the following modifications:
    1. Click the **Customize** button in the upper right corner of the page.
    2. Select the **Rewriting** check box.
    3. Select the **ActiveX Parameter Rewriting** check box below the **Rewriting** check box.
    4. Click **OK**.
3. Select the **Rewriting > ActiveX Parameter Rewriting** tab.
4. On the ActiveX Parameter Rewriting Policies page, click **New Policy.**
5. Enter class ID of the ActiveX control that you want to control with the policy (required) and description of the policy (optional).
6. In the Parameters section, specify the ActiveX parameters that you want to control with the policy and the corresponding actions. Possible actions include:
    - **Rewrite URL and response (Static HTML only)** - Rewrite the specified URL parameter on the system. The system also rewrites any response from the Web server requesting the URL. Note that you should select this option if the Web page embeds the ActiveX control using only static HTML.
    - **Rewrite URL and response (Static and dynamic HTML)** - Rewrite the specified URL on the client in addition to rewriting on the system. The system also rewrites any response from the Web server requesting the URL. Note that you should select this option if the Web page embeds the ActiveX control using dynamic HTML.
    - **Rewrite URL (Static HTML only**) - Rewrite the specified URL parameter on the system. Note that you should select this option if the Web page embeds the ActiveX control using only static HTML.
    - **Rewrite URL (Static and dynamic HTML)** - Rewrite the specified URL on the client in addition to rewriting on the system. Note that you should select this option if the Web page embeds the ActiveX control using dynamic HTML.
    - **Rewrite hostname (Static HTML only)** - Rewrite the specified hostname parameter on the system. Note that you should select this option if the Web page embeds the ActiveX control using only static HTML.
    - **Rewrite hostname (Static and dynamic HTML)** - Rewrite the specified hostname on the client in addition to rewriting on the system. Note that you should select this option if the Web page embeds the ActiveX control using dynamic HTML.
    - **Do not rewrite** - Do not rewrite any of the ActiveX component's parameters.
7. Click **Save Changes**.

Restoring the Default ActiveX Resource Policies

The system comes with several predefined resource policies for rewriting the parameters of commonly used ActiveX objects. If you choose to delete any of these policies and then want to restore them later, you can recreate them using the information in Table 10 as a guideline.

Table 10 Predefined Resource Policies

Description	Class ID	Parameter	Action
Citrix NFuse xginen_EmbeddedApp object	238f6f83-b8b4-11cf-8771-00a024541ee3	ICAFile	Rewrite URL and response (Static HTML only)
OrgPlus OrgViewer	DCB98BE9-88EE-4AD0-9790-2B169E8D5BBB	URL	Rewrite URL and response (Static HTML only)
Quickplace	05D96F71-87C6-11D3-9BE4-00902742D6E0	GeneralURL General_ServerName	Rewrite URL and response (Static and dynamic HTML) Rewrite hostname (Static and dynamic HTML)
iNotes Discussion	5BDBA960-6534-11D3-97C7-00500422B550	FullURL	Rewrite URL and response (Static and dynamic HTML)
B20D9D6A-0DEC-4d76-9BEF-175896006B4A	B20D9D6A-0DEC-4d76-9BEF-175896006B4A	Error URL ServerURL	Rewrite URL and response (Static and dynamic HTML) Rewrite hostname (Static and dynamic HTML)
Citrix NFuse Elite	2E687AA8-B276-4910-BBFB-4E412F685379	ServerURL	Rewrite URL and response (Static HTML only)
WebPhotos LEAD	00120000-B1BA-11CE-ABC6-F5B2E79D9E3F	BitmapDataPath	Rewrite URL and response (Static and dynamic HTML)
Shockwave Flash	D27CDB6E-AE6D-11cf-96B8-444553540000	Src Movie	Rewrite URL and response (Static and dynamic HTML) Rewrite URL and response (Static and dynamic HTML)
iNotes Blue	3BFFE033-BF43-11d5-A271-00A024A51325	General_URL General_ServerName	Rewrite URL and response (Static and dynamic HTML) Rewrite hostname (Static and dynamic HTML)

Description	Class ID	Parameter	Action
Tabular Data Control	333C7BC4-460F-11D0-BC04-0080C7055A83	DataURL	Rewrite URL (Static HTML only)
Windows Media Player	6BF52A52-394A-11D3-B153-00C04F79FAA6	URL	Rewrite URL and response (Static HTML only)
FlowPartPlace	4A266B8B-2BB9-47db-9B0E-6226AF6E46FC	URL	Rewrite URL and response (Static HTML only)
HTML Help	adb880a6-d8ff-11cf-9377-00aa003b7a11	Item1	Rewrite URL and response (Static and dynamic HTML)
MS Media Player	22d6f312-b0f6-11d0-94ab-0080c74c7e95	FileName	Rewrite URL and response (Static HTML only)
CSV Files Handler	333c7bc4-460f-11d0-bc04-0080c7055a83	DataURL	Rewrite URL and response (Static HTML only)
Special ActiveX control for Microsoft OWA	D801B381-B81D-47a7-8EC4-EFC111666AC0	mailboxUrl	Rewrite URL and response (Static HTML only)
FlowPartPlace1	639325C9-76C7-4d6c-9B4A-523BAA5B30A8	Url	Rewrite URL and response (Static HTML only)
scriptx print control	5445be81-b796-11d2-b931-002018654e2e	Path	Rewrite URL and response (Static HTML only)
94F40343-2CFD-42A1-A774-4E7E48217AD4	94F40343-2CFD-42A1-A774-4E7E48217AD4	HomeViewURL	Rewrite URL and response (Static HTML only)
Microsoft License Manager	5220cb21-c88d-11cf-b347-00aa00a28331	LPKPath	Rewrite URL and response (Static HTML only)
Domino 7 beta 2 UploadControl	E008A543-CEFB-4559-912F-C27C2B89F13B	General_URL General_ServerName	Rewrite URL and response (Static and dynamic HTML) Rewrite hostname (Static and dynamic HTML)
iNotes	1E2941E3-8E63-11D4-9D5A-00902742D6E0	General_URL General_ServerName	Rewrite URL and response (Static and dynamic HTML) Rewrite hostname (Static and dynamic HTML)

Description	Class ID	Parameter	Action
ActiveCGM	F5D98C43-DB16-11CF-8ECA-0000C0FD59C7	FileName	Rewrite URL and response (Static HTML only)
00130000-B1BA-11CE-ABC6-F5B2E79D9E3F	00130000-B1BA-11CE-ABC6-F5B2E79D9E3F	BitmapDataPath	Rewrite URL and response (Static and dynamic HTML)

Creating Rewriting Filters

Only use the Rewriting Filters tab when instructed to do so by the Pulse Secure Support team.

Writing a Web Compression Resource Policy

The system comes pre-equipped with one Web compression policy (*:*/*) which compresses all applicable Web data. You can enable this policy through the Users > Resource Policies > Web > Compression pages of the admin console.

To write a Web compression resource policy:

1. In the admin console, choose **Users > Resource Policies > Web.**
2. If your administrator view is not already configured to show compression policies, make the following modifications:
    1. Click the **Customize** button in the upper right corner of the page.
    2. Select the **Compression** check box.
    3. Click **OK**.
3. Select the **Compression** tab.
4. On the Web Compression Policies page, click **New Policy**.
5. Enter a name to label this policy (required) and a description of the policy (optional).
6. In the Resources section, specify the URLs to which this policy applies.
7. In the Roles section, specify:
    - **Policy applies to ALL roles** - To apply this policy to all users.
    - **Policy applies to SELECTED roles** - To apply this policy only to users who are mapped to roles in the Selected roles list. Make sure to add roles to this list from the Available roles list.
    - **Policy applies to all roles OTHER THAN those selected below** - To apply this policy to all users except for those who map to the roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.
8. In the Action section, specify:

- **Compress** - Compress the supported content types from the specified resource.
- **Do not compress** - Do not compress the supported content types from the specified resource.
- **Use Detailed Rules** - Select this option to specify one or more detailed rules for this policy.

9. Click **Save Changes**.

Defining an OWA Compression Resource Policy

Due to caching issues with OWA, the system comes with the following built-in resource policies specifying that it should not compress Javascript or CSS files that are routed through OWA:

1. Do Not Compress *:*/exchWeb/controls/*.css (all roles)
2. Do Not Compress *:*/exchWeb/controls/*.js (all roles)
3. Do Not Compress *:*/exchWeb/*/controls/*.css (all roles)
4. Do Not Compress *:*/exchWeb/*/controls/*.js (all roles)

In the last two policies, a wildcard (*) is included in the path to account for different OWA build versions.

Pulse Secure recommends that you do not change the compression resource policies for OWA unless absolutely necessary.

Writing a Web Proxy Resource Policy

Web proxy resource policies specify Web proxy servers for which the system should intermediate content. Note that the system intermediates both forward and backwards proxies, but only enables single sign-on to a proxy when you use these tabs to configure the proxy and thereby specify that you trust it.

To write a Web proxy resource policy:

1. In the admin console, choose **Users > Resource Policies > Web**.
2. If your administrator view is not already configured to show Web proxy policies, make the following modifications:
    1. Click the **Customize** button in the upper right corner of the page.
    2. Select the **Web Proxy** check box.
    3. Select the **Policies** check box below the **Web Proxy** check box.
    4. Click **OK**.
3. Select the **Web Proxy > Policies** tab.
4. On the Web Proxy Policies page, click **New Policy**.
5. Enter a name to label this policy (required) and a description of the policy (optional).
6. In the Resources section, specify the resources to which this policy applies.
7. In the Roles section, specify:

- **Policy applies to ALL roles** - To apply this policy to all users.
- **Policy applies to SELECTED roles** - To apply this policy only to users who are mapped to roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.
- **Policy applies to all roles OTHER THAN those selected below** - To apply this policy to all users except for those who map to the roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.

8. In the Action section, specify:
   - **Access Web resources directly** - Intermediate the user's request to a back-end server and the server's response to the user for requests made to a resource specified in the Resources list.
   - **Access Web resources through a Web proxy** - Specify a Web proxy server in the drop-down list **that you have defined in the Users > Resource Policies > Web > Web Proxy > Servers tab.**
   - **Use Detailed Rules** - To specify one or more detailed rules for this policy.
9. Click **Save Changes**.
10. On the Web Proxy Policies page, order the policies according to how you want to evaluate them. Keep in mind that once the system matches the resource requested by the user to a resource in a policy's (or a detailed rule's) Resource list, it performs the specified action and stops processing policies.

Specifying Web Proxy Servers

You can direct all Web requests to a Web proxy rather than using the system to connect directly to Web servers. This feature can be useful if your network security policy requires this configuration or if you want to use a caching Web proxy to improve performance.

To specify servers for Web proxy resource policies:

1. In the admin console, choose **Users > Resource Policies > Web.**
2. If your administrator view is not already configured to show Web proxy policies, make the following modifications:
   1. Click the **Customize** button in the upper right corner of the page.
   2. Select the **Web Proxy** check box.
   3. Select the **Servers** check box below the **Web Proxy** check box.
   4. Click **OK**.
3. Select the **Web Proxy > Servers** tab.
4. Under Web Proxy Servers, enter the name or IP address of the Web proxy server and the port number at which the proxy server listens, and then click **Add**.
5. Repeat this step to specify additional Web proxy servers.

Writing an HTTP 1.1 Protocol Resource Policy

Protocol resource policies enable or disable HTTP 1.1 protocol support between the system and backend servers. The system supports chunked Transfer-Encoding, gzip and deflate Content-Encoding, connection persistence, and caching headers such as If-Modified-Since, If-None-Match, If-Unmodified-Since and If-Match. The system supports range requests with partial content when you select the Don't rewrite content: Do not redirect to target web server selective rewrite option.

For a detailed description of the HTTP 1.1 protocol, refer to the Hyptertext Transfer Protocol -- HTTP 1.1 specification from the World Wide Web Consortium.

The system only communicates with network servers using HTTP 1.1 if the client also communicates using HTTP 1.1. If the client uses HTTP 1.0, the system communicates with backend servers using HTTP 1.0, regardless of whether or not HTTP 1.1 is enabled.

If you want to use HTTP 1.1 for a specific resource, enable HTTP 1.1 for that policy and ensure that the new policy appears above the default in the list of configured policies. You should add the HTTP 1.1 policy to the top of the policy list because the policy evaluation engine evaluates policies from top to bottom, stopping when it encounters a match.

The system comes with a default policy that disables HTTP 1.1 for all resources. If you want to use HTTP 1.1 for all resources, either redefine the "*:*/*" policy or create a new policy enabling HTTP 1.1 and move it to the top of your policy list. If you delete this default policy (and any other policies that disable HTTP 1.1), the system uses HTTP 1.0 for all resources

To write an HTTP 1.1 protocol resource policy:

1. In the admin console, choose **Users > Resource Policies > Web.**
2. If your administrator view is not already configured to show protocol policies, make the following modifications:
    1. Click the **Customize** button in the upper right corner of the page.
    2. Select the **Protocol** check box.
    3. Click **OK**.
3. Select the **Protocol** tab.
4. On the **Web Protocol Policies** page, click **New Policy.**
5. Enter a name to label this policy (required) and a description of the policy (optional).
6. In the Resources section, specify the URLs to which this policy applies.
7. In the Roles section, specify:
    - **Policy applies to ALL roles** - To apply this policy to all users.
    - **Policy applies to SELECTED roles** - To apply this policy only to users who are mapped to roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.
    - **Policy applies to all roles OTHER THAN those selected below** - To apply this policy to all users except for those who map to the roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.

8. In the Action section, specify:
   - **Disable HTTP 1.1** - Automatically communicate with backend servers via the HTTP 1.0 protocol.
   - **Enable HTTP 1.1** - Automatically communicate with backend servers using the HTTP 1.1 protocol as long as the client also communicates using the HTTP 1.1 protocol.
   - **Use Detailed Rules** - Select this option to specify one or more detailed rules for this policy.
9. Click **Save Changes**.

Creating a Cross Domain Access Policy

The XMLHttpRequest object allows scripts to perform HTTP client functionality, such as submitting form data or loading data from a server. Today's web browsers impose a security restriction on the use of XMLHttpRequest. You are not allowed to make XMLHttpRequests to any server except the server where your web page came from. For example, if both your web application and the data required for that application come from the same web server, then there is no restriction. But, if your web application is on one server and you make a request to a different server, the browser prevents the connection from opening. It is possible to bypass this security, however.

You can create a resource profile that determines whether or not to impose this restriction and to what level. By default, this restriction is bypassed and cross domain access is allowed.

To create a cross domain access policy:

1. In the admin console, choose **Users > Resource Policies > Web.**
2. If your administrator view is not already configured to show cross-domain policies, make the following modifications:
   1. Click the **Customize** button in the upper right corner of the page.
   2. Select the **Rewriting** check box.
   3. Select the **Cross Domain Access** check box below the **Rewriting** check box.
   4. Click **OK**.
3. Select the **Rewriting > Cross Domain Access** tab.
4. On the **Cross Domain Access** page, enter a name to label this policy (required) and a description of the policy (optional).
5. In the Resources section, specify the URLs to which this policy applies.
6. In the Roles section, specify:
   - **Policy applies to ALL roles** - To apply this policy to all users.
   - **Policy applies to SELECTED roles** - To apply this policy only to users who are mapped to roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.
   - **Policy applies to all roles OTHER THAN those selected below** - To apply this policy to all users except for those who map to the roles in the **Selected roles** list. Make sure to add roles to this list from the **Available roles** list.

7. In the Action section, specify:
   - **Allow Cross Domain Access** - To not impose any restriction and allow cross domain access.
   - **Deny XMLHttpRequest Cross Domain Access only** - To deny cross domain access if the XMLHttpRequest object is used in the call.
   - **Deny all Cross Domain Access** - To deny cross domain access regardless of whether or not the XMLHttpRequest object is used in the call.
   - **Use Detailed Rules** - To specify one or more detailed rules for this policy.
8. Click **Save Changes**.

Defining Resource Policies: General Options

When you enable the Web resource policy options described in this section, the system compiles a list of hostnames specified in the Resources field of each Web resource policy. The system then applies the enabled options to this comprehensive list of hostnames.

To specify Web resource options:

1. In the admin console, navigate to **Users > Resource Policies > Web.**
2. If your administrator view is not already configured to show Web options, make the following modifications:
   1. Click the **Customize** button in the upper right corner of the page.
   2. Select the **Options** check box.
   3. Click **OK**.
3. Select the **Options** tab.
4. Select **IP based matching for Hostname based policy resources** if you want the system to look up IP address corresponding to each hostname specified in a Web resource policy. When a user tries to access a server by specifying an IP address rather than the hostname, the system compares the IP to its cached list of IP addresses to determine if a hostname matches an IP. If there is a match, then the system accepts the match as a policy match and applies the action specified for the resource policy.

Note: This option does not apply to hostnames that include wildcards and parameters.

5. Select **Case sensitive matching for the Path and Query string components in Web resources** if you want to require users to enter a case-sensitive URL to a resource. For example, use this option when passing username or password data in a URL.
6. Click **Save Changes**.

Managing Resource Policies: Customizing UI Views

You can control which Web resource policy configuration pages the system displays so that you only have to view those pages that you actually use. Or, if you have a new system installation, you can use these settings to display additional pages (since the system only displays the most commonly used resource policy pages to new users).

To control which Web resource policy configuration pages to display:

1. In the admin guide, choose **Users > Resource Policies > Web > Policy Type**.
2. Click the **Customize** View button in the upper right corner of the console.
3. In the **Customize View** dialog box, specify which Web resource policies you want to display in the admin console. You may manually select individual check boxes, click **All Pages** to display all Web resource policy configuration pages, or click **Common Pages** to display the most commonly used Web resource policy configuration pages. Note you cannot hide the Web Access Policies page.
4. Click **OK**.

Silverlight Support

The system supports Silverlight for rewriting for:

- **Sharepoint 2010** - only the default Silverlight pages on Sharepoint 2010
- **OWA 2010** - including file attachments in e-mails

The system does not support custom XAP packages (custom Silverlight applications that a user can upload to a Sharepoint site).

No configuration steps are required for Silverlight support.

To support Silverlight on Sharepoint 2010, a system-generated rewriting resource policy is added for ActiveX with the following details:

Classid= DFEAF541-F3E1-4C24-ACAC-99C30715084A

Parameter= initParams:mediaSource,previewImageSource

Action= Rewrite URL and response (Static HTML only)

This policy is loaded during both upgrades and new installations. Do not remove this resource policy.

The colon (:) in the above Parameter field means the object tag contains the initParams parameter followed by comma separated name-value pairs.

For example, the above policy works for the following tag:

```
<param name="initParams" value="mediaSource=/silverlight/media/
Wildlife.wmv,previewImageSource=/silverlight/image/VideoPreview.png" />
```

SNI TLS Extension

Server Name Indication (SNI) is an extension to the TLS protocol by which a TLS client indicates which hostname it is attempting to connect to at the start of the handshake process. This allows TLS Web server to present multiple certificates serving multiple secure (HTTPS) websites for the same IP address and TCP port number without requiring to use the same certificate for multiple websites.

SNI is supported only when PCS is acting as a TLS Client. PCS sends SNI server name extension when the Backend Server is accessed using hostname and not IP address. If the backend server has the SNI capability, then it responds with a certificate matching the hostname sent in the SNI server name extension or else it responds with a default certificate.

Note: Some Backend Web Server has Strict SNI Capability which doesn't allow TLS connection when SNI server name extension is not sent in TLS handshake. This behavior will be seen when Backend Server is accessed using IP address by the PCS.

Following are the PCS supported TLS Backend Applications that support and do not support SNI:

Table 11 PCS Supported TLS Backend Applications that Support or Do not Support SNI

Backend Application	Supported
Rewriter	Yes
PTP	Yes
SAML	Yes
JSAM	Yes
PSAM	Yes
Pulse One	Yes
License Server	Yes
CRL	Yes
ActiveSync	Yes
Syslog	Yes
SCEP	Yes
OCSP	No
LDAPS	No
PushConfig	No