CHAPTER 3

Configuring SSL Termination

This chapter describes the steps required to configure a context on the Cisco 4700 Series Application Control Engine (ACE) appliance as a virtual SbSL server for SSL termination. It contains the following major sections:

Note To verify that the SSL connection from a client to the ACE was properly initiated, you can monitor the handshake counters in the **show stats crypto server** command output (see Chapter 6, Displaying SSL Information and Statistics). The handshake counters increment for successful connections. For example, the SSLv3 Full Handshakes counter indicates that the handshake completed successfully and the SSLv3 Resumed Handshakes counter indicates that the handshake resumed successfully by using a session ID. When traffic is flowing, those numbers should increment. If there are failures, then the alerts sent and received counters should also increment.

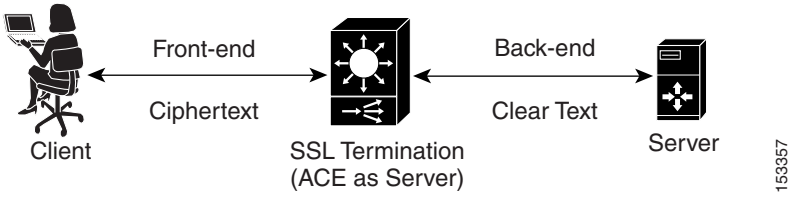
SSL Termination Overview

SSL termination occurs when the ACE, acting as an SSL proxy server, terminates an SSL connection from a client and then establishes a TCP connection to an HTTP server. When the ACE terminates the SSL connection, it decrypts the ciphertext from the client and transmits the data as clear text to an HTTP server.

Figure 3-1 shows the following network connections in which the ACE terminates the SSL connection with the client:

- Client to ACE—SSL connection between a client and the ACE acting as an SSL proxy server
- ACE to Server—TCP connection between the ACE and the HTTP server

Figure 3-1 SSL Termination with a Client

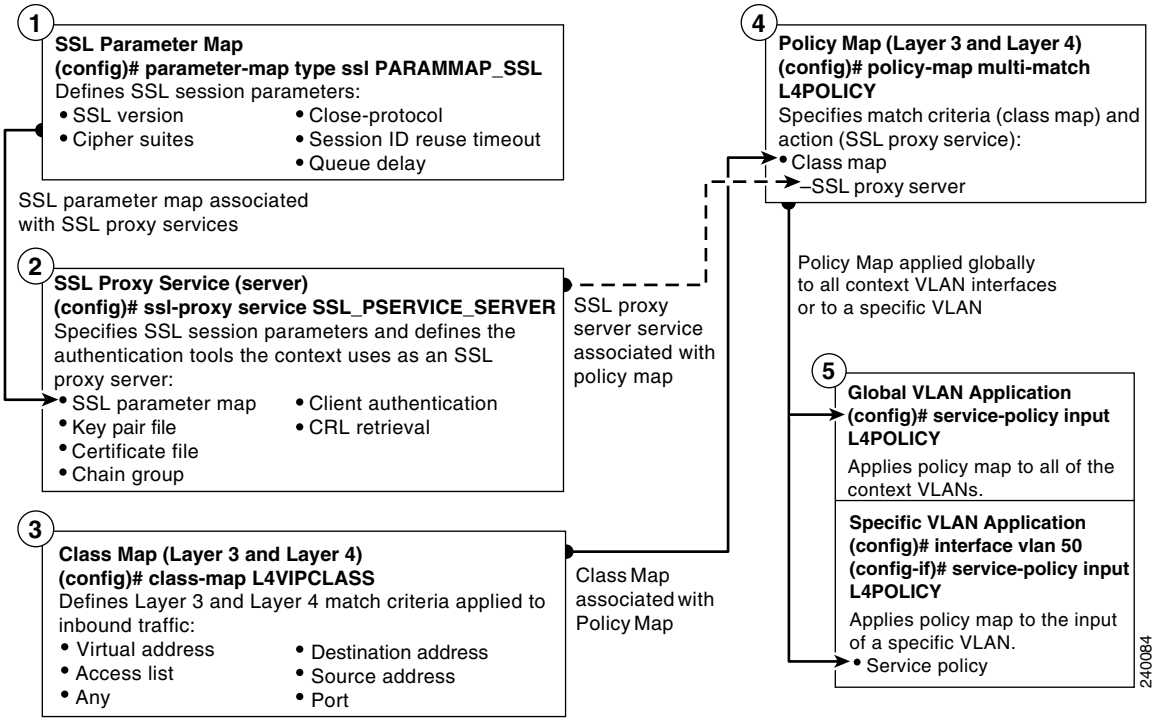


The ACE uses parameter maps, SSL proxy services, and class maps to build the policy maps that determine the flow of information between the client, the ACE, and the server. SSL termination is a Layer 3 and Layer 4 application because it is based on the destination IP addresses of the inbound traffic flow from the client. For this type of application, you create a Layer 3 and Layer 4 policy map that the ACE applies to the inbound traffic.

When configuring a policy map for SSL termination, you associate a parameter map and SSL proxy server service with the policy map to define the SSL session parameters and client/server authentication tools, such as the certificate and RSA key pair. You also associate a class map with the policy map to define the virtual SSL server IP addresses that the destination IP address of the inbound traffic must match. When a match occurs, the ACE negotiates with the client to establish an SSL connection. You can define a maximum of 250 virtual SSL servers for a single class map.

Figure 3-2 provides a basic overview of the process required to build and apply the Layer 3 and Layer 4 policy map that the ACE uses for SSL termination. The figure also shows how you associate the various components of the policy map configuration with each other.

Figure 3-2 Basic SSL Termination Configuration Flow Diagram



ACE SSL Termination Configuration Prerequisites

Before configuring your ACE for SSL operation, you must first configure it for server load balancing (SLB). During the real server and server farm configuration process, when you associate a real server with a server farm, ensure that you assign an appropriate port number for the real server. The default behavior by the ACE is to automatically assign the same destination port that was used by the inbound connection to the outbound server connection if you do not specify a port.

For example, if the incoming connection to the ACE is a secure client HTTPS connection, the connection is typically made on port 443. If you do not assign a port number to the real server, the ACE will automatically use port 443 to connect to the server, which results in the ACE making a clear-text HTTP connection over port 443. In this case, you would typically define an outbound destination port of 80, 81, or 8080 for the back-end server connection.

During the SLB traffic policy configuration process, you create the following configuration objects:

- Layer 7 class map
- Layer 3 and Layer 4 class map
- Layer 7 policy map
- Layer 3 and Layer 4 policy map

After configuring SLB, you modify the existing SLB class maps and policy maps with the SSL configuration requirements described in this guide for SSL termination.

To configure your ACE for SLB, see the *Cisco 4700 Series Application Control Engine Appliance Server Load-Balancing Configuration Guide*.

SSL Termination Configuration Quick Start

Table 3-1 provides a quick overview of the steps required to configure the ACE for SSL termination. Each step includes the CLI command or a reference to the procedure required to complete the task. For a complete description of each feature and all the options associated with the CLI commands, see the sections following Table 3-1.

Note The following quick start does not include a procedure for creating a parameter map as shown in Figure 3-2. The ACE uses the default parameter map settings as described in Table 3-2.

Table 3-1 SSL Termination Configuration Quick Start

Task and Command Example

1. If you are operating in multiple contexts, observe the CLI prompt to verify that you are operating in the desired context. If necessary, log directly in to, or change to, the correct context.
`host1/Admin# changeto C1`
`host1/C1#`
The rest of the examples in this table use the Admin context. For details on creating contexts, see the *Cisco 4700 Series Application Control Engine Appliance Virtualization Configuration Guide*.

2. Enter configuration mode.
`host1/Admin# config`
`host1/Admin(config)#`

3. Create an SSL proxy server service to define the handshake parameters that the ACE, acting as an SSL server, applies to a policy map.
`host1/Admin(config)# ssl-proxy service SSL_PSERVICE_SERVER`
`host1/Admin(config-ssl-proxy)#`

4. Configure the SSL proxy server service by defining the certificate and corresponding RSA key pair.
`host1/Admin(config-ssl-proxy)# key MYRSAKEY_SERVER`
`host1/Admin(config-ssl-proxy)# cert MYCERT_SERVER`
`host1/Admin(config-ssl-proxy)# exit`
`host1/Admin(config)#`

Table 3-1 SSL Termination Configuration Quick Start (continued)

Task and Command Example

5. Create a Layer 3 and Layer 4 class map and configure it with the input traffic match criteria as required.
`host1/Admin(config)# class-map L4VIPCLASS`
`host1/Admin(config-cmap)# match virtual-address 192.168.10.24 tcp any`
`host1/Admin(config-cmap)# exit`
`host1/Admin(config)#`

6. Create a policy map and associate the class map created in Step 5 with it.
`host1/Admin(config)# policy-map multi-match L4POLICY`
`host1/Admin(config-pmap)# class L4VIPCLASS`
`host1/Admin(config-pmap-c)#`

7. Associate the SSL proxy server service created in Step 3 with the policy map.
`host1/Admin(config-pmap-c)# ssl-proxy server SSL_PSERVICE_SERVER`
`host1/Admin(config-pmap-c)# exit`
`host1/Admin(config-pmap)# exit`
`host1/Admin(config)#`

8. Apply the policy map to the input traffic of the desired interface as follows:
Apply the policy map globally to all context VLANs.
`host1/Admin(config)# service-policy input L4POLICY`
Apply the policy map to a specific VLAN.
`host1/Admin(config)# interface vlan 50`
`host1/Admin(config-if)# service-policy input L4POLICY`

Table 3-1 SSL Termination Configuration Quick Start (continued)

Task and Command Example
9. Display the running configuration to verify the information that you just added is configured properly. `host1/Admin(config-if)# do show running-config`
10. (Optional) Save the configuration changes to flash memory by copying the running configuration to the startup configuration. `host1/Admin(config-if)# do copy running-config startup-config`

Creating and Defining an SSL Parameter Map

An SSL parameter map defines the SSL session parameters that the ACE applies to an SSL proxy service. Creating an SSL parameter map allows you to apply the same SSL session parameters to different proxy services. Table 3-2 describes each SSL session parameter with its default value.

Table 3-2 SSL Session Parameters of an SSL Parameter Map

SSL Session Parameter	Description	Default Value
Cipher suites	Defines the cipher suites that the ACE supports during the SSL handshake (see Table 3-3 for a list of available cipher suites that the ACE supports)	The ACE supports all of the available cipher suites
Close-protocol	Defines how the ACE executes close-notify messages	**none**—The ACE sends a close-notify alert message to its peer when closing a session but has no expectation of receiving one back from the peer

Table 3-2 SSL Session Parameters of an SSL Parameter Map (continued)

SSL Session Parameter	Description	Default Value
Version	Defines the SSL and TLS versions that the ACE supports during the SSL handshake	The ACE supports versions SSL3 and TLS1
Queue delay time	Defines the amount of time that the ACE keeps packet data from the server before encrypting it for the client	Disabled
Session cache timeout	Defines the amount of time that the SSL session ID remains valid before the ACE requires a new SSL handshake to establish a new SSL session.	Disabled
Expired CRL	Defines whether the ACE rejects all incoming client certificates if the CRL is expired.	Disabled

Note If you want an SSL proxy service to use the default values for the SSL session parameters, you do not need to create an SSL parameter map or associate one with the proxy service. When you do not associate a parameter map with the SSL proxy service, the ACE automatically applies the default values for the session parameters listed in Table 3-2 to the proxy service.

You can create an SSL parameter map by using the **parameter-map type ssl** command in configuration mode.

The syntax of this command is as follows:

parameter-map type ssl *parammap_name*

The *parammap_name* argument is the name of the SSL parameter map. Enter an unquoted alphanumeric string with a maximum of 64 characters.

For example, to create the SSL parameter map PARAMMAP_SSL, enter:

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL
```

After you create an SSL proxy parameter map, the CLI enters parameter map SSL configuration mode.

```
host1/Admin(config-parammap-ssl)#
```

If you exit out of parameter map SSL configuration mode without defining any of its SSL session parameters, the ACE configures the parameter map with the default values listed in Table 3-2.

To delete an existing SSL parameter map, enter:

```
host1/Admin(config)# no parameter-map type ssl PARAMMAP_SSL
```

This section contains the following topics:

- Adding a Cipher Suite
- Defining the Close-Protocol Behavior
- Defining the SSL and TLS Version
- Configuring the SSL Queue Delay
- Configuring the SSL Session Cache Timeout
- Rejecting Expired CRL Client Certificates

Adding a Cipher Suite

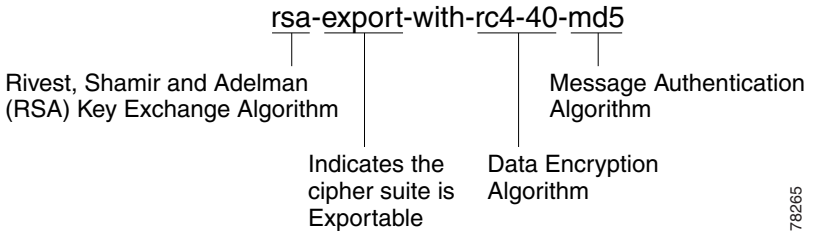
The SSL protocol supports a variety of different cryptographic algorithms, or ciphers, for use in operations such as the following:

- Authenticating the server and client to each other
- Transmitting certificates
- Establishing session keys

Clients and servers may support different cipher suites, or sets of ciphers, depending on various factors, such as the version of SSL that they support, company policies regarding acceptable encryption strength, and government restrictions on export of SSL-enabled software. Among its other functions, the SSL handshake protocol determines how the server and client negotiate which cipher suite they will use to authenticate each other, transmit certificates, and establish session keys.

As shown in Figure 3-3, a cipher suite consists of the following three algorithms: key exchange algorithm, data encryption algorithm, and message authentication (hash) algorithm.

Figure 3-3 Cipher Suite Algorithms



Note Exportable cipher suites are those cipher suites that are not as strong as some of the other cipher suites (for example, 3DES or RC4 with 128-bit encryption) as defined by U.S. export restrictions on software products. Exportable cipher suites may be exported to most countries from the United States and provide the strongest encryption available for exportable products.

To define each of the cipher suites that you want the ACE to support during a secure session, use the **cipher** command in ssl parameter-map configuration mode. The cipher suite that you choose depends on your environment and security requirements and must correlate to the certificates and keys that you have loaded on the ACE.

Note By default, the ACE supports all of the cipher suites listed in Table 3-3. The default setting works only when you do not configure the SSL parameter map with any specific ciphers. To return to using the all cipher suites setting, you must delete each specifically defined cipher from the parameter map by using the **no** form of the command.

The syntax of this command is as follows:

cipher *cipher_name* [**priority** *cipher_priority*]

The keywords and arguments are as follows:

- *cipher_name*—Name of the cipher suite that you want the ACE to support. Table 3-3 lists the cipher suites that the ACE supports. Enter one of the supported cipher suites from the table.
- **priority**—(Optional) Assigns a priority level to the cipher suite. The priority level represents the preference ranking of the cipher suite, with 10 being the most preferred and 1 being the least preferred. By default, all configured cipher suites have a priority level of 1. When negotiating which cipher suite to use, the ACE selects from the client list based on the cipher suite configured with the highest priority level. A higher priority level will bias towards the specified cipher suite. For SSL termination applications, the ACE uses the priority level to match cipher suites in the client's ClientHello handshake message. For SSL initiation applications, the priority level represents the order in which the ACE places the cipher suites in its ClientHello handshake message to the server.
- *cipher_priority*—Priority level of the cipher suite. Enter an integer from 1 to 10. The default is 1.

For example, to add the cipher suite rsa_with _aes_128_cbc_sha with a priority 2 level, enter:

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL
host1/Admin(config-parammap-ssl)# cipher rsa_with_aes_128_cbc_sha
priority 2
```

Repeat the **cipher** command for each cipher suite that you want to include in the SSL parameter map.

To delete a cipher suite from the SSL parameter map, enter:

```
host1/Admin(config-parammap-ssl)# no cipher rsa_with_aes_128_cbc_sha
```

Table 3-3 lists the available cipher suites that the ACE supports and indicates which of the supported cipher suites are exportable from the ACE. The table also lists the authentication certificate and encryption key required by each cipher suite.

If you use the default setting in which the ACE implicitly supports all of the cipher suites listed in Table 3-3 or you explicitly define each cipher suite with equal priority and the client connection uses multiple ciphers, the ACE sends the cipher suites to its peer in the same order as they appear in the table, starting with RSA_WITH_RC4_128_MD5.

Caution Cipher suites with "export" in the title indicate that they are intended for use outside of the domestic United States and have encryption algorithms with limited key sizes.

Table 3-3 SSL Cipher Suites Supported by the ACE

Cipher Suite	Exportable	Authentication Certificate Used	Key Exchange Algorithm Used
RSA_WITH_RC4_128_MD5	No	RSA certificate	RSA key exchange
RSA_WITH_RC4_128_SHA	No	RSA certificate	RSA key exchange
RSA_WITH_DES_CBC_SHA	No	RSA certificate	RSA key exchange
RSA_WITH_3DES_EDE_CBC_SHA	No	RSA certificate	RSA key exchange
RSA_WITH_AES_128_CBC_SHA	No	RSA certificate	RSA key exchange
RSA_WITH_AES_256_CBC_SHA	No	RSA certificate	RSA key exchange
RSA_EXPORT_WITH_RC4_40_MD5	Yes	RSA certificate	RSA key exchange
RSA_EXPORT1024_WITH_RC4_56_MD5	Yes	RSA certificate	RSA key exchange
RSA_EXPORT_WITH_DES40_CBC_SHA	Yes	RSA certificate	RSA key exchange
RSA_EXPORT1024_WITH_DES_CBC_SHA	Yes	RSA certificate	RSA key exchange
RSA_EXPORT1024_WITH_RC4_56_SHA	Yes	RSA certificate	RSA key exchange

Defining the Close-Protocol Behavior

You can configure how the ACE handles the sending of close-notify messages by using the **close-protocol** command in the parameter map SSL configuration mode.

The syntax of this command is as follows:

close-protocol {**disabled** | **none**}

The keywords are as follows:

- **disabled**—Specifies that the ACE does not send a close-notify alert message to its peer when closing a session with no expectation of receiving one back from the peer.

- **none**—Specifies that the ACE sends a close-notify alert message to its peer when closing a session but has no expectation of receiving one back from the peer.

For example, to set **close-protocol** to disabled, enter:

```
host1/Admin(config)# parameter-map type ssl SSL_PARAMMAP_SSL
host1/Admin(config-parammap-ssl)# close-protocol disabled
```

To configure the **close-protocol** command with the default setting of none, use the **no** form of the command:

```
host1/Admin(config-parammap-ssl)# no close-protocol
```

Defining the SSL and TLS Versions

You can specify the version of the security protocol that the ACE supports during the SSL handshake with its peer by using the **version** command in parameter map SSL configuration mode.

The syntax of this command is as follows:

version {**all** | **ssl3** | **tls1**}

The keywords are as follows:

- **all**—(Default) The ACE supports both SSL Version 3.0 and Transport Layer Security (TLS) Version 1.0.
- **ssl3**—The ACE supports only SSL Version 3.0.
- **tls1**—The ACE supports only TLS Version 1.0.

For example, to specify SSL Version 3.0 for the parameter map, enter:

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL
host1/Admin(config-parammap-ssl)# version ssl3
```

To remove a security protocol version from the SSL proxy parameter map, enter:

```
host1/Admin(config-parammap-ssl)# no version tls1
```

Configuring the SSL Queue Delay

The ACE queues packet data from the server before encrypting it for transmission to the client. The ACE empties the data from the queue for encryption when one of the following events occurs:

- The queue reaches 4096 bytes.
- The server sends a TCP-FIN segment.
- The queue delay time on the ACE has passed even though the queue had not reached 4096 bytes.

The queue delay time is the amount of time that the ACE waits before emptying the queued data for encryption. By default, the queue delay timer is disabled. You can set the delay time by using the **queue-delay timeout** command in parameter map SSL configuration mode. The syntax of this command is as follows:

queue-delay timeout *milliseconds*

The *milliseconds* argument is the time in milliseconds before the data is emptied from the queue. Enter an integer from 0 to 10000. A value of 0 disables the delay timer, causing the ACE to encrypt data from the server as it arrives and then send the encrypted data to the client.

Note The queue delay applies only to encrypted data that the ACE sends to the client.

For example, to set the queue delay time to 500 milliseconds, enter:

```
host1/Admin(config-parammap-ssl)# queue-delay timeout 500
```

To disable the queue delay timer, enter:

```
host1/Admin(config-parammap-ssl)# no queue-delay timeout
```

Configuring the SSL Session Cache Timeout

An SSL session ID is created every time that the client and the ACE perform a full SSL key exchange and establish a new master secret key. To quicken the SSL negotiation process between the client and the ACE, the SSL session ID reuse feature allows the ACE to reuse the secret key information in the session cache. On subsequent connections with the client, the ACE reuses the key stored in the cache from the last negotiated session. The ACE can store a maximum of 100,000 SSL session IDs in the session cache.

By default, SSL session ID reuse is disabled on the ACE. You can enable session ID reuse by setting a session cache timeout value for the total amount of time that the SSL session ID remains valid before the ACE requires a full SSL handshake to establish a new session.

You can set the session cache timeout by using the **session-cache timeout** command in parameter map SSL configuration mode. The syntax of this command is as follows:

session-cache timeout *seconds*

The *seconds* argument is the time in seconds that the ACE reuses the key stored in the cache before removing the session IDs. Enter an integer from 0 to 72000 (20 hours). By default, session ID reuse is disabled. A value of 0 causes the ACE to remove the session IDs from the cache when the cache is full and to implement the least-recently used (LRU) timeout policy.

For example, to set the session cache timeout to 600 seconds, enter:

```
host1/Admin(config-parammap-ssl)# session-cache timeout 600
```

To disable the timer and allow the SSL full handshake to occur for each new connection with the ACE, enter:

```
host1/Admin(config-parammap-ssl)# no session-cache timeout
```

To clear the session cache information for the context, use the **clear crypto session-cache** command. The syntax of this command is as follows:

clear crypto session-cache [**all**]

The **all** optional keyword clears all session cache information for all contexts. This option is available in the Admin context only.

Rejecting Expired CRL Client Certificates

When you configure Certificate Revocation Lists (CRLs) on the ACE for client authentication, as described in the "Using CRLs During Client Authentication" section, the CRLs contain an update field that specifies the date when a new version would be available. By default, the ACE does not use CRLs that contain an update field with an expired date and, thus, does not reject incoming client certificates using the CRL.

To configure the ACE to reject a client certificate when the CRL in use has expired, use the **expired-crl reject** command in parameter map SSL configuration mode. The syntax of this command is as follows:

expired-crl reject

For example, enter:

```
host1/Admin(config-parammap-ssl)# expired-crl reject
```

To reset the default behavior of the ACE accepting a client certificate after the CRL in use has expired, enter:

```
host1/Admin(config-parammap-ssl)# no expired-crl reject
```

Creating and Defining an SSL Proxy Service

The SSL proxy service defines the SSL parameter map, key pair, certificate, and chain group that the ACE uses during the SSL handshake. For SSL termination, you configure the ACE with an SSL proxy *server* service because the ACE acts as an SSL server.

You can create an SSL proxy server service by using the **ssl-proxy service** command in configuration mode.

The syntax of this command is as follows:

ssl-proxy service *pservice_name*

The *pservice_name* argument is the name of the SSL proxy server service. Enter an unquoted alphanumeric string with a maximum of 64 characters.

For example, to create the SSL proxy server service PSERVICE_SERVER, enter:

```
host1/Admin(config)# ssl-proxy service PSERVICE_SERVER
```

After you create an SSL proxy server service, the CLI enters SSL proxy configuration mode.

```
host1/Admin(config-ssl-proxy)#
```

To delete an existing SSL proxy server service, enter:

```
host1/Admin(config)# no ssl-proxy PSERVICE_SERVER
```

This section contains the following topics:

Associating an SSL Parameter Map with the SSL Proxy Server Service

You can associate an SSL parameter map with the SSL proxy server service by using the **ssl advanced-options** command in SSL proxy configuration mode.

The syntax of this command is as follows:

ssl advanced-options *parammap_name*

The *parammap_name* argument is the name of an existing SSL parameter map (see the "Creating and Defining an SSL Parameter Map" section). Enter an unquoted alphanumeric string with a maximum of 64 characters.

For example, to associate the parameter map PARAMMAP_SSL with the SSL proxy service, enter:

```
host1/Admin(config)# ssl-proxy service PSERVICE_SERVER
host1/Admin(config-ssl-proxy)# ssl advanced-options PARAMMAP_SSL
```

To remove the association of an SSL parameter map with the SSL proxy service, enter:

```
host1/Admin(config-ssl-proxy)# no ssl advanced-options PARAMMAP_SSL
```

Specifying the Key Pair

You can specify the key pair that the ACE uses during the SSL handshake for data encryption by using the **key** command in SSL proxy configuration mode.

Note The public key in the key pair file that you select must match the public key embedded in the certificate that you select (see the "Specifying the Certificate" section). For information on verifying a public key match, see the "Verifying a Certificate Against a Key Pair" section in Chapter 2, Managing Certificates and Keys.

The syntax of this command is as follows:

key *key_filename*

The *key_filename* argument is the name of an existing key pair file loaded on the ACE. Enter an unquoted alphanumeric string with a maximum of 40 characters.

For example, to specify the private key in the key pair file MYKEY.PEM, enter:

```
host1/Admin(config)# ssl-proxy service PSERVICE_SERVER
host1/Admin(config-ssl-proxy)# key MYKEY.PEM
```

To delete a private key from the SSL proxy service, enter:

```
host1/Admin(config-ssl-proxy)# no key MYKEY.PEM
```

Specifying the Certificate

You can specify the certificate that the ACE uses during the SSL handshake process to prove its identity by using the **cert** command in SSL proxy configuration mode.

Note The public key embedded in the certificate that you select must match the public key in the key pair file that you select (see the "Specifying the Key Pair" section). For information on verifying a public key match, see the "Verifying a Certificate Against a Key Pair" section in Chapter 2, Managing Certificates and Keys.

The syntax of this command is as follows:

cert *cert_filename*

The *cert_filename* argument is the name of an existing certificate file loaded on the ACE. Enter an unquoted alphanumeric string with a maximum of 40 characters.

For example, to specify the certificate in the certificate file MYCERT.PEM, enter:

```
host1/Admin(config)# ssl-proxy service PSERVICE_SERVER
host1/Admin(config-ssl-proxy)# cert MYCERT.PEM
```

To delete a certificate file from the SSL proxy service, enter:

```
host1/Admin(config-ssl-proxy)# no cert MYCERT.PEM
```

Specifying the Certificate Chain Group

You can specify the certificate chain that the ACE sends to its peer during the SSL handshake by using the **chaingroup** command in SSL proxy configuration mode. The ACE includes the certificate chain with the certificate that you specified for the SSL proxy service (see the "Specifying the Certificate" section).

The syntax of this command is as follows:

chaingroup *group_name*

The *group_name* argument is the name of an existing certificate chain group (see the "Creating a Chain Group" section in Chapter 2, Managing Certificates and Keys). The maximum size of a chain group is 16 KB. Enter an unquoted alphanumeric string with a maximum of 64 characters.

Note When you make a change to a chain-group certificate, the change takes effect only after you respecify the associated chain group in the SSL proxy service using the **chaingroup** command.

For example, to specify the certificate chain group MYCHAINGROUP, enter:

```
host1/Admin(config)# ssl-proxy service PSERVICE_SERVER
host1/Admin(config-ssl-proxy)# chaingroup MYCHAINGROUP
```

To delete a certificate chain group from the SSL proxy service, enter:

```
host1/Admin(config-ssl-proxy)# no chaingroup MYCHAINGROUP
```

Enabling Client Authentication

During the flow of a normal SSL handshake, the server sends its certificate to the client. The client verifies the identity of the server through the certificate. However, the client does not send any identification of its own to the server. When you enable the client authentication feature on the ACE, the ACE requires that the client sends a certificate to the server. The server then verifies the following information on the certificate:

- A recognized CA issued the certificate.
- The valid period of the certificate is still in effect.
- The certificate signature is valid.
- The CA has not revoked the certificate.

You can specify the certificate authentication group that the ACE uses during the SSL handshake and enable client authentication on this SSL proxy service by using the **authgroup** command in SSL proxy configuration mode. The ACE includes the certificates configured in the group with the certificate that you specified for the SSL proxy service (see the "Specifying the Certificate" section).

The syntax of this command is as follows:

authgroup *group_name*

The *group_name* argument is the name of an existing certificate authentication group (see the "Configuring a Group of Certificates for Authentication" section in Chapter 2, Managing Certificates and Keys). Enter an unquoted alphanumeric string with a maximum of 64 characters.

Note When you enable client authentication on the ACE, a significant performance decrease may occur on the ACE. Additional latency may occur when you configure CRL retrieval (see the "Using CRLs During Client Authentication" section).

Note When you make a change to an authgroup, the change takes effect only after you respecify the associated authgroup in the SSL proxy service using the **authgroup** command.

For example, to specify the certificate authentication group AUTH-CERT1, enter:

```
host1/Admin(config-ssl-proxy)# authgroup AUTH-CERT1
```

To delete a certificate authentication group from the SSL proxy service, enter:

```
host1/Admin(config-ssl-proxy)# no authgroup AUTH-CERT1
```

Using CRLs During Client Authentication

By default, the ACE does not use certificate revocation lists (CRLs) during client authentication. The ACE supports CRL downloads through HTTP. You can configure the SSL proxy service to use a CRL in one of the following ways:

- The ACE can scan each client certificate for the service to determine if it contains a CRL Distribution Point (CDP) pointing to a CRL in the certificate extension and then retrieve the CRL from that location if the CDP is valid. If the CDP has an http:// based URL, it uses the URL to download the CRL to the ACE appliance.
- You can manually configure the download location for the CRL from which the ACE retrieves it (see the "Configuring the Download Location for CRLs" section).

Note By default, the ACE does not reject client certificates when the CRL in use has passed its update date. To configure the ACE to reject certificates when the CRL is expired, use the **expired-crl reject** command. For more information, see the "Rejecting Expired CRL Client Certificates" section.

You can determine which CRL information to use for client authentication by using the **crl** command in SSL proxy configuration mode. The syntax of this command is as follows:

crl {*crl_name* | **best-effort**}

The argument and keyword are as follows:

- *crl_name*—Name that you assigned to the CRL when you downloaded it with the configuration mode **crypto crl** command. See the "Configuring the Download Location for CRLs" section.
- **best-effort**—Specifies that the ACE scans each client certificate to determine if it contains a CDP pointing to a CRL in the certificate extension and then retrieves the CRLs from that location, if the CDP is valid.

For example, to enable the CRL1 CRL for client authentication on an SSL proxy service, enter:

```
host1/Admin(config-ssl-proxy)# crl CRL1
```

To scan the client certificate for CRL information, enter:

```
host1/Admin(config-ssl-proxy)# crl best-effort
```

To disable the use of a downloaded CRL during client authentication, enter:

```
host1/Admin(config-ssl-proxy)# no crl CRL1
```

To disable the use of client certificates for CRL information during client authentication, enter:

```
host1/Admin(config-ssl-proxy)# no crl best-effort
```

Configuring the Download Location for CRLs

You can configure the location that the ACE uses to download the CRL to the SSL proxy service for client authentication. If the service is not configured on a policy map or the policy map is not active, the ACE does not download the CRL. The ACE downloads the CRL under the following conditions:

- When you first configure the CRL and apply it to an active Layer 4 policy map as an action (see the "Associating an SSL Proxy Server Service with the Policy Map" section).
- When you reload the ACE.
- When the NextUpdate arrives, as provided within the CRL itself, the ACE reads this information and updates the CRL based on it. The ACE downloads the updated CRL upon the next client authentication request.

You can configure a maximum of four CRLs per context. After you configure the CRL, assign it to an SSL proxy service for client authentication (see the "Using CRLs During Client Authentication" section).

The ACE translates the hostnames within the CRLs to IP addresses using a Domain Name System (DNS) client that you configure. For details about configuring a DNS client, see the "Configuring a DNS Client" section.

To configure a downloaded CRL, use the **crypto crl** command in configuration mode. The syntax of this command is as follows:

crypto crl *crl_name*

The arguments are as follows:

- *crl_name*—Name that you want to assign to the CRL. Enter an unquoted alphanumeric string with a maximum of 64 characters.
- *url*—URL where the ACE retrieves the CRL. Enter the URL full path including the CRL filename in an unquoted alphanumeric string with a maximum of 255 characters. Start the URL with the http:// prefix. Only HTTP URLs are supported.

For example, to configure a CRL that you want to name CRL1 from http://crl.verisign.com/class1.crl, enter:

```
host1/Admin(config)# crypto crl CRL1
http://crl.verisign.com/class1.crl
```

To remove the CRL, enter:

```
host1/Admin(config)# no crypto crl CRL1
```

Configuring a DNS Client

With the client authentication feature, you can configure a Domain Name System (DNS) client on the ACE to communicate with a DNS server to provide hostname-to-IP-address translation for hostnames in CRLs. For details about client authentication, see the "Using CRLs During Client Authentication" section.

Before you configure a DNS client on the ACE, ensure that one or more DNS name servers are properly configured and reachable. Otherwise, translation requests (domain lookups) from the DNS client will be discarded. You can configure a maximum of three name servers. The ACE attempts to resolve the hostnames with the configured name servers in order until the translation succeeds. If the translation fails, the ACE reports an error.

For unqualified hostnames (hostnames that do not contain a domain name), you can configure a default domain name or a list of domain names that the ACE can use to perform the following tasks:

- Complete the hostname
- Attempt a host-name-to-IP-address resolution with a DNS server

To display the DNS client configuration, use the **show running-config** command.

This section contains the following topics:

Enabling Domain Lookups

To enable the ACE to perform a domain lookup (host-to-address translation) with a DNS server, use the **ip domain-lookup** command in configuration mode. By default, this command is disabled. The syntax of this command is as follows:

ip domain-lookup

For example, to enable domain lookups, enter:

```
host1/Admin(config)# ip domain-lookup
```

To return the state of domain lookups to the default value of disabled, enter:

```
host1/Admin(config)# no ip domain-lookup
```

Configuring a Default Domain Name

The DNS client feature allows you to configure a default domain name that the ACE uses to complete unqualified hostnames. An unqualified hostname is one that does not contain a domain name (any name without a dot). When domain lookups are enabled and a default domain name is configured, the ACE appends a dot (.) and the configured default domain name to the unqualified hostname and attempts a domain lookup.

To configure a default domain name, use the **ip domain-name** command in configuration mode. The syntax of this command is as follows:

ip domain-name *name*

The *name* argument is an unquoted text string with no spaces and a maximum of 85 alphanumeric characters.

For example, to specify a default domain name of cisco.com, enter:

```
host1/Admin(config)# ip domain-name cisco.com
```

In the above example, the ACE appends .cisco.com to any unqualified hostname in a CRL before the ACE attempts to resolve the hostname to an IP address using a DNS name server.

To remove the default domain from the configuration, enter:

```
host1/Admin(config)# no ip domain-name cisco.com
```

Configuring a Domain Name Search List

Instead of configuring a single default domain name, you can configure a domain name search list that the ACE uses to complete unqualified hostnames. The domain name list can contain a maximum of three domain names. If you configure both a domain name list and a default domain name, the ACE uses only the domain name list and not the single default name. After you have enabled domain name lookups and configured a domain name list, the ACE uses each domain name in turn until it can resolve a single domain name into an IP address.

To configure a domain name search list, use the **ip domain-list** command. The syntax of this command is as follows:

ip domain-list *name*

The *name* argument is an unquoted text string with no spaces and a maximum of 85 alphanumeric characters.

For example, to configure a domain name list, enter:

```
host1/Admin(config)# ip domain-list cisco.com
host1/Admin(config)# ip domain-list foo.com
host1/Admin(config)# ip domain-list xyz.com
```

To remove a domain name from the list, enter:

```
host1/Admin(config)# no ip domain-list xyz.com
```

Configuring a Domain Name Server

To translate a hostname to an IP address, you must configure one or more (maximum of three) existing DNS name servers on the ACE. Ping the IP address of each name server before you configure it to ensure that the server is reachable.

To configure a name server, use the **ip name-server** command in configuration mode. The syntax of this command is as follows:

ip name-server *ip_address*

The *ip_address* argument is the IP address of a name server in dotted decimal notation (for example, 192.168.12.15). You can enter up to three name server IP addresses in one command line.

For example, to configure three name servers for the DNS client feature, enter:

```
host1/Admin(config)# ip name-server 192.168.12.15 192.168.12.16
192.168.12.17
```

To remove a name server from the list, enter:

```
host1/Admin(config)# no ip name-server 192.168.12.15
```

Configuring SSL URL Rewrite

When a client sends encrypted traffic to the ACE in an SSL termination configuration, the ACE terminates the SSL traffic and then sends clear text to the server. Because the server is unaware of the encrypted traffic flowing between the client and the ACE, the server may return to the client a URL in the Location header of HTTP redirect responses (301: Moved Permanently or 302: Found) in the form http://www.cisco.com instead of https://www.cisco.com. In this case, the client makes a request to the unencrypted insecure URL, even though the original request was for a secure URL. Because the client connection changes to HTTP, the requested data may not be available from the server using a clear text connection.

To solve this problem, the ACE provides SSLURL rewrite, which changes the redirect URL from http:// to https:// in the Location response header from the server before sending the response to the client. By using URL rewrite, you can avoid nonsecure HTTP redirects. All client connections to the web server will be SSL, ensuring the secure delivery of HTTPS content back to the client. The ACE uses regular expression matching to determine whether the URL needs rewriting. If a Location response header matches the specified regular expression, the ACE rewrites the URL. In addition, the ACE provides commands to add or change the SSL and the clear port numbers.

This section contains the following topics:

- Configuring an Action List
- Defining the SSL URL Rewrite String and Port
- Associating an Action List with a Layer 7 HTTP Load-balancing Policy Map

Configuring an Action List

To configure SSL URL rewrite, you must first create a new action list or use an existing action list of type modify. An action list is a named group of related actions that you want the ACE to perform. For example, to create an action list, enter the following command in configuration mode:

```
host1/Admin(config)# action-list type modify http SSL_ACTLIST
host1/Admin(config-actlist-mod)#
```

For more information about action lists, see the *Cisco 4700 Series Application Control Engine Appliance Server Load-Balancing Configuration Guide*.

Defining the SSL URL Rewrite String and Port

You can define the SSL URL, SSL port, and clear port for rewrite by using the **ssl url rewrite** command in action list modify configuration mode. The syntax of this command is as follows:

ssl url rewrite location *expression* [**sslport** *number1*] [**clearport** *number2*]

The arguments, keywords, and options are as follows:

- **location** *expression*—Specifies the rewriting of the URL in the Location response header based on a URL regular expression match. If the URL in the Location header matches the URL regular expression string that you specify, the ACE rewrites the URL from http:// to https:// and rewrites the port number. Enter an unquoted text string with no spaces and a maximum of 255 alphanumeric characters. Alternatively, you can enter a text string with spaces if you enclose the entire string in quotation marks (“).

  The location regex that you enter must be a pure URL (for example, www\.cisco\.com) with no port or path designations. To match a port, use the **sslport** and **clearport** keywords as described later in this section. If you need to match a path, use the HTTP header rewrite feature to rewrite the string. For information about the HTTP header rewrite feature, see the *Cisco 4700 Series Application Control Engine Appliance Server Load-Balancing Configuration Guide*.

  The ACE supports the use of regular expressions for matching data strings. See Table 3-4 for a list of the supported characters that you can use in regular expressions.

Note When matching data strings, the period (.) and question mark (?) characters do not have a literal meaning in regular expressions. Use the brackets ([ ]) character classes to match these symbols (for example, enter www[.]xyz[.]com instead of www.xyz.com). You can also use a backslash (\) to escape a dot (.) or a question mark (?).

- **sslport** *number1*—(Optional) Specifies the SSL port number from which the ACE translates a clear port number before sending the server redirect response to the client. Enter an integer from 1 to 65535. The default is 443.
- **clearport** *number2*—(Optional) Specifies the clear port number to which the ACE translates the SSL port number before sending a server redirect response to the client. Enter an integer from 1 to 65535. The default is 80.

For example, to specify SSL URL rewrite for the URL www.cisco.com or www.cisco.net using the default SSL port of 443 and a clear port of 8080, enter:

```
host1/Admin(config-actlist-mod)# ssl url rewrite location
www\.cisco\.* sslport 443 clearport 8080
```

In the above example, the ACE attempts to perform the following tasks:

- Match all HTTP redirects to http://www.cisco.com:8080 or http://www.cisco.net:8080
- Rewrite the HTTP redirects as https://www.cisco.com:443 or https://www.cisco.net:443
- Forward the HTTP redirects to the client

After you enter the **ssl url rewrite** command, associate the action list with a Layer 3 and Layer 4 policy map. See the "Associating an Action List with a Layer 7 HTTP Load-balancing Policy Map" section.

Table 3-4 Special Characters for Matching String Expressions

Convention	Description
.	One of any character.
.*	Zero or more of any character.
\.	Period (escaped).
[*charset*]	Match any single character from the range.

Table 3-4 Special Characters for Matching String Expressions (continued)

Convention	Description
[^charset]	Do not match any character in the range. All other characters represent themselves.
()	Expression grouping.
(expr1 | expr2)	OR of expressions.
(expr)*	0 or more of expression.
(expr)+	1 or more of expression.
expr{m,n}	Repeat the expression between *m* and *n* times, where *m* and *n* have a range of 1 to 255.
expr{m}	Match the expression exactly *m* times. The range for *m* is from 1 to 255.
expr{m,}	Match the expression *m* or more times. The range for *m* is from 1 to 255.
\a	Alert (ASCII 7).
\b	Backspace (ASCII 8).
\f	Form-feed (ASCII 12).
\n	New line (ascii 10).
\r	Carriage return (ASCII 13).
\t	Tab (ASCII 9).
\v	Vertical tab (ASCII 11).
\0	Null (ASCII 0).
\\	Backslash.
\x##	Any ASCII character as specified in two-digit hexadecimal notation.

Associating an Action List with a Layer 7 HTTP Load-balancing Policy Map

You can associate an action list with a Layer 7 HTTP loadbalancing policy map by using the **action** command in policy map load balance class configuration mode. The syntax of this command is as follows:

action *name*

The *name* argument is the identifier of an existing action list. Enter an unquoted text string with a maximum of 64 alphanumeric characters.

For example, to associate an action list for SSL URL rewrite with a Layer 7 HTTP load-balancing policy map, enter:

```
host1/Admin(config)# policy-map type loadbalance http first-match
L7_POLICY
host1/Admin(config-pmap-lb)# class L7CLASS
host1/Admin(config-pmap-lb-c)# action SSL_ACTLIST
```

To disassociate the action list from the policy map, enter:

```
host1/Admin(config-pmap-lb-c)# no action SSL_ACTLIST
```

Note

Note

Creating a Layer 3 and Layer 4 Class Map for SSL Termination

The class map that you associate with a policy map acts as a filter for traffic that matches the criteria that you specify. For SSL termination, you can define the match criteria based on one or more of the following traffic characteristics:

- Access list

- Virtual IP address
- Source IP address and subnet mask
- Destination IP address and subnet mask
- TCP/UDP port number or port range

You can create a Layer 3 and Layer 4 class map by using the **class-map** command in configuration mode. For details on creating and configuring a Layer 3 and Layer 4 class map, see the *Cisco 4700 Series Application Control Engine Appliance Server Load-Balancing Configuration Guide*.

Creating a Layer 3 and Layer 4 Policy Map for SSL Termination

For SSL termination, you configure the ACE so that it is recognized as an SSL server by a client. To accomplish this, you configure a Layer 3 and Layer 4 policy map that the ACE applies to the inbound traffic. The policy map uses the Layer 3 and Layer 4 class map that you associate with it to determine whether the inbound traffic matches the criteria that you specify. When a match is found, the ACE engages the client in the SSL handshake and establishes an SSL session using the parameters that you specify in the associated SSL proxy server service.

This section contains the following topics:

Creating a Layer 3 and Layer 4 Policy Map

You can create an SSL termination policy map by using the **policy-map** command in configuration mode.

The syntax of this command is as follows:

policy-map multi-match *policy_name*

The *policy_name* argument is the name that you assign to the policy map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, to create the policy map L4POLICY, enter:

```
host1/Admin(config)# policy-map multi-match L4POLICY
```

After you create a policy map, the CLI enters into policy map configuration mode.

```
host1/Admin(config-pmap)#
```

To delete an existing policy map, enter:

```
host1/Admin(config)# no policy-map L4POLICY
```

For information on associating an SSL class map with the policy map, see the "Associating the Layer 3 and Layer 4 Class Map with the Policy Map" section.

Associating the Layer 3 and Layer 4 Class Map with the Policy Map

You can associate the Layer 3 and Layer 4 class map with the policy map by using the **class** command in policy map configuration mode.

The syntax of this command is as follows:

class *class-map*

The *class-map* argument is the name of an existing class map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, to associate the class map L4VIPCLASS with the policy map, enter:

```
host1/Admin(config)# policy-map multi-match L4POLICY
host1/Admin(config-pmap)# class L4VIPCLASS
```

After you associate a class map with the policy map, the CLI enters into policy-map class-map configuration mode.

```
host1/Admin(config-pmap-c)#
```

To remove the association of a class map to the policy map, enter:

```
host1/Admin(config-pmap)# no class L4VIPCLASS
```

For information on associating an SSL proxy service with the class map, see the "Associating an SSL Proxy Server Service with the Policy Map" section.

Associating an SSL Proxy Server Service with the Policy Map

You can associate an SSL proxy server service with the policy map by using the **ssl-proxy server** command in policy map class configuration mode.

The syntax of this command is as follows:

ssl-proxy server *pservice*

The *pservice* argument is the name of an existing SSL proxy server service. Enter an unquoted alphanumeric string with a maximum of 64 characters.

For example, to associate the SSL proxy server service PSERVICE_SERVER with the policy map, enter:

```
host1/Admin(config)# policy-map multi-match L4POLICY
host1/Admin(config-pmap)# class L4VIPCLASS
host1/Admin(config-pmap-c)# ssl-proxy server PSERVICE_SERVER
```

To remove the class map association, enter:

```
host1/Admin(config-pmap-c)# no ssl-proxy server PSERVICE_SERVER
```

Applying the Policy Map to the VLANs

This section describes how to apply the Layer 3 and Layer 4 policy map to the VLAN traffic. The ACE allows you to apply the policy globally to all VLANs within the current context or to a specific VLAN in the context.

This section contains the following topics:

- Applying the Policy Map Globally
- Applying the Policy Map to a Specific VLAN

Applying the Policy Map Globally

You can globally apply the policy map to all VLANs in the context by using the **service-policy** command in configuration mode.

The syntax of this command is as follows:

service-policy input *policy_name*

The *policy_name* argument is the name of an existing policy map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, to globally apply the policy map L4POLICY to all VLANs in the context, enter:

```
host1/Admin(config)# service-policy input L4POLICY
```

To globally remove the policy from all VLANs, enter:

```
host1/Admin(config)# no service-policy input L4POLICY
```

Applying the Policy Map to a Specific VLAN

To apply a policy map to a specific VLAN interface, you must enter interface configuration mode by using the **interface** command in configuration mode.

The syntax of this command is as follows:

interface vlan *vlan*

The *vlan* argument is the context VLAN number. Enter an integer from 2 to 4094.

For example, to enter interface configuration mode for VLAN 10, enter:

```
host1/Admin(config)# interface vlan 10
host1/Admin(config-if)#
```

You can apply the policy map to the interface by using the **service-policy** command in interface configuration mode.

The syntax of this command is as follows:

service-policy input *policy-name*

The *policy-name* argument is the name of an existing policy map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, to apply the policy map L4POLICY to VLAN 10, enter:

```
host1/Admin(config)# interface vlan 10
host1/Admin(config-if)# service-policy input L4POLICY
```

To remove the policy from the interface, enter:

```
host1/Admin(config-if)# no service-policy input L4POLICY
```

Example of an SSL Termination Configuration

The following example illustrates a running configuration of the ACE acting as an SSL proxy server; terminating SSL or TLS connections from a client and then establishing a TCP connection to an HTTP server. When the ACE terminates the SSL or TLS connection, it decrypts the cipher text from the client and transmits the data as clear text to the HTTP server. The SSL termination configuration appears in bold in the example.

```
access-list ACL1 line 10 extended permit ip any any

probe https GEN-HTTPS
  port 80
  interval 50
  faildetect 5
  expect status 200 200

serverfarm host SFARM1
  description SERVER FARM 1 FOR SSL TERMINATION
  probe GEN-HTTPS
  rserver SERVER1 80
    inservice
  rserver SERVER2 80
    inservice
  rserver SERVER3 80
    inservice
  rserver SERVER4 80
    inservice

serverfarm host SFARM2
  description SERVER FARM 2 FOR SSL TERMINATION
  probe GEN-HTTPS
```

```
rserver SERVER5 80
  inservice
rserver SERVER6 80
  inservice
rserver SERVER7 80
  inservice
rserver SERVER8 80
  inservice
```

```
parameter-map type ssl PARAMMAP_SSL_TERMINATION
  cipher RSA_WITH_3DES_EDE_CBC_SHA
  cipher RSA_WITH_AES_128_CBC_SHA priority 2
  cipher RSA_WITH_AES_256_CBC_SHA priority 3
  version all
parameter-map type connection TCP_PARAM
  syn-data drop
  exceed-mss allow

ssl-proxy service SSL_PSERVICE_SERVER
  ssl advanced-options PARAMMAP_SSL_TERMINATION
  key MYKEY.PEM
  cert MYCERT.PEM

class-map type http loadbalance match-all L7_SERVER_CLASS
  description Sticky for SSL Testing
  2 match http url .*.jpg
  3 match source-address 192.168.130.0 255.255.255.0
class-map type http loadbalance match-all L7_SLB-HTTP_CLASS
  2 match http url .*
  3 match source-address 192.168.130.0 255.255.255.0
class-map match-all L4_SSL-TERM_CLASS
  description SSL Termination VIP
  2 match virtual-address 192.168.130.11 tcp eq https

policy-map type loadbalance first-match L7_SSL-TERM_POLICY
  class L7_SERVER_CLASS
    serverfarm SFARM1
    insert-http I_AM header-value "SSL_TERM"
    insert-http SRC_Port header-value "%ps"
    insert-http DEST_IP header-value "%id"
    insert-http DEST_Port header-value "%pd"
    insert-http SRC_IP header-value "is"
  class L7_SLB-HTTP_CLASS
    serverfarm SFARM1
    insert-http I_AM header-value "SSL_TERM"
    insert-http SRC_Port header-value "%ps"
    insert-http DEST_IP header-value "%id"
    insert-http DEST_Port header-value "%pd"
    insert-http SRC_IP header-value "is"
policy-map multi-match L4_SSL-VIP_POLICY
  class L4_SSL-TERM_CLASS
    loadbalance vip inservice
    loadbalance policy L7_SSL-TERM_POLICY
    loadbalance vip icmp-reply
    ssl-proxy server SSL_PSERVICE_SERVER
    connection advanced-options TCP_PARAM
```

```
interface vlan 120
  description Upstream VLAN_120 - Clients and VIPs
  ip address 192.168.120.1 255.255.255.0
  fragment chain 20
  fragment min-mtu 68
  access-group input ACL1
  nat-pool 1 192.168.120.70 192.168.120.80 netmask 255.255.255.0 pat
  service-policy input L4_SSL-VIP_POLICY
  no shutdown
ip route 10.1.0.0 255.255.255.0 192.168.120.254
```