SSL Proxy Deployment Guide

Legal Notice

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

www.symantec.com

3/9/2020

Table of Contents

About the Blue Coat SSL Proxy

HTTPS traffic poses a major security risk to enterprises. Because SSL (Secure Sockets Layer) content is encrypted, it cannot be intercepted by normal means. Users can bring in viruses, access forbidden sites, and leak business confidential information over an HTTPS connection, which uses port 443.

Because IT organizations have no visibility into SSL sessions, they are blind to any potential security threats sent over HTTPS.

In addition to the security threat, encrypted traffic makes it difficult for IT to assess bandwidth usage and apply intelligent content control policies to ensure maximum user productivity.

Before the SSL Proxy, the only solution for managing HTTPS traffic was to deny HTTPS altogether or severely limit its usage.

Using SSL proxy can increase control by:

- Distinguishing between SSL and non-SSL traffic on the same port.
- Distinguishing HTTPS from other protocols over SSL.
- Categorizing sites by their SSL server certificate hostname.
- Applying web application control policies for HTTPS traffic.
- Enhancing security through:
    - Server certificate validation, including revocation checks with the help of CRLs and OCSP.
    - Virus scanning and URL filtering of HTTPS content.

The SSL proxy also improves visibility into SSL traffic by creating log files, and enhances performance by caching data.

For more information about the Blue Coat SSL Proxy, see the following topics:

- Digital Certificates and Certificate Authorities
- SSL Proxy Versus HTTPS Reverse Proxy
- What Does SSL Proxy Do?
- When to Use SSL Proxy

About SSL

SSL is the standard protocol for establishing a secure, encrypted link between a remote application server and the client Web browser on the local user's desktop. At the lowest level, SSL is layered on top of TCP/IP. SSL uses the SSL Handshake Protocol to allow the server and client to authenticate each other, and to negotiate the encryption cipher before the application

protocol transmits or receives its first byte of data.

SSL is a proven technology with strong appeal to IT organizations because each secure session link is automatically established on demand using standards-based protocols, encryption techniques, and certificate exchange—all without the need for any IT administration.

The process of setting up the private connection is automatically initiated by the server communicating directly with the browser. The result is a private, encrypted tunnel used to move information between the server and client desktop. When the session is over, the connection is automatically terminated.

However, SSL sessions have become a conduit for a variety of enterprise security threats—including spyware, viruses, worms, phishing, and other malware.

Digital Certificates and Certificate Authorities

Server certificates are used to authenticate the identity of a server. A certificate is an electronic confirmation that the owner of a public key is who he or she really claims to be, and thus holds the private key corresponding to the public key in the certificate. The certificate contains other information, such as its expiration date.

The association between a public key and a particular server occurs by generating a certificate signing request using the server's public key. A Certificate Authority (CA) verifies the identity of the server and generates a signed certificate. The resulting certificate can then be offered by the server to clients who can recognize the CA's signature and trust that the server is who it claims to be. Such use of certificates issued by CAs have become the primary infrastructure for authentication of communications over the Internet.

ProxySG appliances ship with many popular CA certificates already installed. You can review these certificates using the Management Console or the CLI. You can also add certificates for your own internal certificate authorities. CA certificates installed on the ProxySG appliance are used to verify the certificates presented by HTTPS servers and the client certificates presented by browsers (when browsers are configured to do so). In addition, the ProxySG appliance will automatically download an updated `browser-trusted` CA certificate list from Blue Coat Systems, Inc. every seven days by default.

The ProxySG appliance can also be configured to check Certificate Revocation Lists (CRLs, which are provided and maintained by CAs and manually installed on the appliance) for certificates that have been revoked or configured to use the Online Certificate Status Protocol (OCSP) to check certificate revocation status in real time.

.

What Does SSL Proxy Do?

The SSL Proxy can be used to tunnel or intercept HTTPS traffic. The SSL Proxy tunnels all HTTPS traffic by default unless there is an exception, such as a certificate error or a policy denial. In such cases the SSL Proxy intercepts the SSL connection and sends an error page to the user. The SSL Proxy allows interception of HTTPS traffic even when there are no errors. Such interception enables the application of various security policies to HTTPS content.

Some HTTPS traffic, such as financial information, should not be intercepted, but instead passed through in a dedicated tunnel. The following table lists the available functions depending on whether the SSL proxy is used to tunnel or intercept HTTPS traffic:

SSL Proxy Function	Tunneling	Interception
Validate server certificates, including revocation checks using Certificate Revocation Lists (CRLs) and Online Certificate Status Protocol (OCSP).	✓	✓
Check various SSL parameters such as cipher and version.	✓	✓
Log useful information about the HTTPS connection.	✓	✓
Cache HTTPS content.		✓
Apply HTTP-based authentication mechanism.		✓
Scan for viruses and filter specified URLs.		✓
Apply granular policy (such as validating MIME type and filename extension).		✓

The Blue Coat SSL proxy allows you to:

- Determine what HTTPS traffic to intercept through existing policy conditions, such as destination IP address and port number or user or group name. You can also use the hostname in the server certificate to make the intercept versus tunnel decision.
- Validate the server certificate to confirm the identity of the server, and check Certificate Revocation Lists (CRLs) to be sure the server certificate has not been revoked.
- Apply caching, virus scanning, and URL filtering policies to intercepted HTTPS traffic.

SSL Proxy Versus HTTPS Reverse Proxy

Depending on your needs, you can use the ProxySG appliance as either an SSL proxy or an HTTPS reverse proxy. SSL proxy functionality enables the ProxySG appliance to act as forward proxy for HTTPS requests.

- An SSL proxy is a client-side proxy typically used for applying security and performance features such as authentication, URL filtering, and caching.
- An HTTPS reverse proxy is a server-side proxy typically used to offload SSL processing from server to the proxy. Reverse proxies are deployed in proximity to the server. The communication between the HTTPS reverse proxy and server might or might not use SSL. The ProxySG appliance can be used as an HTTPS reverse proxy with the help of the existing HTTPS Reverse Proxy service. Performance and application protection are usually the objectives.

This deployment guide discusses the HTTPS forward proxy. To configure the ProxySG appliance as an HTTPS reverse proxy, refer to the Blue Coat SGOS 6.5 Administration GuideBlue Coat SGOS 6.5 Administration Guide.

Use an SSL Proxy for Privacy, Authentication, and Data Integrity

The SSL proxy can manage the SSL sessions in such a way as to prevent enterprise security threats while at the same time allowing you to determine the level of control.

If the HTTPS traffic contains financial information, you most likely do not want to intercept that traffic.

However, many other kinds of traffic can and should be intercepted by the SSL proxy.

1. "Determine What HTTPS Traffic to Intercept" below
2. "Manage Decrypted Traffic" below

Determine What HTTPS Traffic to Intercept

By default, the SSL proxy only intercepts HTTPS traffic when there is an exception–such as a certificate error–and tunnels all other HTTPS traffic. However, if you want to apply other security measures such as virus scanning or content filtering to SSL traffic you must configure the appliance to intercept SSL traffic. Keep in mind that the encryption and decryption operations required for SSL intercept are resource intensive; therefore you should only intercept the SSL traffic that you believe poses a threat to your network. You specify what SSL traffic to intercept by creating policy rules in the SSL Intercept layer.

To intercept HTTPS traffic for reasons other than error reporting, many existing policy conditions, such as destination IP address and port number or user or group name, can be used.

Additionally, the SSL proxy can use the hostname in the server certificate to make the decision to intercept or tunnel the traffic. The server certificate hostname can be used as-is to make intercept decisions for individual sites, or it can be categorized using any of the various URL databases supported by Blue Coat. Categorization of server certificate hostnames can help place the intercept decision for various sites into a single policy rule.

Recommendations for intercepting traffic include:

- Intercept suspicious Internet sites, particularly those that are categorized as none in the server certificate.
- Intercept sites that provide secure Web-based e-mail, such as Gmail, over HTTPS.
- Intercept social networking traffic

Manage Decrypted Traffic

After the HTTPS connection is intercepted, you can do the following:

- Anti-virus scanning over ICAP.
- URL filtering (on-box and off-box). Blue Coat recommends on-box URL/Content filtering if you use transparent proxy. When the URL is sent off-box for filtering, only the hostname or IP address of the URL (not the full path) is sent for security reasons.
- Filtering based on the server certificate hostname.
- Control web applications
- Caching.

Configure the ProxySG appliance to provide client certificates in policy for HTTPS applications that require browsers to present client certificates to secure Web servers.

If you intercept HTTPS traffic, be aware that local privacy laws might require you to notify the user about interception or obtain consent. You can also use the HTML Notify User object to notify users after interception. You can use consent certificates to obtain consent prior to interception. The HTML Notify User feature is easier; however, the ProxySG appliance must decrypt the first request from the user before it can issue an HTML notification page.

Deploy the ProxySG Appliance with SSL Proxy

Select a topic from the Table of Contents on the left.

What do I Need to Know Before Deploying the SSL Proxy?

The default mode of operation for the SSL proxy is "intercept on exception, tunnel otherwise". Common examples of exceptions for which the SSL Proxy intercepts traffic in this default mode are certificate errors and policy based denials. To intercept HTTPS traffic for purposes other than error reporting (such as antivirus scanning or caching), you must create additional policy.

The SSL proxy can detect the following certificate errors for both intercepted and tunneled traffic:

- The certificate has expired (or is valid at a future date).
- The certificate issuer is untrusted; that is, the ProxySG appliance does not recognize or trust the issuer of the certificate.
- The certificate has been revoked. The ProxySG appliance does a revocation check using Certificate Revocation Lists (CRLs) or Online Certificate Status Protocol (OCSP) to determine if the issuer of the certificate has revoked the certificate.

Recommendations

1. Audit all internal HTTPS servers to verify that they use valid certificates before enabling the SSL Proxy on the ProxySG appliance. This ensures that internal HTTPS sites accessed through the ProxySG appliance do not break after enabling the SSL Proxy.

   In the case of server certificate errors, the SSL proxy intercepts the connection in default mode and sends an exception page to the browser showing the cause of the error. In addition, from the SSL access logs you can monitor the following fields to learn which servers present certificates with errors and what the ProxySG appliance is doing:

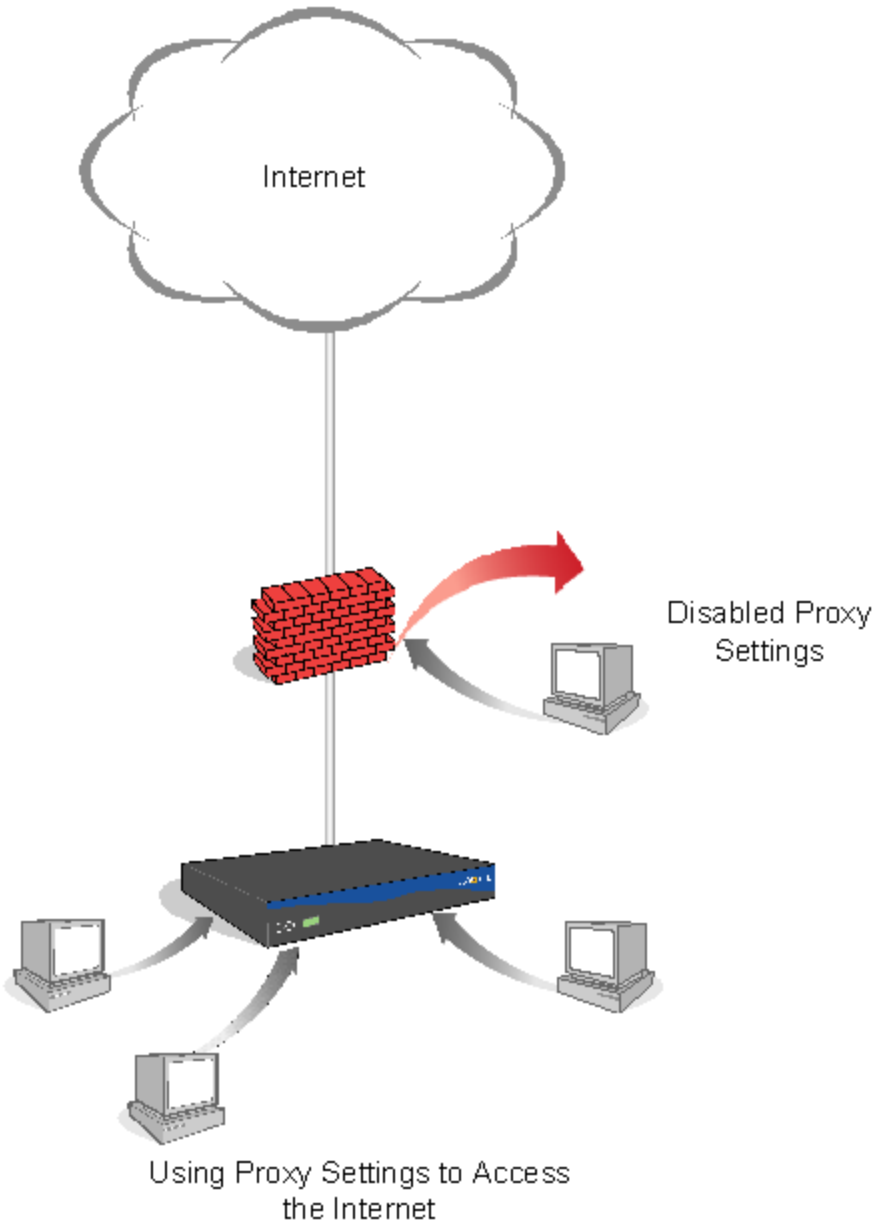
   - `x-rs-certificate-observed-errors`: Shows all the actual error(s) detected with the certificate except hostname-mismatch errors. Detected errors include `untrusted-issuer`, `expired`, and `revoked`.
   - `x-rs-certificate-validate-status`: Shows the certificate validation status after following policy rules. If policy ignores a specific certificate validation error, this field shows the status as `CERT_VALID`, although the certificate presented by a server has the error.
2. Leave the SSL proxy in its default mode where it intercepts the connection in case of errors and reports an exception to the browser. If no errors are found, traffic is tunneled. This allows you to better understand the SSL traffic in your network and helps you write suitable interception policy.

How Do I Set Up SSL Proxy in Explicit Mode?

The SSL Proxy can be used in explicit mode in collaboration with the HTTP Proxy or SOCKS Proxy. You must create an HTTP Proxy service or SOCKS Proxy service and use it as the explicit proxy from desktop browsers. When requests for HTTPS content are sent to either a SOCKS proxy or an HTTP proxy, the proxies can detect the use of the SSL protocol on such connections and enable SSL Proxy functionality. Note that SSL protocol detection should be enabled for the proxy service in use (HTTP or SOCKS).

To create an explicit SSL proxy, complete the following steps:

1. Configure the browser on the desktop to use a proxy or point to a PAC file that points to the proxy.
2. Coordinate with other devices, such as a firewall, to prevent users from accessing the internet without a proxy.
3. Confirm that an HTTP proxy or SOCKS proxy service is present on the desired port and that protocol detection is enabled for that service.
4. Create or import an issuer keyring or use the defaults.
5. Configure SSL proxy rules through VPM.



How do I Deploy SSL Proxy in Transparent Mode?

In a transparent proxy configuration, neither the client (browser) nor the desktop knows that the traffic is being processed by a machine other than the origin content server (OCS). The browser believes it is talking to the OCS, so the request is formatted for the OCS; the proxy determines for itself the destination server based on information in the request, such as the destination IP address in the packet, or the `Host:` header in the request.

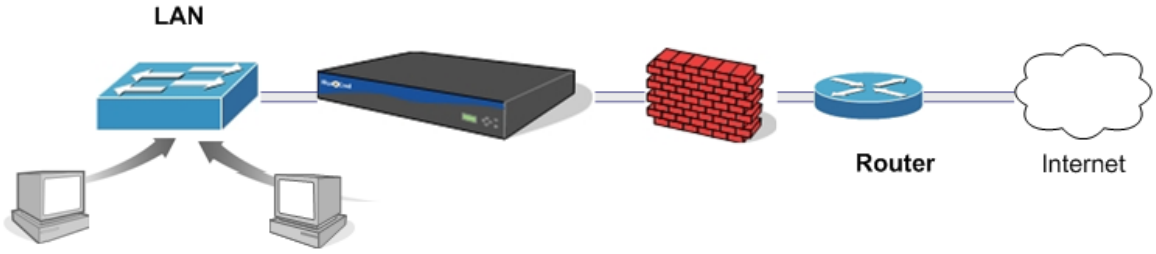
A transparent proxy requires one of the following:

- A hardware bridge
- A WCCP switch
- An L4 switch

If you want to use an L4 switch, WCCP, or an explicit proxy instead of bridging, disable the bridging Pass-Thru card.

Bridging functionality allows ProxySG appliances to be deployed in environments where L4 switches, explicit proxies, and WCCP-capable routers are not feasible options.

A branch office that would take advantage of a bridging configuration is likely to be small (from 20 to 50 users); for example, it might have only one router and one firewall in the network, as shown below.

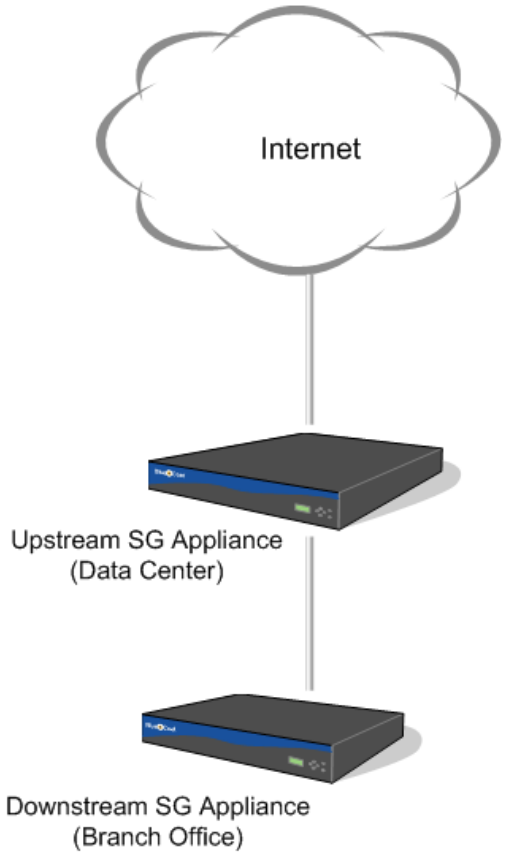


To create a transparent SSL proxy, configure the hardware to use a transparent proxy:

- Create an SSL service on port 443.
- Create or import an issuer keyring or use the defaults.
- Configure SSL proxy through VPM or CPL.

How do I Deploy the SSL Proxy in a Proxy Chain?

The ProxySG appliance at the branch office (the downstream device) uses the ProxySG appliance at the data center (the upstream device) as its forwarding host, allowing SSL Proxy functionality to be enabled on both appliances. A typical SSL proxy chain is shown below.



NOTE For information on using forwarding hosts, refer to the Configuring the Upstream Network Environment chapter in the Blue Coat SGOS 6.5 Administration Guide.

Tips on Setting Up SSL Proxy Chaining Functionality

- The upstream ProxySG appliance is configured as the forwarding host of type “HTTP proxy” for the downstream ProxySG appliance.
- Both proxies have identical SSL related policy; that is, each should make identical decisions in terms of which SSL connections are intercepted and which SSL connections are tunneled.
- The issuer certificate used by the upstream ProxySG appliance to sign emulated certificates should be imported as a

CA certificate on the downstream ProxySG appliance. This ensures that the downstream device can successfully verify emulated certificates presented by the upstream device.

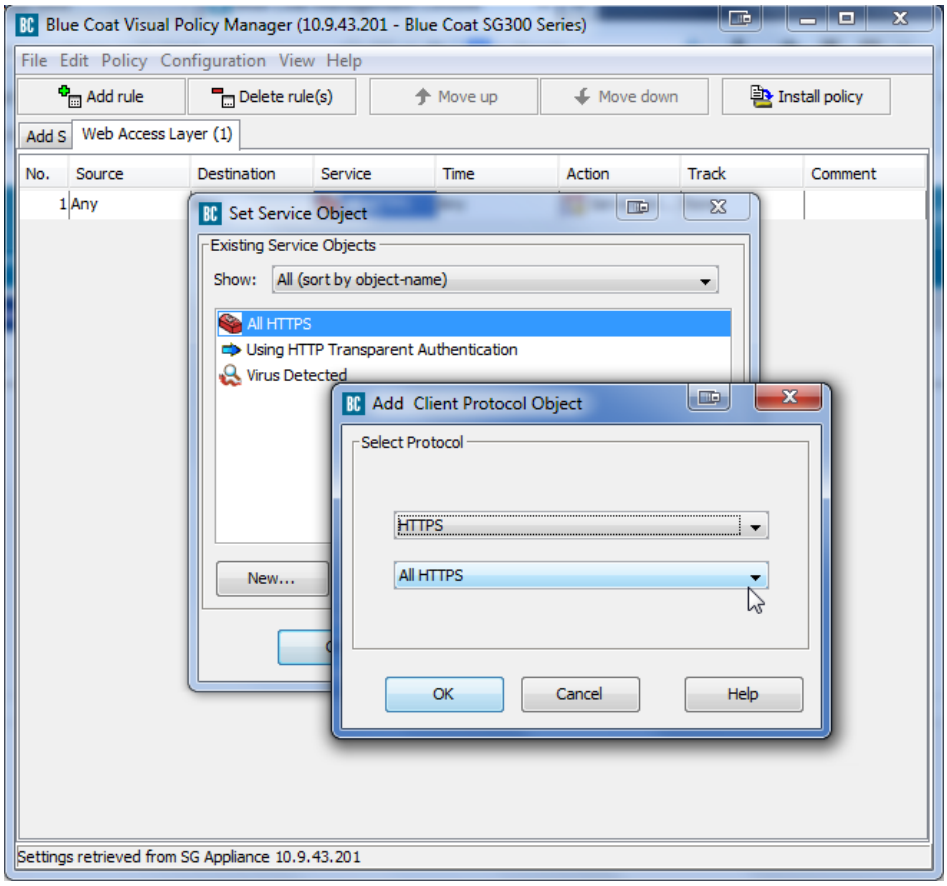
Note that this applies to intercepted SSL connections only. For tunneled SSL connections the downstream ProxySG appliance sees the original server certificate.

Now, when an SSL connection is intercepted at the upstream appliance, the ProxySG appliance emulates the server certificate and presents the emulated server certificate to the downstream ProxySG appliance.

How do I set up HTML notification?

Set up HTML notification only for HTTPS sites:

1. Launch the Visual Policy Manager from **Configuration > Policy > Visual Policy Manager**.
2. Add a new rule to the Web Access layer.
    a. Right click the **Action** field; select **Set**.
    b. Click **New**, then select the **Notify User** object.
    c. Customize the **Notify User object** as needed.
    d. Click **OK**.
    e. Click **OK**.
    f. Right click the **Service** field; select **Set**.
    g. Click **New**, and select the **Client Protocol** object.



h. Select **HTTPS** from the drop-down list in the top field; make sure **ALL HTTPS** is selected from the drop-down list in the lower field.

i. Click **OK**.

3. Click **OK**.

4. Apply the policy by clicking **Install Policy** in the upper-right-hand corner.

How Do I Set Up Certificate Revocation Checks?

Hacked root CA certificates introduce the risk for undetected man-in-the-middle attacks. When users in your network go to a site with a hacked certificate, they may be allured to open fraudulent Web page, leading to possible data leakage or malware infection. To prevent these types of security breaches, the ProxySG appliance can be configured to use Certificate Revocation Lists (CRLs) to verify that certificates presented to it are still valid. A CRL is a list of client and/or server certificates that are no longer valid. The ProxySG appliance can only verify CRLs that are issued by a trusted CA. There are two ways to configure certificate revocation checks on the ProxySG appliance:

Import CRLs onto the ProxySG Appliance

You can manually import CRLs onto the ProxySG appliance. The appliance will then use these local CRLs to check the revocation status of client or server certificates when proxying SSL traffic. Keep in mind that the ProxySG appliance can only verify CRLs issued by a trusted issuer. You can therefore only import CRLs from CAs for which the ProxySG appliance has an issuer certificate. In addition, you can only import one CRL per certificate issuing authority.

If you import a CRL that is expired or that is effective only in the future, a warning will be displayed in the log.

Import a CRL:

1. From the ProxySG appliance Management Console, select **Configuration > SSL > CRLs**.
2. Click **New**. The Add CRL dialog displays.
3. Enter a **Name** for the CRL.
4. Select the method you want to use to import the CRL by selecting a value from the **Install CRL from** drop-down list:
   - **Remote URL**—Select this option to install a CRL file that is located on a remote system that has been posted on a Web server that is accessible from the machine where you are running the Management Console. Click **Install** and then enter the **Installation URL** and then click **Install**. After the CRL is successfully installed, click **OK**.
   - **Local File**—Select this option to install a configuration text file that is located on the system from which you're accessing the Management Console. Click **Install** and then browse to the configuration file and click **Open** to install the selected file.
   - **Text Editor**—Select this option to enter the CRL directly into the Management Console text editor and install the settings. Click **Install**. The Edit and Install the CRL dialog displays. Paste the contents of the CRL file in the text box and then click **Install**.
5. Click **Apply**.

Configure OCSP to Perform Real-Time Certificate Revocation Status Checks

Because each CRL you store on the ProxySG appliance requires memory for storage, you should consider using OCSP if you find that you require a large number of CRLs. With OCSP, you do not need to store the CRLs locally on the appliance. Instead, the ProxySG appliance acts as an OCSP and queries a remote OCSP responder on the intranet or Internet each time it needs to verify a certificate. In addition, OCSP provides the most secure means of checking certificate revocation status because the checks are done in real time.

The OCSP responder sends one of the following certificate statuses back to the ProxySG appliance (the OCSP client):

- **Good**—The certificate is not revoked and valid at the time of the query.
- **Revoked**—The certificate has been revoked either permanently or temporarily.
- **Unknown**—The responder does not know the revocation status of the certificate.

The ProxySG appliance can also cache OCSP responses and has the ability to respect, override or ignore the timestamps related to cacheability in the OCSP response.

For more details on how to use OCSP with the ProxySG appliance, refer to the Blue Coat SGOS 6.5 Administration Guide.

To enable an OCSP revocation check, configure an OCSP responder profile:

1. Select **Configuration > SSL > OCSP**.
2. Click **New** to create a new OCSP responder. The Create OCSP responder dialog displays.
3. Configure the OCSP responder options:
    a. Enter a **Name** for the responder.
    b. **URL**—Indicates the location of the OCSP responder. The ProxySG appliance needs this URL to locate the responder. This location can be obtained from the certificate's Authority Information Access (AIA) extension or from a user-defined configuration. The default is to use the URL from the certificate.
        - **Use URL from certificate**—Select this option if you want the ProxySG appliance to look up the OCSP server location from the subject certificate's AIA extension.
        - **Use URL**—Select this option if you know the location of the designated OCSP responder. Enter a specific responder HTTP or HTTPS URL.
    c. **Issuer CCL**—This option is used to decide which responder is contacted for a given client or server certificate. Typically each certificate issuer uses a designated OCSP responder for all the certificates it issues. The issuer CCL attribute allows the administrator to specify the certificate authorities (issuers) for which the responder in question is the designated responder. This means that when a certificate is signed by one of the CAs in this CCL, the OCSP query for that certificate will be sent to this responder.

        From the drop-down list, select a CA Certificate List (CCL) that contains the CA certificate names for which this is the designated responder. Each CA may only appear in one responder's Issuer CCL. The default is None. Thus, for a given certificate, this CCL is used to determine which responder to use when doing an OCSP check.
    d. **Response CCL**—This attribute is used during verification of OCSP responses. From the drop-down list, select the CCL list you want to use. The default value is **browser-trusted**.
    e. **Device Profile**—This attribute is used when the responder URL is an HTTPS URL. From the drop-down list, select the device profile you want to use when connecting to the OCSP server via SSL. All existing profiles on the ProxySG appliance appear. The device profile is a unique set of SSL cipher-suites, protocols, and keyrings. When the responder URL is HTTPS the ProxySG appliance makes the HTTPS connection with this responder

using its device profile. If the URL is HTTP the device profile is not used. The default value for the device profile attribute is **default**.

f. **Response Cache TTL**—This option indicates how many days an OCSP response is cached on the ProxySG appliance. The default is to use TTL from OCSP response.

- **Use the TTL from OCSP response**—Select this option to use the value of `nextUpdate` timestamp (see section 2.2 of RFC 2560) in the OCSP response. If this timestamp is not set or is in the past, the OCSP response is not cached on the ProxySG appliance. The ProxySG appliance permits a clock skew of up to five minutes with the responder's clock when validating the `nextUpdate` timestamp.
- **Use the TTL**—Enter the length of time (in days) you want the OCSP response to be cached regardless of nextUpdate timestamp in the OCSP response. If TTL is set to 0, the response is not cached.

g. **Enable forwarding**—This option specifies that OCSP requests are to be sent through a forwarding host, if configured. The default is to have forwarding enabled. Based on whether the responder URL is HTTPS or HTTPS the usual forwarding rules apply.

4. Configure the extensions options:

a. **Enable nonce**—To avoid replay attacks, click **Enable nonce**. A *nonce* is a random sequence of 20 bytes places in an OCSP response. The default is to disable the use of a nonce.

b. **Request signing keyring**— This keyring is used when an OCSP request is required to be signed. In this case, the ProxySG appliance includes the certificate chain (minus the root CA) that is associated with this keyring to help the OCSP responder verify the signature.

When a valid keyring is selected then OCSP request signing is enabled. When **None** is selected no request signing occurs.

5. Configure the following **Ignore Settings**:

- **Ignore request failures**—This setting ignores various connection errors. By default, connection errors are not ignored. The following failures are ignored by this setting:
  - The responder's URL is set to from-certificate and the URL in the certificate's AIA extension is neither HTTP or HTTPS, or is not a valid URL.
  - The TCP layer fails to connect with the responder.
  - The responder URL is HTTPS and the initial SSL connection fails with the responder.
  - The TCP connection times out while reading the response from the responder.
  - The TCP connection fails for any reason not already listed.
  - The responder URL is HTTPS and a hostname mismatch error occurs on the responder's certificate.

- The responder URL is HTTPS and an error occurs while analyzing the response. Any other error not caught is covered by the following ignore settings.
- The OCSP responder returns an error message that is described in section 2.3 of RFC 2560. For instance, when an OCSP query is sent to a responder that is not authorized to return an OCSP status for that certificate, the responder returns and `unauthorized` error, that appears as `Responder error (unauthorized)` in event-log of the ProxySG appliance. Enabling this setting causes this error to be ignored as well as other errors described in the RFC.
- The OCSP responder returns a response that is not a basic OCSP response (see section 4.2.1 of RFC 2560).

- **Ignore expired responder certificate**–This setting ignores invalid dates in the responder certificate. By default, invalid responder certificate dates cause the subject certificate verification to fail.
- **Ignore untrusted responder certificate**–This setting ignores the response validation error that occurs when the responder's certificate cannot be trusted. By default, any untrusted certificate failure is an error and causes subject certificate verification to fail.
- **Ignore OCSP signing purpose check**–This setting ignores errors related to the OCSP signing delegation and applies only to Scenarios B and C. (Refer to "Basic OCSP Setup Scenarios" in the Blue Coat SGOS 6.5 Administration Guide.) The errors might occur in one of two ways:
  - Scenario B–The response signer certificate is not delegated for the OCSP signing. The event log records this error as `missing ocspsigning usage`.
  - Scenario C–The root CA does not have the trust setting enabled for the OCSP Signing. The event log records this error as `root ca not trusted`.

    Either of these errors may be ignored by enabling this setting.
- **Ignore unknown revocation status**–Select this setting to ignore unknown revocation status as an error. By default, unknown status is an error and causes subject certificate verification to fail.

6. Click **OK**.
7. Click **Apply**.

SSL Proxy Best Practices

Select a topic from the Table of Contents on the left.

How do I Fix Server Certificate Errors?

The SSL Proxy can detect the following certificate errors:

- `untrusted-issuer`
- `expired`
- `revoked`
- `hostname mismatch` (intercepted connections only)

The most secure way to fix any of these errors is to get a new certificate that does not have the detected error. Many times, however, the sites presenting a bad certificate are not in administrative control. In this case, the SSL Proxy provides a way to ignore certificate errors for certain sites through policy.

Recommendation

If internal HTTPS servers use certificates issued by an internal Certificate Authority (CA), the SSL Proxy flags such certificates with the `untrusted-issuer` error. To prevent such errors, import the internal CA certificate to the ProxySG appliance as a trusted certificate. Do not ignore `untrusted-issuer` errors through policy because an `untrusted-issuer` error means that nothing from the certificate can be trusted.

Do not disable certificate validation globally. Assess ignorable certificate errors on a case-by-case basis, as discussed below.

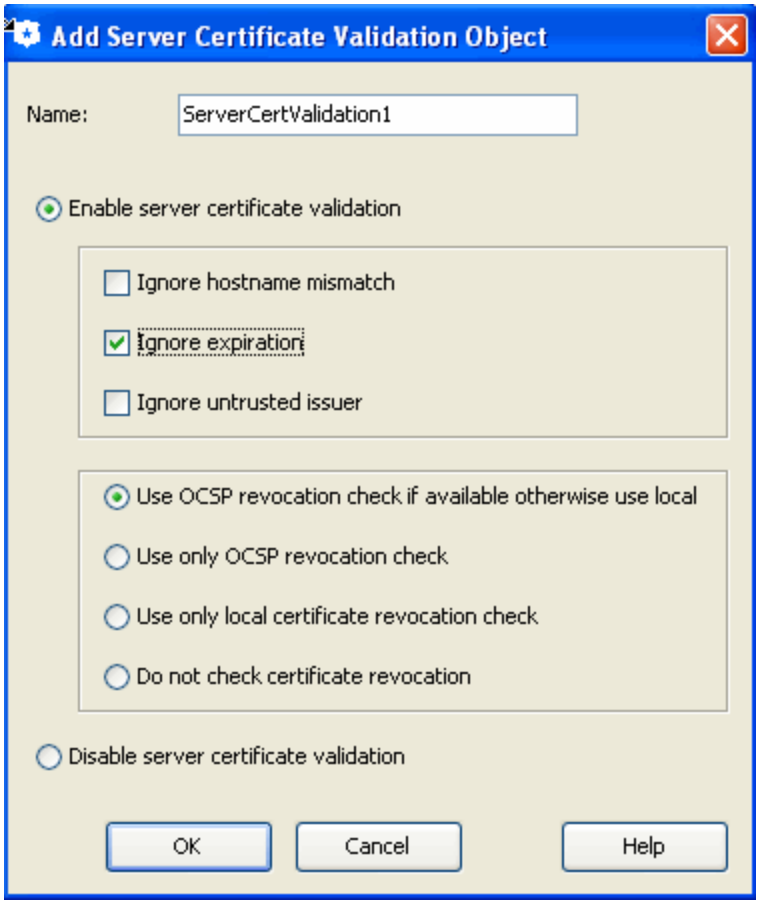
For detailed information on the Visual Policy Manager, refer to the Blue Coat SGOS 6.5 Visual Policy Manager Reference.

To ignore certificate errors for specific sites:

1. Launch the Visual Policy Manager:
   a. Select **Configuration > Policy > Visual Policy Manager**.
   b. Click the **Launch** button.
2. Select **Policy > Add SSL Access Layer** from the menu bar.

   A policy row is added by default when you create a layer.
3. Right-click the **Destination** field; select **Set** to open the Set Destination Object dialog.

4. Click **New**, then:
    a. Add a condition for **Destination Host/Port** or **Server URL**.
    b. Add the IP address and the port or the server URL.
    c. Click **Close**.
    d. Click **OK**.
5. Right click the **Action** field; select **Set**.
6. Click **New** and select **Set Server Certificate Validation**.



7. Select the certificate errors to ignore (for example, **Ignore expiration**), and then click **OK**.
8. To close the Set Action Object dialog, click **OK**.
9. Click **Install Policy**.

Can I configure the ProxySG appliance to tunnel SSL traffic when errors occur?

Yes. The ProxySG appliance has an option named Tunnel on Protocol Error. It applies when non-SSL traffic arrives at the SSL port (443 by default). A common scenario that causes this is having peer-to-peer applications (viz, Skype, BitTorrent, Gnutella, older AOL-IM and eMule) configured to enable port 443 for peer-to-peer traffic without SSL set as the transport protocol. A ProxySG appliance transparently intercepting all 443 traffic cannot process these connections, rendering the application unusable (and the user receives an exception page).

With an explicit proxy deployment, SSL errors during the initial handshake cause the same issue. The following example illustrates this:

- ProxySG appliance is configured to have an explicit HTTP service on port 8080.
- The HTTP service is configured with detect protocol enabled, which hands off SSL traffic to the SSL proxy from an HTTP CONNECT request.

The same applies to an explicit SOCKS proxy deployment with protocol detection enabled or an explicit TCP listener.

1. In the Management Console, select the **Configuration > Proxy Settings > General > General** tab.
2. In the Tunnel on Protocol Error area, select **TCP tunnel requests when a protocol error is detected**.
3. Click **Apply**.

Additional Policy

As a companion piece to this feature, the Visual Policy Manager (VPM) provides the Client Certificate Requested object in the **SSL Intercept Layer > Service** column (the equivalent CPL is `client.certificate.requested {yes | no}`). Use this policy to minimize traffic disruption when the SSL proxy intercepts secure traffic and encounters cases where intercepting further is not an option. For example, the SSL proxy does not have enough information to continue intercepting eMule because to allow the SSL traffic, the OCS requires client certificate authentication. This policy works seamlessly when the SSL proxy is configured to tunnel the secure traffic.

How do I selectively intercept SSL traffic?

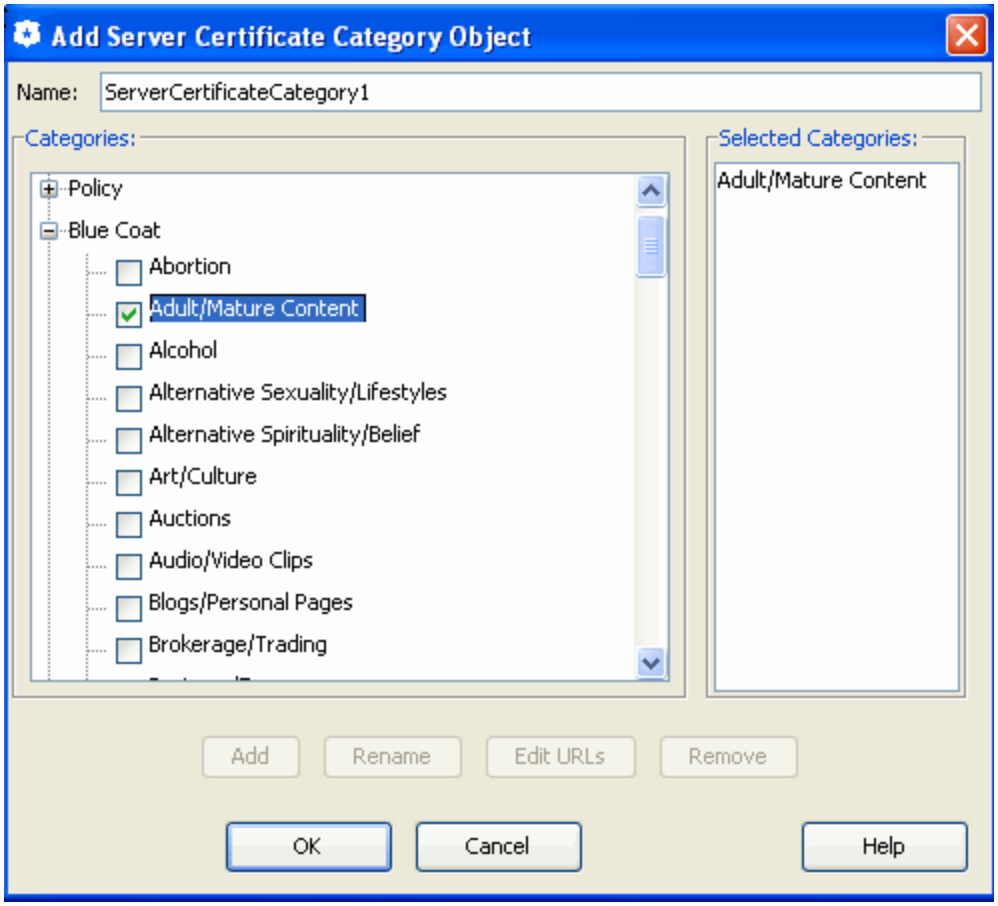
To selectively intercept SSL traffic using the most preferred method, configure a URL filter database.

The following procedure applies to in-path deployments only. For additional information on configuring the SSL proxy, refer to Managing the SSL Proxy chapter in the Blue Coat SGOS 6.5 Administration Guide.

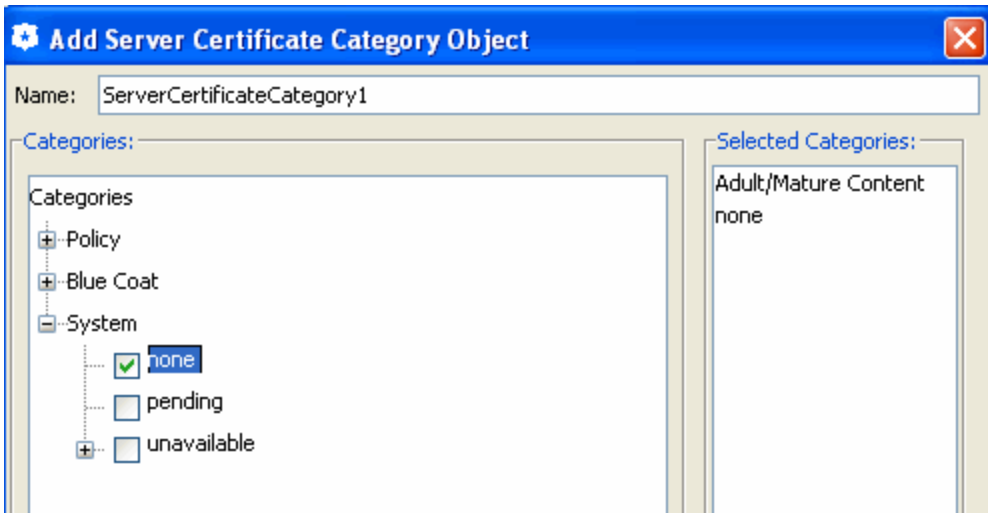
Using the Blue Coat Web Filter as an example, the following steps show how to create a rule that intercepts selected categories.

1. Launch the Visual Policy Manager from **Configuration > Policy > Visual Policy Manager**.
2. Add an SSL Intercept Layer by selecting **Policy > Add SSL Intercept Layer** from the menu bar.

   A policy row is added by default when you create a layer.
3. Right click the **Destination** field; select **Set**, then **New**.
4. Select the **Server Certificate Category** and expand the Blue Coat category.



5. Select the categories to intercept.



6. Expand the **System** category; select **none** to intercept Web sites whose categorization is unknown.

   This allows you to treat unrated sites as suspicious and apply security policies to the data transferred to and from such sites.

7. Click **OK**.
8. Click **OK**.
9. Right click the **Action** field; select **Set**, then **New**.
10. Select **Enable HTTPS Interception**.
11. To allow SSL content to be examined, select:

    a. **Issuer Keyring**: Accept the default keyring or select this option and from the drop-down list select a previously generated keyring. This is the keyring used for signing emulated certificates.

    b. **Hostname**: The hostname you enter here is the hostname in the emulated certificate.

    c. **Splash Text**: The limit is 200 characters. The splash text is added to the emulated certificate as a certificate extension. The splash text is added to the emulated certificate as a certificate extension. For example:

       ```
       Visit http://example.com/https_policy.html
       ```

       To add substitution variables to the splash text, click Edit and select from the list.

    d. **Splash URL**: The splash text is added to the emulated certificate as a certificate extension.

       The SSL splash can be caused by such occurrences as when a browser receives a server certificate signed by an unknown CA, or a host mismatch.

Not all browsers display the splash text and splash URL correctly.

12. Click **OK**.
13. Click **OK**.
14. Apply the policy by clicking **Install Policy** in the upper-right-hand corner.

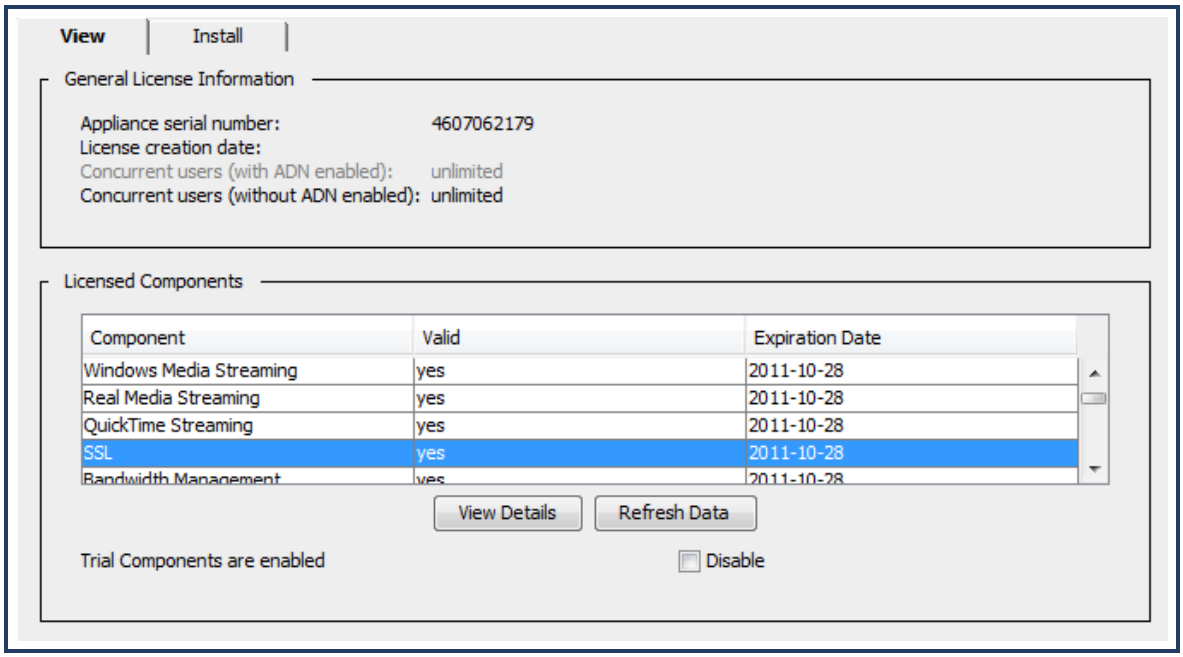
Can I Select Which SSL Traffic to Intercept Based on Authentication Credentials?

By default, the SSL proxy only intercepts HTTPS traffic when there is an exception–such as a certificate error–and tunnels all other HTTPS traffic. However, if you want to apply other security measures such as virus scanning or content filtering to SSL traffic you must configure the appliance to intercept SSL traffic. Keep in mind that the encryption and decryption operations required for SSL intercept are resource intensive; therefore you should only intercept the SSL traffic that you believe poses a threat to your network. You specify what SSL traffic to intercept by creating policy rules in the SSL Intercept layer.

One way to define which SSL traffic to intercept is to restrict intercept based on user or group membership as follows:

1. Make sure you have an SSL license.

   Although someProxySG appliance models include an SSL license, other models require that you purchase and install an add-on license. To see whether you have a valid SSL license, launch theProxySG appliance Management Console and select **Maintenance > Licensing > View**. You should see an entry for **SSL** in the list of licensed components.



2. Set up the issuer keyring and CA certificate list (CCL) to allow the ProxySG appliance to emulate server certificates.

When the ProxySG appliance intercepts HTTPS traffic, it establishes two separate SSL connections: one between the client and the appliance and one between the appliance and the OCS. In order to establish the SSL connection with the client and to enable it to decrypt the data, the appliance emulates the OCS certificate, making itself (the ProxySG appliance) the certificate issuer. To enable this behavior you must:

a. Determine which keyring to use to emulate OCS certificates. You can:

- Use any of the built-in keyrings that include both a certificate and a key pair (such as **default**).
- Create a new keyring and generate a self-signed certificate.
- Use a local CA (such as Microsoft Certificate Services) to act as your Root CA and use it to generate a subordinate CA certificate for the ProxySG appliance.

b. Designate the keyring you have chosen as the **Issuer Keyring** within the SSL proxy configuration (**Configuration > Proxy Settings > SSL Proxy**).

c. (Optional) If the issuer keyring contains a certificate generated by a CA that the client browsers do not trust (such as a self-signed certificate), you must download the ProxySG appliance certificate as a CA certificate to client browsers to prevent the browsers from displaying a pop-up warning to the users when they browse to an HTTPS site. To simplify this process, email the URL that corresponds to the issuer certificate to your end users (**Statistics > Advanced > SSL > Download a ProxySG Certificate as a CA certificate**).

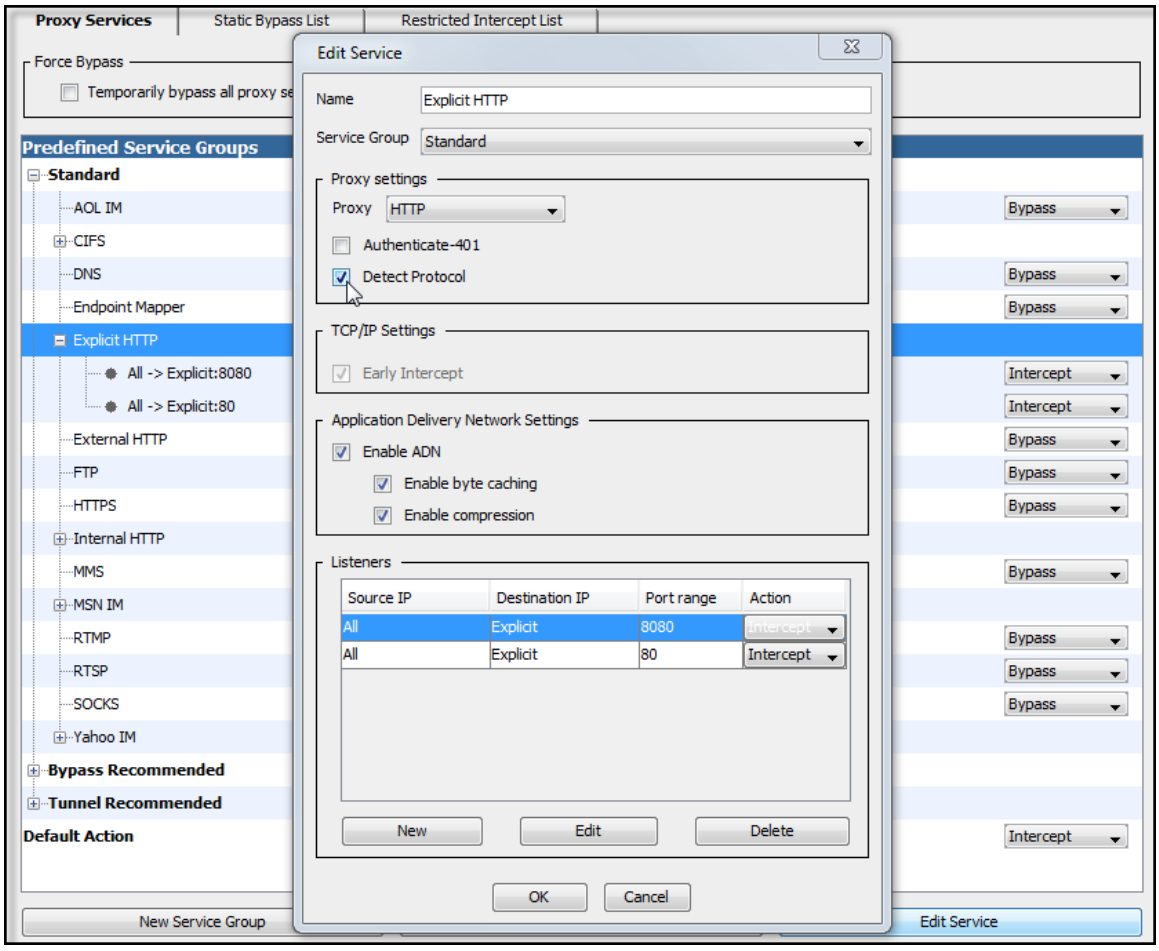
3. Create the authentication realm for authenticating the users/groups you will be using to enforce your SSL intercept policy.

4. Set up the proxy services as necessary to intercept HTTPS traffic.

   The way you do this depends on your deployment type:

   - **Explicit proxy**

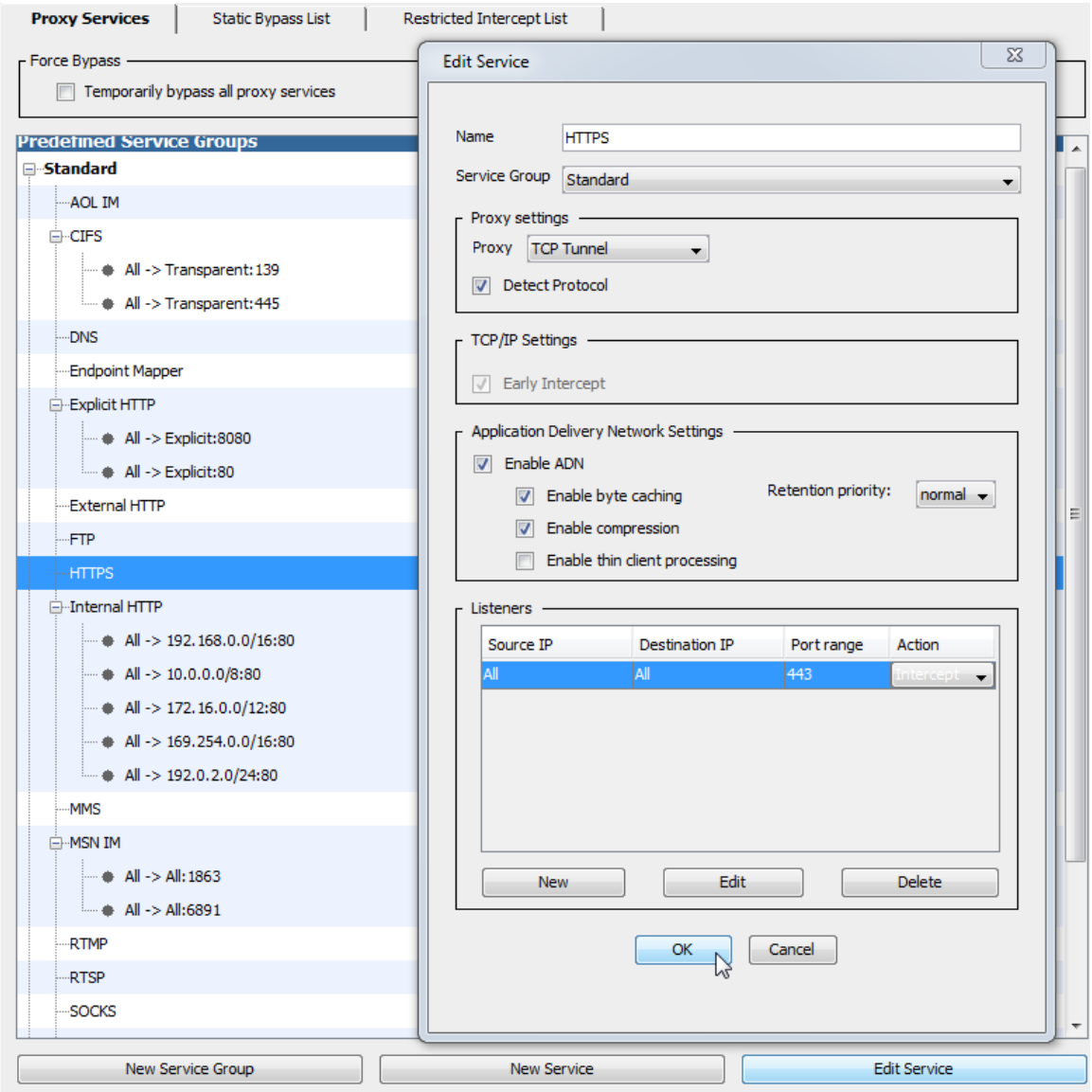
     Configure the Explicit HTTP Proxy service or the SOCKS Proxy service to intercept all explicit traffic (usually on ports 80 and 8080 for HTTP or port 1080 for SOCKS) and enable the **Detect Protocol** attribute. This allows the HTTP or SOCKS proxy to detect HTTPS traffic and hand it off to the SSL Proxy.



   - **Transparent proxy**

     Edit the HTTPS service to use the TCP Tunnel **Proxy** to intercept traffic on port 443. You must also enable the

Detect Protocol attribute. This allows the TCP tunnel proxy to detect HTTPS traffic and hand it off to the SSL Proxy.



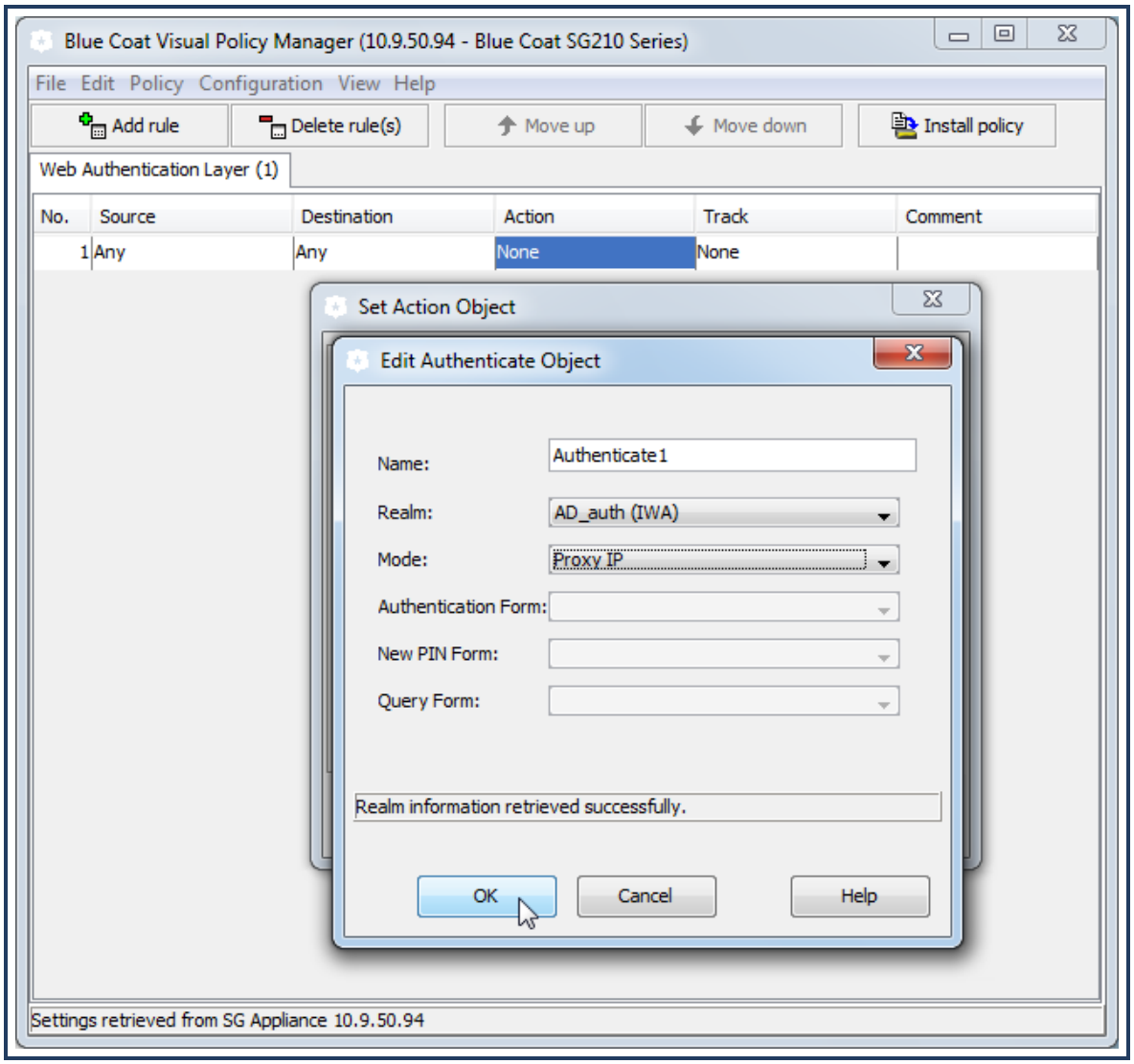
In this specific deployment where you want to make SSL intercept decisions based on user or group, you must use the TCP Tunnel proxy rather than the SSL proxy because the SSL proxy does not authenticate until after it has determined whether to intercept traffic. You can use SSL proxy when making intercept decisions that do not require access to user or group information. However, in this case you would also need to edit each proxy service to define which HTTPS traffic should be intercepted by each.

5. Create a Web Authentication layer to enable authentication using the realm you created.

   a. Launch the Visual Policy Manager (VPM) by selecting **Configuration > Policy > Visual Policy Manager > Launch**.

   b. Select **Policy > Add Web Authentication Layer**.

   c. Create the rule to enable authentication using the authentication realm you created in Step 3. Keep in mind that for transparent proxy, when you configure the **Authenticate** object in the **Action** field, you must select an authentication **Mode** that uses an IP surrogate (**Origin IP Redirect** or **Form IP Redirect** for transparent proxy or **Proxy IP** for explicit proxy).



The SSL proxy can only make a decision to intercept HTTPS traffic based on user if it can associate the IP address in a client request with a username. However, because it does not support transparent authentication, it can only do this if the appliance has an existing IP surrogate credential for the client. This means that unless the client has recently authenticated using a protocol that supports transparent authentication, such as HTTP, the appliance will not be able to make SSL intercept decisions based on user.

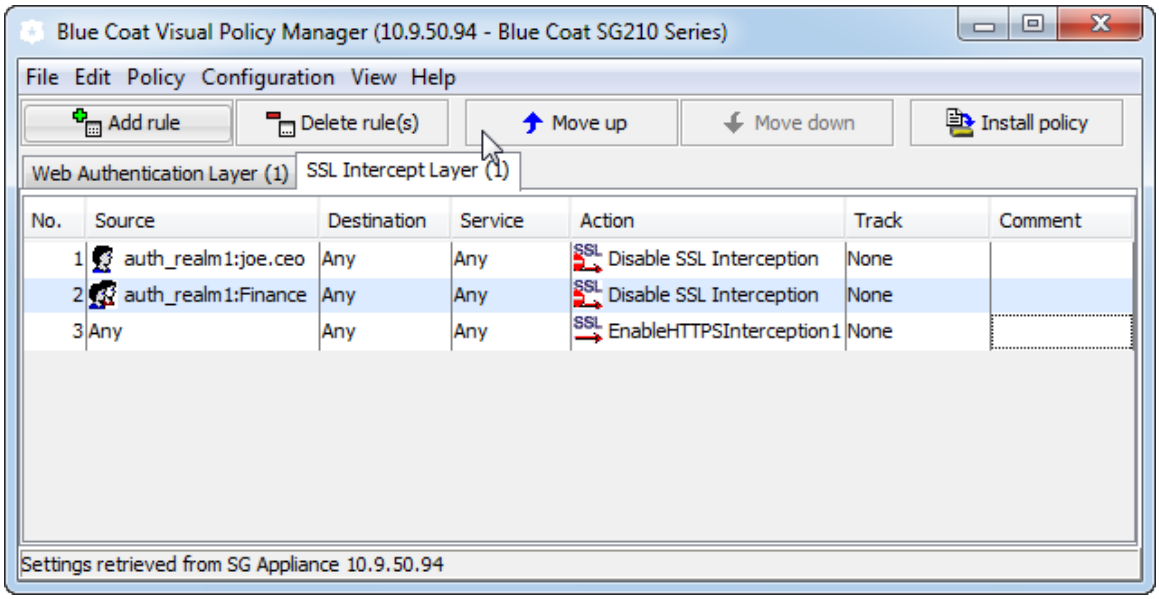
6. Create an SSL Intercept layer.

   a. In the first rule, set the **Source** to the user or group you want to use to determine whether to intercept SSL traffic. For example, if you want all traffic from users in the Finance group to bypass SSL intercept, you would select the **Group** object and set the value to Finance.

   b. Specify how to handle traffic that matches the conditions you set in the **Source** field by selecting **Action**. For example, to bypass SSL interception for the users in the Finance group, you would select **Disable SSL Interception**.

   c. Create any additional user- and/or group-based rules by setting the **Source** and **Action** as specified in steps a and b. For example, you might create a second rule that disables SSL intercept for the CEO. In this case, you would set the select a **User** object in the source field and specify the CEO's username and **Disable SSL Interception** as the **Action**.

   d. Create a final rule that specifies what to do with remaining SSL traffic that does not match any of the rules you defined. For example, to intercept any SSL traffic that does not match the previous rules, create a rule that sets the **Action** to **Enable HTTPS Interception** and leaves the other fields set to the default values. In this example, this would mean that SSL traffic from users in the finance department or from the CEO will not be intercepted but all other SSL traffic will be intercepted.



7. (Explicit proxy only) Set up each client Web browser to use the ProxySG appliance as its proxy server.

   Typically, the browser proxy configuration requires the IP address or hostname of the ProxySG appliance and the port on which the ProxySG appliance will listen for traffic. The default port is 8080. The required hostname format (that is, whether you must provide a fully qualified DNS hostname or a short hostname) depends on the DNS configuration on the client systems.

8. Verify your SSL intercept policy.

   To do this, browse to a secure site—such as `https://www.facebook.com`—and check to see that the traffic was intercepted (or not intercepted) as expected based on your policy. There are two ways you can do this:

   - **From the client browser**– Check to see that the certificate used for the SSL session is the certificate issued to the ProxySG appliance rather than the certificate issued to the OCS. For example, if you are using the ProxySG appliance Default keyring for SSL traffic, the organization name of the certificate will have a `ProxySG:` appliance prefix.
   - **From the appliance**–If you have access logging enabled, you can look in the SSL access log the for the corresponding HTTPS requests in the logs.

Can the ProxySG appliance distribute issuer certificates to client desktops?

When the SSL Proxy intercepts an SSL connection, it presents an emulated server certificate to the client browser. The client browser issues a security pop-up to the end-user because the browser does not trust the issuer used by the ProxySG appliance. This pop-up does not occur if the issuer certificate used by SSL Proxy is imported as a trusted root in the client browser's certificate store.

The ProxySG appliance makes all configured certificates available for download via its management console. You can ask end users to download the issuer certificate through the browser and install it as a trusted CA in their browser of choice. This eliminates the certificate popup for emulated certificates.

Download a certificate through Internet Explorer

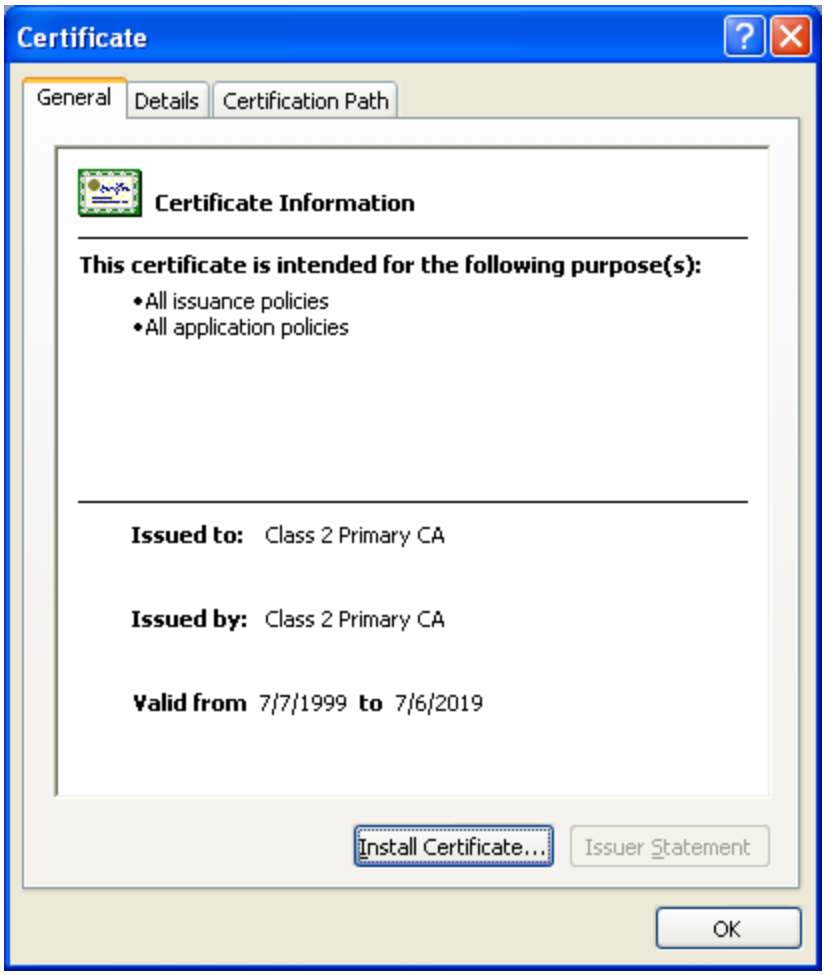
E-mail the console URL corresponding to the issuer certificate to end users so that the end-user can install the issuer certificate as a trusted CA.

1. Select **Statistics > Advanced**.
2. From the **SSL** section, click **Download a ProxySG Appliance Certificate as a CA Certificate**; the list of certificates on the system display.

3. Click a certificate (it need not be associated with a keyring); the File Download Security Warning dialog displays asking what you want to do with the file.
4. Click **Save**. When the Save As dialog displays, click **Save**; the file downloads.

   - or -

   Click **Open** to view the Certificate properties; the Certificate dialog displays.
5. Click the **Install Certificate** button to launch the Certificate Import Wizard.



6. Verify that the **Automatically select the certificate store based on the type of certificate** option is enabled before completing the wizard. The wizard announces when the certificate is imported.
7. (Optional) To view the installed certificate, go to Internet Explorer, and select **Tools > Internet Options > Contents > Certificates**, and open either the **Intermediate Certification Authorities** tab or the **Trusted Root Certification Authorities** tab, depending on the certificate you downloaded.

Download a certificate through Firefox

TIP E-mail the console URL corresponding to the issuer certificate to end users so that the end-user can install the issuer certificate as a trusted CA.

1. Go to **Statistics > Advanced**.
2. Select **SSL**.
3. Click **Download a ProxySG Appliance Certificate as a CA Certificate**; the list of certificates on the system display.
4. Click a certificate (it need not be associated with a keyring); the **Download Certificate** dialog displays.

5. Enable the options as needed. View the certificate before trusting it for any purpose.
6. Click **OK**; close the Advanced Statistics dialog.

How do I Create a Web Page to Explicitly Warn Users of Invalid Certificates?

In addition to browser warnings, how do I create a web page to explicitly warn users of invalid certificates and allow them the choice to ignore the error and continue to the content?

Some servers may have invalid certificates, which trigger warnings from browsers for instances such as self-signed certificates (untrusted issuer), expired certificates, and hostname mismatches with the certificate. Users' connected to these sites through the ProxySG appliance with the SSL proxy enabled can receive an additional error page explaining the reason why users could not access the page.

See also "Present Untrusted Certificates to a Browser" on page 48.

Recommendation

Present a warning message to users that allows them to connect to the HTTPS site by clicking on a link. This requires two components: policy and modified exception pages.

If you are redirecting the request to an object that is being served by the ProxySG appliance (such as a PAC/WPAD file) and you also have a policy in place that requires a response from an upstream server (such as a scan decision from a ProxyAV), the following policy will not work.

You must:

- Ensure SSL traffic is in intercept mode:

  In the VPM, create an SSL Intercept Layer; intercept only the URLs you want to apply to the Certificate Not Valid policy.

- Modify the built-in exceptions:
  - `ssl_domain_invalid`
  - `ssl_server_cert_expired`
  - `ssl_server_cert_untrusted_issuer`

- **Add the Certificate Not Valid Exception in your local policy.**

```
(exception.ssl_domain_invalid
(contact)
(details "Your request contacted a host which presented a certificate with a Common Name
that did not match the domain requested.")
(format <<--eof--
Your request contacted a host which presented a certificate with a Common Name that did not
match the domain requested.
<br>
<br>
<form method="get" action="$(url)">
<input type="submit" style="width:400;height:24;"
value="Click here if you have a legitimate reason to access this site"></form>
<br>
--eof--
)
(help "This is typically caused by a Web Site presenting an incorrect or invalid
certificate, but could be because of a configuration error.")
(summary "SSL Certificate Hostname Mismatch")
(http
(code "409")
(contact)
(details)
(format)
(help)
(summary)
)
)
(exception.ssl_server_cert_expired
(contact)
```

```
(details "Your request contacted a host which presented an
expired or Invalid certificate")
(format <<--eof--
Your request contacted a host which presented an expired or Invalid certificate.
<br>
<br>
<form method="get" action="$(url)">
<input type="submit" style="width:400;height:24;"
value="Click here if you have a legitimate reason to access this site"></form>
<br>
--eof--
)
(help "This is typically caused by a Web Site presenting an incorrect or invalid
certificate, but could be because of a configuration error. ")
(summary "Expired SSL Server Certificate")
(http
(code "503")
(contact)
(details)
(format)
(help)
(summary)
)
)
(exception.ssl_server_cert_untrusted_issuer
(contact)
(details "Your request contacted a host which presented a certificate signed by an
untrusted issuer.")
(format <<--eof--
Your request contacted a host which presented a certificate signed by an untrusted issuer.
<br>
<br>
<form method="get" action="$(url)">
<input type="submit" style="width:400;height:24;"
value="Click here if you have a legitimate reason to access this site"></form>
<br>
--eof--
)
(help "This is typically caused by a Web Site presenting an incorrect or invalid
certificate, but could be because of a configuration error.")
(summary "Untrusted SSL Server Certificate")
(http
(code "503")
(contact)
(details)
(format)
(help)
(summary)
)
)
```

- **Install the Certificate Not Valid Policy.**

```
<exception> condition=sslexception
action.mycookie(yes)
<proxy>
condition=sslallow request.header.cookie="sslallow"\
action.rewtohttps(yes)
request.header.cookie="sslallow" action.red(yes)
<ssl>
condition=sslallow server.certificate.validate.ignore(all)
define action mycookie
set(exception.response.header.set-cookie, "sslallow$(url.cookie_domain)")
```

```
end
define action rewtohttps
rewrite(url,"(.*)\?xyzallow","$(1)")
end
define action red
redirect(307,"(.*)","$(1)?xyzallow")
end
define condition sslallow
url.substring="xyzallow"
end
define condition sslexception
exception.id=ssl_server_cert_untrusted_issuer
exception.id=ssl_server_cert_expired
exception.id=ssl_domain_invalid
end
```

- For an invalid certificate, the `xyzallow` value is appended to the URL after the user clicks **Accept**. This is expected behavior.

- Only use the `request_redirect` gesture for objects returned from the ProxySG appliance itself, such as the `accelerated_pac_base.pac`. Do not apply to redirects for objects from an OCS (Origin Content Server). Continue to use `redirect( )` for redirects to an OCS.

How do I Protect End-user Privacy and Prevent Accidental Exposure of Sensitive Information When Intercepting SSL Traffic?

For intercepted SSL traffic, potentially sensitive information is available in cleartext in the following locations:

If ICAP scanning is enabled for intercepted HTTPS traffic, such data is sent without encryption to the ICAP server.

You can log request and response headers containing sensitive information to the access log and event log.

If you use an off-box URL filtering solution, part of the URL may be sent in cleartext to the URL database service point. Note that such a service point can be located on the Internet.

Intercepted HTTPS content that is cacheable is also available on the disk in the clear.

Recommendation

Take the following measures to prevent accidental exposure of sensitive information:

- Enable secure ICAP scanning between the ICAP server and the ProxySG appliance. If you use plain ICAP scanning for intercepted HTTPS content, ensure that the network link between the ProxySG appliance and the ICAP server cannot be snooped.
- Use care in determining which sites to intercept. Avoid intercepting well-known banking and financial sites. On-box URL databases and server certificate categories can be used in determining which sites to intercept.

- Use on-box URL databases, such as Blue Coat Web Filter or a third-party content filtering vendor, to avoid transmitting URLs in clear text. Further, if you use WebPulse services for dynamic categorization of Web requests, enable the **Use secure connections** checkbox to ensure that all communication with the WebPulse service occurs in secure mode.
- Implement HTML notification for intercepted sites to inform end-users that their HTTPS traffic will be monitored and that they can opt-out if they do not want their traffic to be intercepted. HTML notification is also helpful if a site is accidentally intercepted.
- Do not log URL or header information for intercepted HTTPS traffic. (By default, the SSL log does not log this information.)

The ProxySG appliance allows you to set up notification two ways, HTML notification and client consent certificates.

For information on Client Consent Certificates, refer to the Managing X.509 Certificates chapter in the Blue Coat SGOS 6.5 Administration Guide.

How do I Allow Non-SSL traffic on Port 443 to Certain Servers?

In my transparent proxy deployment. How do I allow non-SSL traffic on port 443 to certain servers while still enabling the SSL Proxy for other port 443 traffic? Some legitimate applications, such as the SOCKS-based VPN clients from Aventail and Permeo, use port 443 to communicate with the VPN gateway. However, the protocol they use is not SSL. An SSL service created on port 443 that transparently terminates such TCP connections breaks these applications. That is because the SSL service enforces the use of the SSL protocol.

Administrators can allow such SOCKS-based VPN tunnels to a few trusted partner sites.

For information on creating TCP-tunnel services, refer to the Services chapter in the Blue Coat SGOS 6.5 Administration Guide.

1. Create a transparent TCP-tunnel service on port 443. Do not create an SSL service on port 443.
2. Specify the list of servers that can use port 443 for non-SSL protocols in policy:

```
define condition Trusted_non_ssl_servers
      url.address=1.1.1.1
      url.address=2.2.2.2
      end condition Trusted_non_ssl_servers
```

3. Write a <proxy> layer that forces all other traffic on port 443 to use the SSL protocol:

```
<proxy>
      proxy.port=443 condition =! Trusted_non_ssl_servers force_protocol(ssl)
```

These rules ensures that port 443 connections to the list of trusted servers are tunneled without intervention while all other port 443 connections use the SSL protocol.

Windows Updates Fail When I Use the SSL Proxy to Intercept All SSL Connections

SSL connections for Windows updates should always be tunneled. For example:

```
<ssl-intercept>
server.certificate.hostname=update.microsoft.com \
      ssl.forward_proxy(no)
      ssl.forward_proxy(https)
```

The same policy can be created in VPM using the **SSL Intercept Layer**, the **Server Certificate Object**, and the **SSL Forward Proxy object**.

Note that you only need to do this if the policy intercepts everything. If you do selective interception, as recommended, this issue does not arise.

Can I Use CA Hierarchy for Certificate Emulation?

Some enterprises have a well-defined CA Certificate hierarchy (chain) in place. Consider the hypothetical example of Clothing-Max, a retail clothing outlet with 150 stores in the U.S. and Canada.

The Clothing-Max Root CA Certificate is at the top of the hierarchy and has issued a CA certificate for the Clothing-Max IT department. In turn, the IT department issues a CA certificate for the IT security team.

If the security team wants to deploy the SSL proxy using its CA certificate as the issuer for emulated certificates, the team will import this certificate and its private key on the ProxySG appliance. The intermediate CA must be imported in two places on the ProxySG appliance:

- Under the Keyrings panel where both the private key and the certificate are stored.
- Under CA Certificates panel on ProxySG appliance. This second step ensures that the SSL Proxy chains the intermediate CA certificate along with the emulated certificate.

The ProxySG appliance now signs the emulated certificates using the private key of the Clothing-Max IT Security Team CA Certificate. The certificate chain for an emulated certificate for a Clothing-Max server will be:

Root CA	Intermediate CAs	Emulated Certificate
Clothing-Max	Clothing-Max IT	
	Clothing-Max IT Security Team	Clothing-Max Server

In this case, the browser does not show a security pop-up if it is able to verify all certificates in the certificate hierarchy.

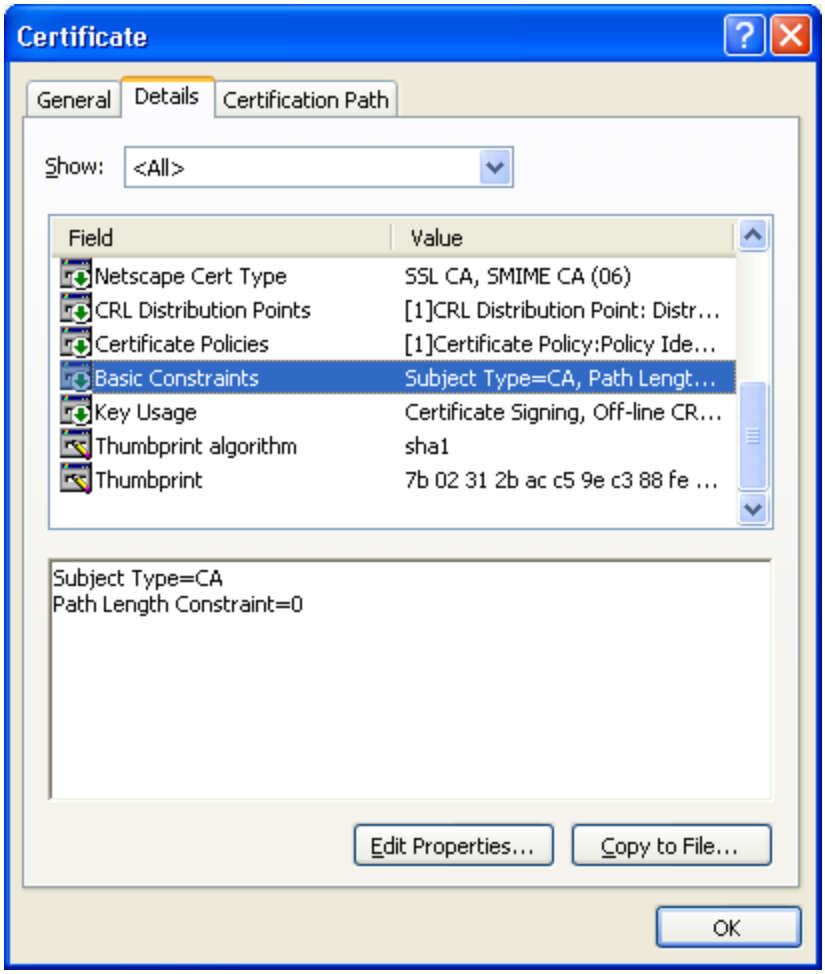
If you use Internet Explorer, additional requirements are necessary on the intermediate CA certificates in the certificate chain.

Intermediate CA certificates must contain the basic constraints certificate extension with the Subject Type set to CA. Also, if your intermediate CA certificate has a KeyUsage extension, make sure it has the “Certificate Signing” attribute present.

Root CA certificates are exempt from this requirement:

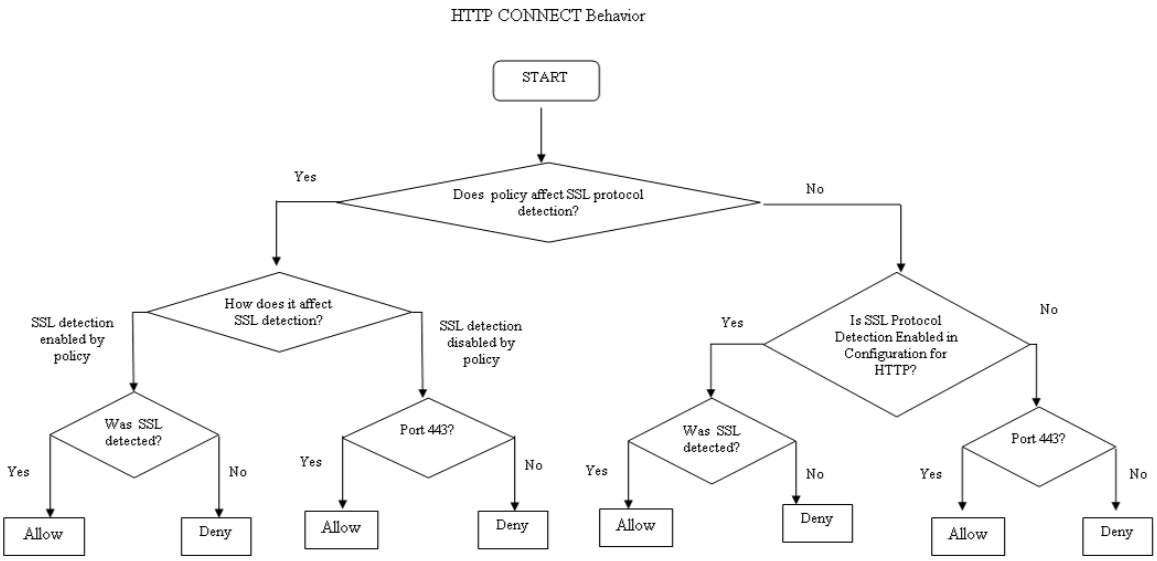
Root CA	Intermediate CA	Intermediate CA
Clothing-Max	Clothing-Max IT	Clothing-Max IT Security Team

The illustration below shows a Verisign Class 2 Intermediate Certificate Basic Constraints Extension.



How does the HTTP Proxy Securely Process the CONNECT Method?

It follows the rules outlined in the following flow chart:



How do I Authenticate Intercepted Transparent SSL Traffic and Add the Username to the Access Log?

If you are using explicit proxy, you do not need to perform any additional tasks to authenticate and log proxied HTTPS traffic. However, for transparent proxy, you must complete the following steps:

1. Create an authentication realm, such as LDAP, IWA, or RADIUS, based on the environment.

   Management Console Location: **Configuration > Authentication > Realm_Name**

2. As part of realm authentication, change the virtual URL for the realm to https://hostname:444. The hostname, which must not be a fully qualified domain name, must resolve to the IP address of the ProxySG appliance.

   Management Console Location: **Configuration > Authentication > Realm_Name > General**

3. Make sure that transparent proxy is set to the session cookie method. This is the default.

   Management Console Location: **Configuration > Authentication > Transparent Proxy**

4. An HTTPS (SSL) Service already exists on the system by default. Modify the default HTTPS service, if needed, to intercept traffic on port 443.

   Management Console Location: **Configuration > Services > Proxy Services > Encrypted Service Group > HTTPS > Edit Service**

5. Create an HTTPS reverse proxy on the ProxySG appliance so that connections to the virtual URL are intercepted by

the ProxySG appliance.

Management Console Location: **Configuration > Services > Proxy Services > Reverse Proxy Service Group > New Service**

6. (Optional) If you use a TCP-tunnel service on 443 in transparent mode instead of the SSL service, enable protocol detection on the TCP-tunnel service.

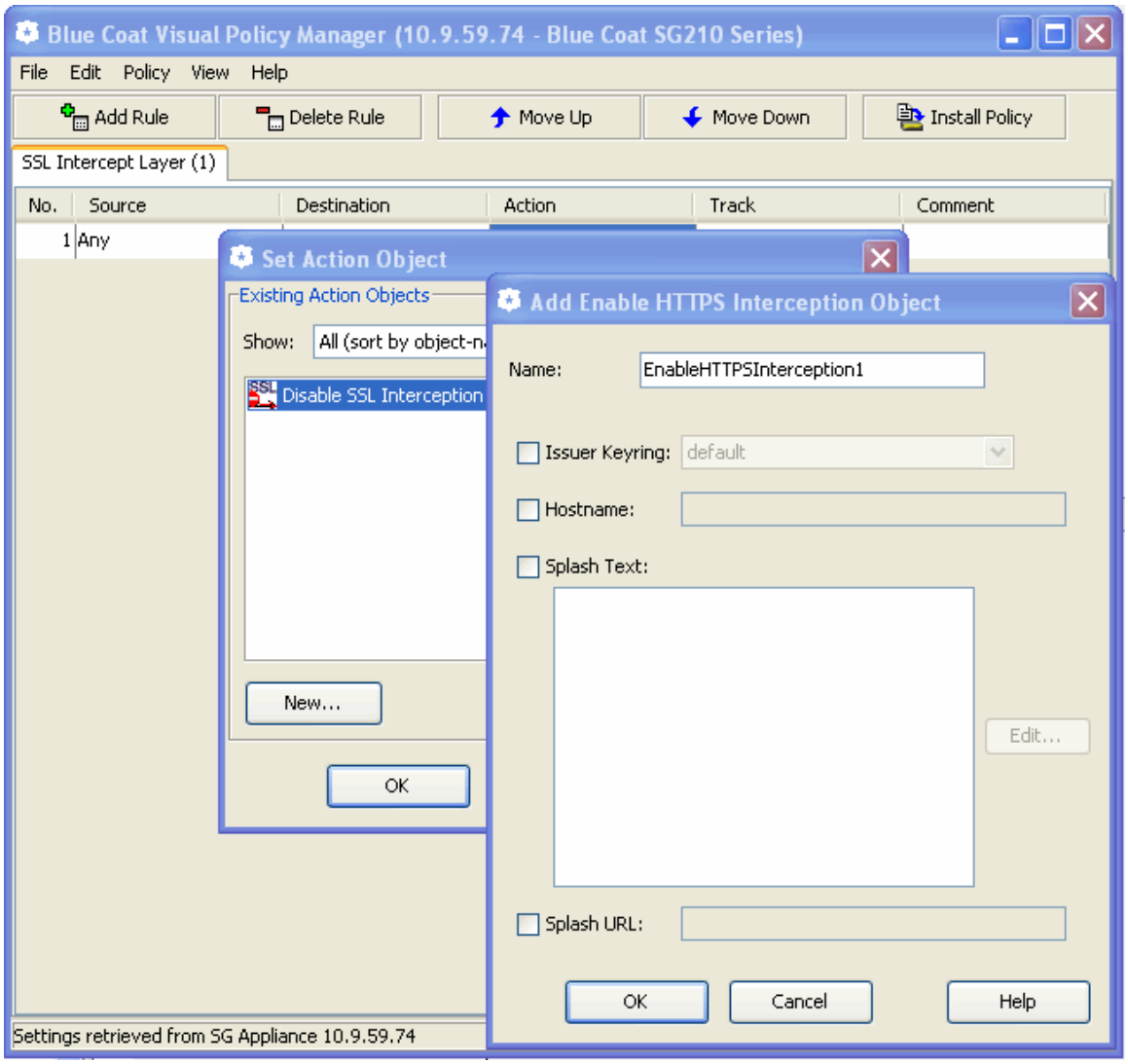
   Management Console Location: **Configuration > Services > Proxy Services > Other Service Group > New Service**

7. Write policy to enable SSL Proxy functionality using the Visual Policy Manager. For an example of policy using CPL, see "Sample CPL for Transparent Authentication" below.

   a. From the Management Console, launch the Visual Policy Manager: **Configuration > Policy > Visual Policy Manager > Launch**.

   b. From the **Policy** menu, select **Add SSL Intercept Layer**.

   c. Right-click the **Action** cell and select **Set**. Click **New** and select **Enable HTTPS Interception** on the **Add SSL Interception Object** window.

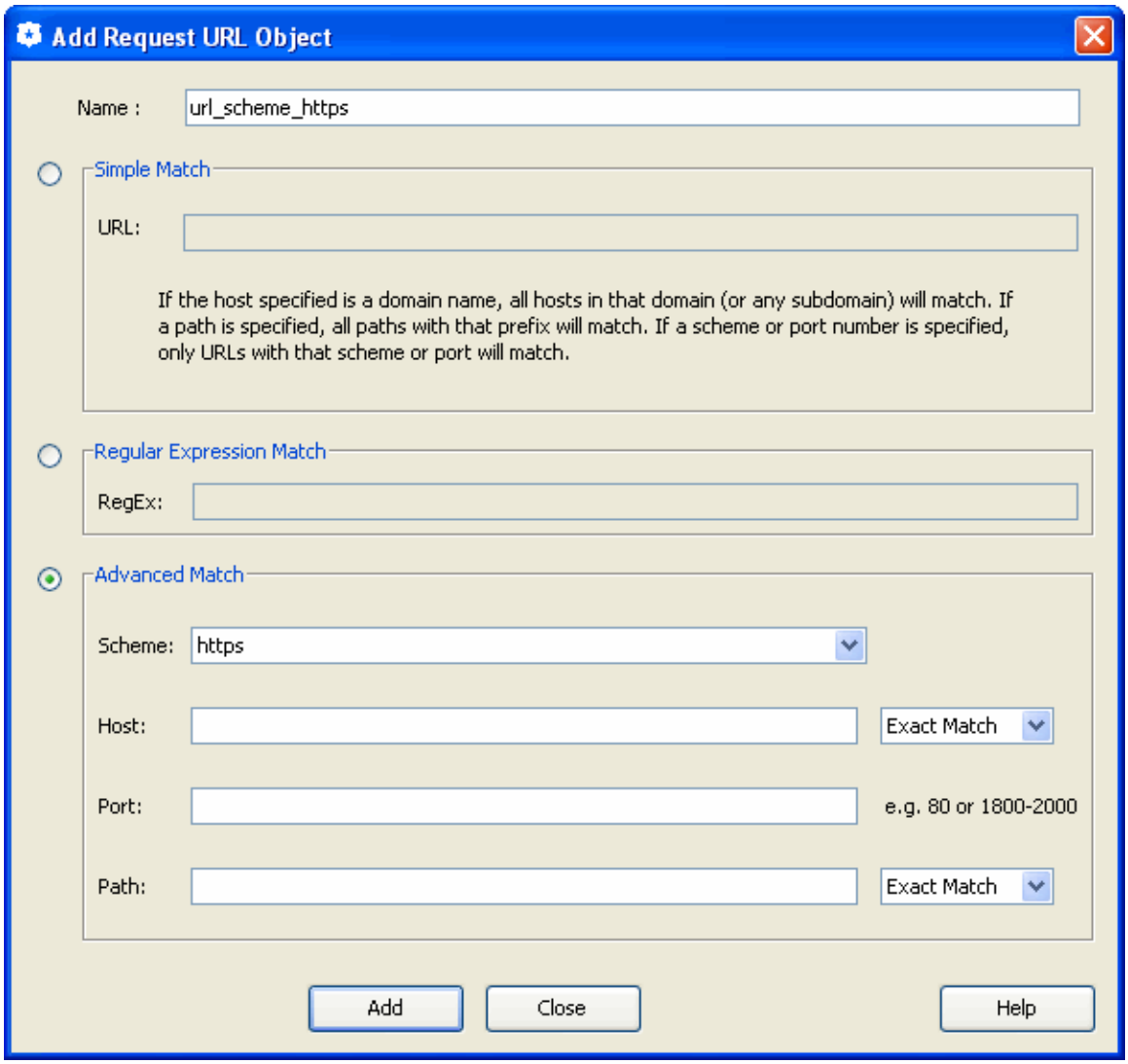


d. Click **OK** to add the interception object, and then click **OK** to close the Set Action Object dialog.

e. From the **Policy** menu, select **Add Web Authentication Layer**. You will be creating a combined object containing two Request URL objects: HTTPS, and HTTP.

f. Right-click the **Destination** cell and select **Set**. Click **New** and select **Request URL**.

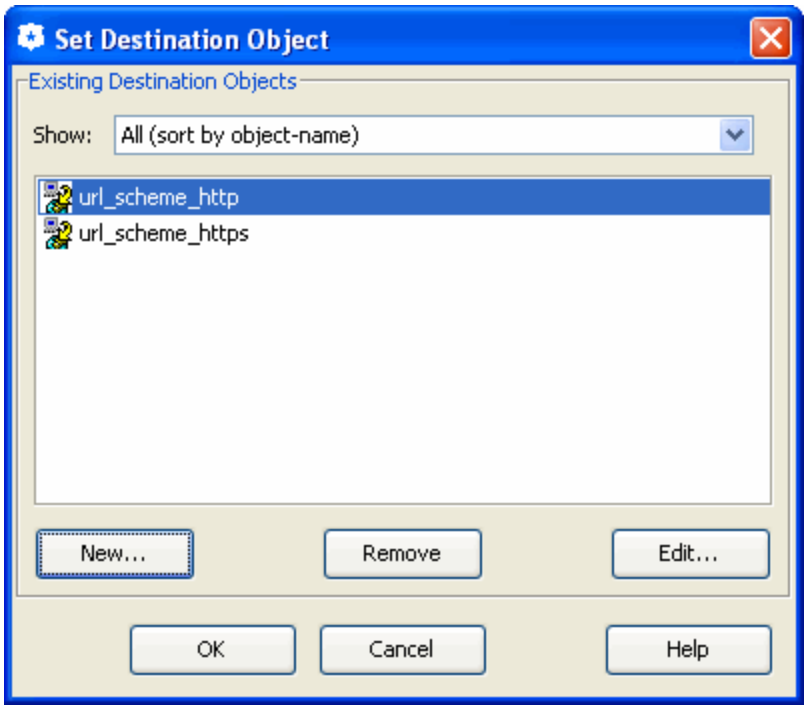
g. Select **Advanced Match**. In the **Name** field, type **url_scheme_https**. From the **Scheme** drop-down list, select **https**.



h. Click **Add** to add the Request URL Object for HTTPS.

i. Now, repeat the same procedure to add a Request URL Object for HTTP. Select **Advanced Match**. In the **Name** field, enter **url_scheme_http**. From the **Scheme** drop-down list, select **http**.

j. Click **Add** and then **Close**. You should now see both **url_scheme_http** and **url_scheme_https** in the Set Destination Object dialog.



k. Click **New** and select **Combined Destination Object**.

l. Shift-click to select both **url_scheme_http** and **url_scheme_https** and then click **Add**.

m. Click **OK** to add the Combined Destination Object to the **Web Access Layer**, and then click **OK** to close the Set Destination Object dialog.

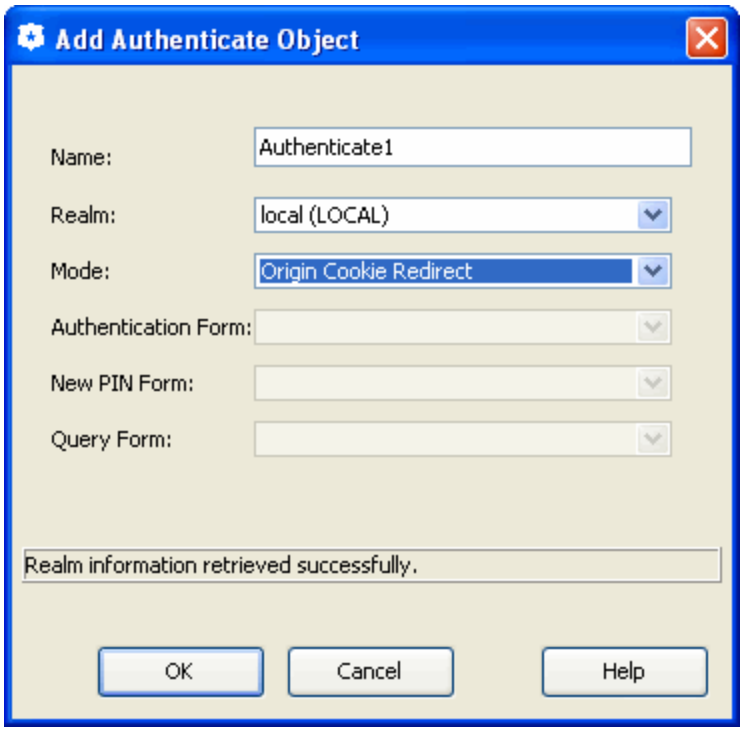
n. Right-click the **Action** cell and select **Set**.

o. Click **New** and select **Authenticate**.

p. Specify the desired **Realm** and select a redirect **Mode**:

- **origin-cookie-redirect**, where the client is redirected to a virtual URL to be authenticated, and cookies are used as the surrogate credential.
- **origin-ip-redirect** (unsecure), where the client is redirected to a virtual URL to be authenticated, and the client ip_address is used as a surrogate credential.
- **form-cookie-redirect**, where a form is presented to collect the user's credentials. The user is redirected to the authentication virtual URL before the form is presented.
- **form-ip-redirect** (unsecure), where the user is redirected to the authentication virtual URL before the form is presented.

In this example, the local realm is set to **Origin-Cookie-Redirect**.



q. Click **OK** to add the Authenticate Object, and then click **OK** to close the Set Destination Object dialog.

r. In the Visual Policy Manager, click **Install Policy**.

8. Add the access log field `cs-username` to the SSL access log format.

   Management Console Location: **Configuration > Access Logging > Formats > SSL > Edit**

Sample CPL for Transparent Authentication

You can also use the CPL to write policy. In this example, realm name is called local and the authentication mode is `origin-cookie-redirect`:

```
<ssl-intercept>
  ssl.forward_proxy(https)

<Proxy>
  authenticate(local) authenticate.mode(origin-cookie- redirect)

;Definitions
define condition client_protocol
      client.protocol=https
      client.protocol=http
end
```

How can I enable LDAP over SSL with a third-party certification authority?

For instructions on configuring an LDAP server to accept LDAP queries over SSL, see the following article on the Microsoft Support site:

How to enable LDAP over SSL with a third-party certification authority
http://support.microsoft.com/kb/321051

How do I Warn Users About Websites with Untrusted Certificates?

Preserve Untrusted Certificate Issuer allows the ProxySG appliance to present the browser with a certificate that is signed by its untrusted issuer keyring. The browser displays certificate information to the user, and lets the user accept the security risk of an untrusted certificate and proceed to the website.

The default-untrusted keyring has been added to the ProxySG appliance to use with the Preserve Untrusted Certificate Issuer feature. The default-untrusted keyring should not be added to any trusted CA lists.

This only applies to SSL forward proxy transactions with HTTPS interception enabled.

To display a warning to users about untrusted certificates on website, you must complete the following tasks:

1. "Present Untrusted Certificates to a Browser" below
2. "Define Behavior in the Visual Policy Manager (VPM)" below, or "Define Behavior in Content Policy Language (CPL)" on the facing page.

Present Untrusted Certificates to a Browser

Configure the ProxySG appliance to act as a certificate authority and present a certificate signed by a specific keyring for all traffic. The default is the `default-untrusted` keyring.

1. From the Management Console, select **Configuration > Proxy Settings > SSL Proxy**.
2. To have the ProxySG appliance act as a Certificate Authority (CA) and present the browser with an untrusted certificate, select **Preserve untrusted certificate issuer**.
3. From the **Untrusted Issuer Keyring** drop-down, select the keyring you want to use to sign untrusted server certificates.
4. Click **Apply**.

Define Behavior in the Visual Policy Manager (VPM)

Override the ProxySG appliance Management Console settings for specific traffic, to specify whether the users should be

prompted when a certificate that has not been signed by a trusted Certificate Authority is encountered.

In the SSL Intercept Layer, add one of the following Actions:

- **Do not Preserve Untrusted Issuer**

  If an OCS presents a certificate to the ProxySG appliance that is not signed by a trusted Certificate Authority (CA), the ProxySG appliance either sends an error message to the browser, or ignores the error and processes the request, based on the configuration of the Server Certificate Validation object.

- **Preserve Untrusted Issuer**

  If an OCS presents a certificate to the ProxySG appliance that is not signed by a trusted Certificate Authority (CA), the ProxySG appliance acts as a CA and presents the browser with an untrusted certificate. A warning message is displayed to the user, and they can decide to ignore the warning and visit the website or cancel the request.

- **Use Default Setting for Preserve Untrusted Issuer**

  The **Preserve untrusted certificate issuer** configuration setting in the ProxySG Management Console is used to determine whether or not untrusted certificate issuer should be preserved for a connection. This is the default behavior.

Define Behavior in Content Policy Language (CPL)

Include the following syntax in policy to specify the behavior of the ProxySG appliance when users encounter a website with an untrusted certificate:

```
ssl.forward_proxy.preserve_untrusted(auto|yes|no)
```

where:

- **auto** – Uses the Preserve untrusted certificate issuer configuration setting in the ProxySG appliance Management Console to determine whether untrusted certificate issuer should be preserved for a connection. This is the default.
- **yes** – Preserve untrusted certificate issuer is enabled for the connection.
- **no** – Preserve untrusted certificate issuer is disabled for the connection.

For example, to use the enable using the preserve untrusted certificate issuer, use the following syntax:

```
<ssl-intercept>
  ssl.forward_proxy.preserve_untrusted(yes)
```

How do I Provide Client Certificates in Policy?

Sometimes, when a user navigates to a secured Web address in a browser, the server hosting the site requests a certificate to authenticate the user. The client certificate authentication feature allows the ProxySG appliance to store client certificates and present the appropriate certificate to the Web server upon request.

The ProxySG appliance stores individual client certificates and keys in individual keyrings. You can then write policy that instructs the appliance which client certificate to use, and when to use it.

For convenience, you can also group client certificates and keyrings into a keylist that contains all of the client certificates for a specific purpose, such as certificates for a specific website or certificates for users in a particular group. If your policy references a keylist rather than an individual keyring, you must specify how to determine which certificate to use. This is done by matching the value of a substitution variable defined in the policy against a specified certificate field attribute value within the certificate. The ProxySG appliance determines what certificate field attribute to use based on an extractor string you supply when you create the keylist.

When a certificate is requested, if the policy selects a client certificate, the appliance presents the certificate to the requesting server. If no certificate is specified in policy, an empty certificate is presented.

This feature is only applicable to intercepted SSL traffic.

To provide a client certificate to a requesting Web address, you must complete the following tasks.

1. "Add Certificates to the ProxySG Appliance" below
2. "Group Related Client Keyrings into a Keylist" on page 52
3. "Specify the Client Certificates to be Used in Policy in the VPM" on page 54, or

   "Specify the Client Certificates to be Used in Policy in the CPL" on page 53

Add Certificates to the ProxySG Appliance

Before certificates can be used in policy, they must be on the ProxySG appliance. Add the certificates to the appliance in one of the following ways:

- "Create a Keydata File" below
- "Import Certificates onto the ProxySG Appliance" on page 52

Create a Keydata File

Bundle multiple keyrings and keylists into a single keydata file for simple importing into the ProxySG appliance. The keydata file does not need to include both keyring and keylist information.

1. Open a new text file.
2. Add keyring information to the keydata file in the following format:

```
#keyring: <keyring_id>
#visibility: {show | show-director | no-show}
<Private Key>
<Certificate>
<CSR>
```

where:

- `keyring_id` – the name of the keyring.
- `visibility` – how the keyring is displayed in the show configuration output. Options include:
  - show: Private keys associated with keyrings created with this attribute can be displayed in the CLI or included as part of a profile or overlay pushed by Director.
  - show-director: Private keys associated with keyrings created with this attribute are part of the show configuration output if the CLI connection is secure (SSH/RSA) and the command is issued from Director.
  - no-show: Private keys associated with keyrings created with this attribute are not displayed in the show configuration output and cannot be part of a Director profile. The no-show option is provided as additional security for environments where the keys will never be used outside of the particular ProxySG appliance.
- `Private Key`, `Certificate`, and `CSR` – Paste the contents of the key, certificate or CSR into the text file, including the `---Begin` and `---End` tags.

In the following example, the private key and certificate has been truncated.

```
#keyring:Keyring1
#visibility:no-show
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQE...KvBgDmSIw6dTXxAT/mMUHGRd7cRew==
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIDdjCCAl4CCQC...TjUwxwboMEyL60z/tixM=
-----END CERTIFICATE-----
#keyring:Keyring2
```

3. Add keylist information to the file in the following format:

```
#keylist: <keylist_name>
#extractor: <extractor>
<keyring_id>
<keyring_id>
```

where:

- `keylist_name` - Type the name of the keylist.
- `extractor` - Enter a string that identifies which certificate field attribute value to extract to determine a policy match, using the `$(field.attribute)` syntax. Substitutions from all attributes of Subject, Issuer, SubjectAltName, IssuerAltName, and SerialNumber certificate fields are supported.
- `keyring_id` - List any keyrings to include in the keylist. The keyrings may be included in the keydata file, or may be keyrings that already exist on the ProxySG appliance.

For example:

```
#keylist:mylist
#extractor: $(Subject.CN)
```

```
Keyring1
Keyring2
```

4. Save the file as .txt on a web server that can be accessed by the ProxySG appliance.

Import Certificates onto the ProxySG Appliance

Use the following procedure to import multiple client certificates (as well as the associated key pair and CSR) into the ProxySG appliance.

1. Select **Configuration > SSL > Keyrings > Import**.
2. In the **URL** field type the path to the keydata file with the keylists and keyrings.
3. (Optional) If you have encrypted the private keys in the keydata file, type the Passphrase for the private keys.

   All keyrings or keylists being imported must have the same Passphrase for the import to be successful.
4. Click **Import**.
5. Click **Apply**.

Group Related Client Keyrings into a Keylist

To easily reference client certificate keyrings in policy, use keylists to group them together. For example it is often useful to group certificates into keylists bundled by:

- all client certificates for a specific web address,
- all client certificates for a group of users,
- all client certificates for a specific user.

All keyrings in the keylist must have the same extractor, but each certificate must have a unique value for the extractor. The evaluation of the keylist extractor string must be unique across the client certificates in the keylist, otherwise changes being applied to the keylist will fail with an error.

1. Select **Configuration > SSL > Keyrings > Keylists**.
2. Click **Create**.
3. In the **Name** field, type a name for the new keylist.
4. In the **Extractor** field enter a string that identifies which certificate field attribute value to extract to determine a policy match. Enter the string using the $(field.attribute) syntax. For example, to extract the value of the CN attribute from the Subject field of the certificate, you would enter $(subject.CN).

Alternatively, select values from the **Field**, **Attribute**, and **Group Name** drop down lists to build an extractor string, and click **Add to extractor**. The new extractor string is appended to any existing text in the **Extractor** field. The Group Name drop down list only appears for IssuerAltName and SubjectAltName fields.

The extractor supports substitutions from all attributes of Subject, Issuer, SubjectAltName, IssuerAltName, and SerialNumber certificate fields. The default extractor value is $(Subject.CN); many other subject attributes are recognized, among them OU, O, L, ST, C, and DC. Field indexes can be used in substitutions on a group name or attribute; for example $(SubjectAltName.DNS.1).

5. From the **Available Keyrings** list, select the keyrings to be included in this keylist and click **Add**.

   To remove a keyring from the list of **Included Keyrings**, select the keyring and click **Remove**.

   If any errors are noted in the Included Keyrings list, the keylist cannot be created. Possible causes for errors are:

   - The included keyring does not contain the specified extractor pattern or substitution variable.
   - Multiple keyrings have the same value for the specified extractor.

   The extracted value in the keyring allows the policy action object to find the appropriate keyring certificate to use. Only one keyring can be utilized by each policy transaction. Therefore, the extractor string evaluation must be unique across the certificates in the keylist. A keyring whose extractor value matches the extractor value of any existing keyring in the keylist will not be added to the keylist. For example, if the extractor $(Subject.DC) is selected, and all keyrings have the same value in the certificate for that extractor, the policy would not be able to determine which keyring to select.

6. To save the keylist click **OK**.
7. Click **Apply**.

Specify the Client Certificates to be Used in Policy in the CPL

To respond to client certificate requests, add a keyring or keylist with the following syntax in the <SSL> layer:

```
server.connection.client_keyring(keyring)
server.connection.client_keyring(keylist, selector)
```

where:

- `keyring`—Specifies the keyring to use for client certificate requests.
- `keylist`—Specifies the keylist to use for client certificate requests. The selector value must also be specified.
- `selector` —Takes a substitution variable.

  All substitution variables are supported; however recommended substitution variables for the selector include `$(user)`, `$(group)`, and `$(server.address)`.

Keyring Examples

- Use the certificate from <keyring> as the client certificate for user <user> connecting to a specific website <url>.

  ```
  url = <url> user = <user> server.connection.client_keyring(<keyring>)
  ```

- Use the certificate from <keyring> as the client certificate for user <user> connecting to any website that requires a client certificate.

  ```
  user = <user> server.connection.client_keyring(<keyring>)
  ```

- Use the certificate from <keyring> as the client certificate for all users of group <group> connecting to a specific website <url>.

  ```
  url = <url> group = <group> server.connection.client_keyring(<keyring>)
  ```

Keylist Examples

- Select a keyring or certificate from the keylist <keylist> whose extractor value is equal to the user of the connection, for a specific website <url>.

  ```
  <SSL>
    url = <url> server.connection.client_keyring(<keylist>, "$(user)")
  ```

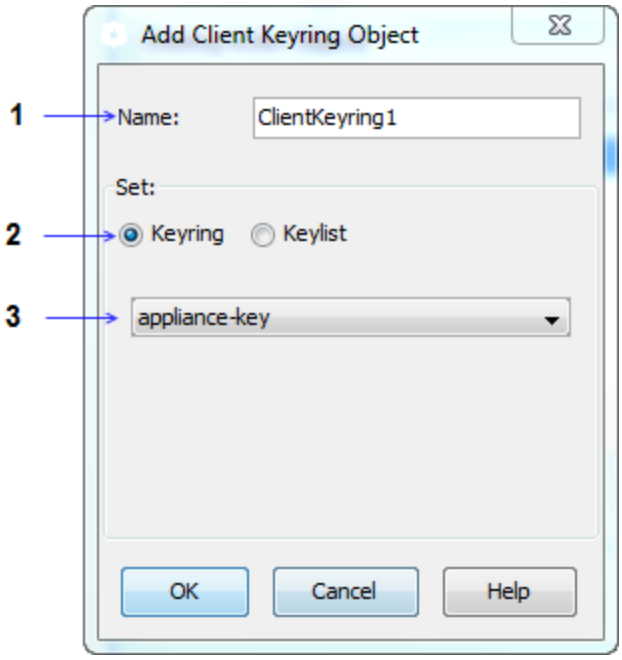
- For connections to a website <url>, this will select a keyring or certificate from keylist <keylist> whose extractor value is equal to the group of the connection.

  ```
  <SSL>
    url = <url> group = (<group>, <group>) server.connection.client_keyring(<keylist>, "$(group)")
  ```

Specify the Client Certificates to be Used in Policy in the VPM

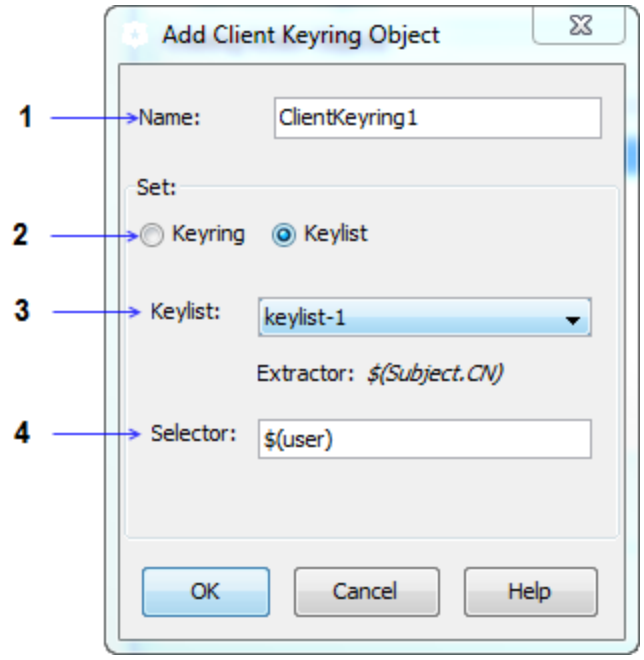
To respond to client certificate requests, in the SSL Access policy layer add an action object with the keyrings or keylists that can provide client certificates when requested.

Use a keyring



1. In the **Name** field, enter a name for the object or leave as is to accept the default.
2. Select **Keyring**.
3. From the drop-down, select the keyring to use in policy.
4. Click **OK**.

Use a keylist



1. In the **Name** field, enter a name for the object or leave as is to accept the default.
2. Select **Keylist**.
3. From the drop-down, select the keylist to use in policy.
4. In the **Selector** field, type a substitution variable.

   All substitution variables are supported; however recommended substitution variables for the selector include `$(user)`, `$(group)`, and `$(server.address)`. For information on substitution variables, see "CPL Substitutions" in the Blue Coat SGOS 6.5 Content Policy Language Reference.

   Note: The Selector value must match the set of extractor values that are displayed when you run the `view` command for a keylist. For example, if the Subject.CN in the certificate is set to represent a user name, use the Selector `$(user)`, and select the Extractor value `$(Subject.CN)`. If the Extractor value was set to `$(Subject.O)`, no match would be found and a certificate would not be sent.

   If you are using the `$(group)` selector, you must also create a list of the groups to be included in the `$(group)` substitution variable. See “Creating the Group Log Order List” in the Blue Coat SGOS 6.5 Visual Policy Manager Reference.
5. Click **OK**.

How Do I Ensure My List of Trusted CA Certificates is Up-To-Date?

When a client sends an HTTPS request to an OCS, the OCS presents its certificate to the browser. The browser then

validates the certificate using the corresponding CA certificate in its list of trusted CAs (if it has one, otherwise it will present a trust dialog to the user). When the ProxySG appliance intercepts an HTTPS connection, it terminates the client request and then initiates a new request to the OCS, posing as the client. Therefore, the ProxySG appliance must also have an up-to-date list of trusted CA certificates to enable the certificate validation process. The ProxySG appliance uses its built-in `browser-trusted` CA Certificate List (CCL) for this purpose.

By default, the ProxySG appliance will automatically download and install the CA Certificate List update from the Blue Coat website every seven days. This smart download compares the existing browser-trusted list on the appliance to the new list and only adds or removes those CA certificates that have changed since the last update. Any manual changes that you have made to the file are preserved.

The updates, which include both an updated `browser-trusted` CCL as well as the corresponding CA certificates, are packaged in a file called a *trust package* (`trust_package.bctp`), which is signed by the Blue Coat CA and must be validated before the ProxySG appliance will install it. No configuration is required to enable this. However, you can perform the following optional tasks to customize your deployment. For example, you can choose to host the trust package locally rather than having the ProxySG appliances connect to the Blue Coat website. Or, you may want to customize the automatic update interval or disable automatic updates altogether.

The `trust_package.bctp` trust package may also contain updates to the image-validation CCL and its associated CA certificates. This CCL is used to validate signed SGOS images.

1. "Set the Download Location" below
2. "Configure Automatic Updates" on the next page
3. "Load the Trust Package" on the next page
4. "Verify Trust Package Downloads" on the next page

Set the Download Location

The downloadable CA list—called a trust package—is available on the Blue Coat website at the following URL:

http://appliance.bluecoat.com/sgos/trust_package.bctp

By default, the ProxySG appliance is configured to download the trust package directly from this Blue Coat URL. As an alternative you can set up your own download site on premise. To do this, you must download the trust package from the Blue Coat website to your download server and then configure the download path on the appliances in your network.

After you determine the download location, you must configure the appliance to point to the location using the following command:

```
#(config) security trust-package download-path <URL>
```

For example, to configure the appliance to download the trust package from a bluecoat folder on a server named download.acme.com, you would enter the following command:

```
#(config) security trust-package download-path http://download.acme.com/bluecoat/trust_
package.bctp
```

The ProxySG appliance can only download and install a `trust_package.bctp` trust package created by Blue Coat Systems, Inc.

Configure Automatic Updates

After you set the download location for the trust package, the appliance automatically downloads and installs the latest trust package every seven days by default. You can disable automatic updates or modify the update interval as follows:

Disable automatic updates

If you prefer to manually download and install the trust package, you can disable automatic updates as follows:

```
#(config) security trust-package auto-update disable
```

Change the update interval

```
#(config) security trust-package auto-update interval <days>
```

where `<days>` is the number of days between updates. This value can be from 1 to 30 inclusive. For example, to set the auto-update interval to 10 days, you would enter the following command:

```
#(config) security trust-package auto-update interval 10
```

Enable automatic updates

If you previously disabled automatic updates, you can re-enable them using the following command:

```
#(config) security trust-package auto-update enable
```

Note that if you previously modified the automatic update interval, your settings will be preserved.

Load the Trust Package

If you want to manually download and install the trust package—either because you have disabled automatic updates or you want to force an update before the next automatic update—enter the following command:

```
#load trust-package
Downloading from "http://appliance.bluecoat.com/sgos/trust_package.bctp"
The trust package has been successfully downloaded.
trust package successfully installed
```

Verify Trust Package Downloads

Use the following command to view the status of the trust package downloads:

```
#show security trust-package
Download url: http://appliance.bluecoat.com/sgos/trust_package.bctp
Auto-update: enabled       Auto-update interval: 7 days
Previous (success) install via manual
  Creation time: Saturday October 1 2011 00:26:43 UTC
  CA Certificate List changes:
     browser-trusted: CAs - 3 added, 4 deleted, 0 modified
```

```
  image-validation install: Tuesday October 11 2011 00:26:27 UTC
Download log:
     Downloaded at: Tuesday October 11 2011 00:26:27 Success
     Downloaded from: http://appliance.bluecoat.com/sgos/trust_package.bctp
```

Solve a Problem

Select a topic from the Table of Contents on the left.

Cannot Reach an HTTPS Site

Problem: A request to an HTTPS site results in a failure to reach the site and the browser displays an HTML error page that describes a certificate error. In the ProxySG appliance event log, one of the following is displayed:

```
"Server certificate validation failed for support.bluecoat.com at depth 0, reason Untrusted
Issuer" 0 310000:1 ../ssl_proxy/sslproxy_worker.cpp:1157
"Server certificate validation failed for www.etrade.com at depth 0, reason Certificate expired
or not valid yet" 0 310000:1 ../ssl_proxy/sslproxy_worker.cpp:1157
```

If a site is rejected by the ProxySG appliance, it does not necessarily mean the certificate is self-signed or not valid.

Certificates not signed by a commercial signing authority, such as those signed by the United States Department of Defense, are rejected until the CA is added to the ProxySG appliance's store.

Resolution #1 (More Secure):

- For untrusted issuer errors:

  Get the CA certificate from the server administrator and import it to the ProxySG appliance. This is secure only if you can trust the CA's policies when they issue server certificates. When validating the new server certificate, make sure that a new browser instance is used.

- For expired certificate errors:

  First check the clock on your proxy. Since the expiration check compares the dates in the certificate against the proxy's clock, make sure that the correct date and time is set.

  If you still get certificate expired errors, the most secure solution is to get a new certificate with valid dates. This may not possible if you do not control the server.

Resolution #2 (Less Secure):

Create and install policy to ignore specific errors.

- To ignore untrusted issuer errors

```
<ssl>
      server_url.host="intranet.company.com" \
      server.certificate.validate.ignore.untrusted_issuer(yes)
```

- To ignore certificate expiration errors:

```
<ssl>
      server_url.host=”intranet.company.com” \
      server.certificate.validate.ignore_expiration(yes)
```

Client Certificates Do Not Work with Internet Explorer

Problem: When the ProxySG appliance requests a client certificate from the browser, it includes the list of CAs it trusts in the “Certificate Request” message. The default list of CA certificates configured on the ProxySG appliance has grown and now spans multiple SSL records. Internet Explorer cannot handle SSL handshake messages that span multiple SSL records.

Resolutions: For the SSL Proxy, this issue means that the client consent certificate feature that allows the ProxySG appliance to notify users in advance of HTTPS interception does not work with Internet Explorer. No workaround exists.

For the HTTPS Reverse Proxy, you can create a CCL, which reduces the number of CAs trusted by a service to the point where Internet Explorer can handle it.

How Do I Include Other Information in the SSL Access Log

Problem: The default access log fields for the SSL log do not contain any sensitive information. Only information that can be seen in the clear on the wire is included in the SSL access log.

Resolutions: The SSL access log is customizable, meaning that you can add fields that contain sensitive information. For more information on configuring access logs, see the Blue Coat SGOS 6.5 Administration Guide.

Why is the SSL Access Log Empty?

Problem: When you intercept and log all traffic, the log remains empty.

Resolutions: You might be logging all https-forward-proxy connections (that is, intercepted connections) to the main facility instead of the SSL facility.

Why is Windows Update Failing?

Problem: Windows Update fails when the SSL Proxy intercepts Windows Update connections. This is because Windows Update does not trust the emulated certificate presented by the SSL Proxy.

Resolutions: Always tunnel SSL connections for Windows Update.

```
<ssl-intercept>
      server.certificate.hostname=update.microsoft.com \ ssl.forward_proxy(no)
      ssl.forward_proxy(https)
```

Why does Login Through HTTP with Windows Live IM Client Fail?

Problem: Logging in to the Windows Live IM client fails if the SSL Proxy is intercepting HTTP traffic, and the proxy does not display a certificate pop-up. This is because the IM client does not trust the emulated certificate presented by the SSL Proxy.

Resolution #1: Write policy to disable SSL interception for login.live.com, such as:

```
<ssl-intercept>
      condition=!DoNotInterceptList ssl.forward_proxy(https)

; Definitions
define condition DoNotInterceptList
      server.certificate.hostname=login.live.com
      server.certificate.hostname=loginnet.passport.com
end
```

Resolution #2: Import the ProxySG appliance's issuer certificate as trusted in the browser.

How Do I Allow Skype for a Specific User?

Problem: While Skype uses HTTP and SSL as transport protocol, the application content is proprietary to Skype and does not adhere to HTTP standards.

Resolutions: To allow Skype for a specific user:

- Create a firewall policy that denies clients from going directly to the Internet.
- Allow only the ProxySG appliance to connect to the Internet for HTTP, HTTPS and FTP services.
- Install SGOS 4.2.2 or higher with a valid SSL proxy license.
- Ensure that the ProxySG appliance has SSL detection enabled for HTTP CONNECT, SOCKS, and TCP Tunnel under **Configuration > Services > SSL Proxy**.
- Verify the policy as described in Controlling Skype TechBrief.

Why are Skype Logins Failing?

Problem: Users cannot log in to Skype.

Resolutions: This might be caused by a known issue that occurs if you enable the Tunnel on Protocol Error option (**Configuration > Proxy Settings > General** tab) and the following conditions are present:

- Skype has a TCP read timeout (typically 20 seconds) that is usually lower than the ProxySG appliance timeout value (the default is 300 seconds).
- When **Tunnel on Protocol Error** is enabled and all ports except 80 and 443 are blocked on the Skype client, Skype

logins fail.

This occurs because when the Skype node connects to port 443 through the ProxySG appliance (that is intercepting SSL traffic), the ProxySG appliance waits for the server certificate for 300 seconds; however, the Skype node is not sending one. The Skype node breaks the connection after its second read timeout, which causes a login failure.

The workaround is to set the ProxySG appliance value to less than the Skype timeout value seconds, which switches the connection to a tunnel because of the server certificate absence. The CLI command to change this value is:

```
# (config ssl) ssl-nego-timeout <seconds>
```

How Do I Decipher Error Messages?

Problem: How can I tell from error messages whether the ProxySG appliance was acting as an SSL server or as an SSL client?

Resolutions: When reading SSL-related event log messages, remember the following:

- If an error message begins with CFSSL:SSL_accept error, that means the ProxySG appliance encountered errors on the client-side connection when acting as an SSL server.
- If an error message begins with CFSSL:SSL_connect error, that means the ProxySG appliance encountered errors on the server side or upstream connection when acting as an SSL client.

For example, the following are errors when the ProxySG appliance was acting as an SSL server:

```
2007-06-05 21:43:57+02:00CEST "CFSSL:SSL_accept error:1408E0F4:SSL routines:SSL3_GET_
MESSAGE:unexpected message" 0 310000:1 ../cf_ssl.cpp:1505
2007-06-05 21:44:03+02:00CEST "CFSSL:SSL_accept error:14089087:SSL routines:SSL3_GET_CLIENT_
CERTIFICATE:cert length mismatch" 0 310000:1 ../cf_ssl.cpp:1505
```

Why are Users Unable to Access Some Secured Web Sites?

Problem: When a user goes to a secure web site, the web server may request a client certificate to authenticate the user. If the ProxySG appliance does not present the certificate, a secure connection cannot be established. If the web server requests the client certificate during initial handshake (and the appliance presents the certificate) and again during a renegotiation handshake, the user may experience a dropped connection.

Resolution 1: Add the client certificate to the ProxySG appliance by either creating a keydata file or importing the certificate to the appliance. Then, install policy that enables the appliance to present the client certificate when a web server requests it. Refer to the *SGOS Administration Guide* for instructions.

Resolution 2: Allow the ProxySG appliance to detect when servers request client certificates and use policy to bypass these connections. By default, servers are added to this list as long as the `client.certificate.requested`condition exists in policy. The appliance uses the list when evaluating the policy and bypasses the transaction when it determines that a client certificate was requested during both initial handshake and renegotiation. You can use content policy language to maintain this bypass list. Refer to the *Content Policy Language Reference* for more information.