


Contents

- [1 Goal](#)
- [2 Design](#)
- [3 Obtaining a Key and Certificate](#)
 - ◆ [3.1 Using OpenSSL to Generate a Self Signed Certificate](#)
 - ◆ [3.2 Using the ACE to Request a Certificate From a CA](#)
- [4 Configuration](#)
- [5 Related show Commands](#)
- [6 Comments](#)
- [7 show running-config](#)
- [8 Related Information](#)

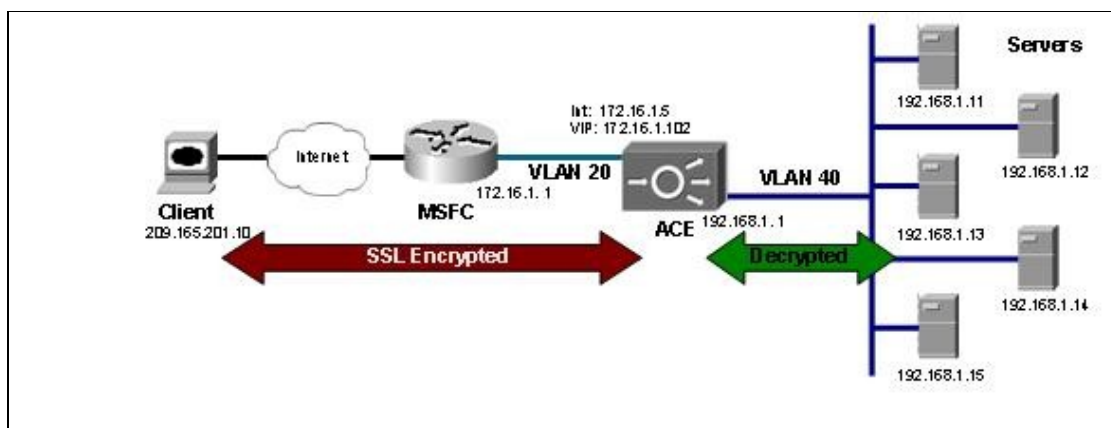
Goal

This document describes how to configure Secure Sockets Layer (SSL) termination on the Cisco® Application Control Engine (ACE) for an existing basic load-balancing configuration. This type of configuration will remove the burden of SSL encryption/decryption from the real servers' CPUs, increasing the amount of traffic they are able to handle in most situations. This document will assume that the reader does not already possess a key and certificate. It will walk the reader through the process of both creating a key and certificate using OpenSSL, as well as using the ACE to request a certificate from a Certificate Authority (CA).

 **Note:** This document does not cover the design and configuration of the basic load-balancing configuration; this is covered in the [Basic Load Balancing Using Routed Mode on the Cisco Application Control Engine Configuration Example](#).

Design

Clients will establish a connection using HTTPS (SSL) to the virtual IP address (VIP) configured on the Cisco ACE. HTTPS causes the client's TCP session to be encrypted between the browser and the ACE. Once the session reaches the ACE, the ACE will decrypt the session and forward it to a real server in clear text (HTTP). The following figure illustrates this process.



Obtaining a Key and Certificate

In order for the Cisco ACE to be able to terminate SSL sessions, it will need to be configured with both an SSL certificate and a corresponding SSL key. SSL files (both certificate and key) can either be generated using a tool such as OpenSSL or requested from a certificate authority such as Verisign or GoDaddy. The next sections will cover both of these methods.

Using OpenSSL to Generate a Self Signed Certificate

OpenSSL is pre-installed on most linux systems and it can be downloaded and installed easily onto Windows systems as well. In this section, OpenSSL will be used to generate an RSA key and a self-signed certificate. While a self signed certificate will function properly with ACE SSL termination, most client browsers will display security warnings as self signed certificate are not trusted by a known root CA. The first step is to generate a 1024 bit RSA key. The following command will create a PEM formatted file named key.pem containing the key.

```
[root@admin]# openssl genrsa ?out key.pem 1024
```

Next use OpenSSL to generate a certificate from the key. The following command will ask for a number of answers which must be provided. The result will be a PEM formatted file named cert.pem containing the self signed certificate.

```
[root@admin]# openssl req -new -x509 -nodes -sha1 -days 365
-key key.pem ?out cert.pem
```

Now the SSL files are created, skip the ?Using the ACE to Request a Certificate From a CA? section and continue to the Configuration section below to complete the ACE configuration using the newly created certificate and key.

Using the ACE to Request a Certificate From a CA

The ACE can be used to generate an RSA key, as well as a certificate request. After submitting the certificate request to a CA, the CA will supply a certificate to be imported into the ACE.

The first step in this process is to generate an RSA key on the ACE.

```
ACE-1/routed# crypto generate key 1024 key.pem
```

Next use the ACE to create a Certificate Signing Request (CSR) from the key generated in the previous step. On the ACE this is a two step process. First a CSR parameter map must be created, and then a CSR is generated from the key and the CSR parameter map. For more information on the various parameters, there is a good reference in the following URL.

http://cisco.com/en/US/docs/interfaces_modules/services_modules/ace/v3.00_A2/configuration/ssl/guide/certkeys.html#w

```
ACE-1/routed(config)# crypto csr-params PARAMS-1
ACE-1/routed(config-csr-params)# common-name www.cisco.com
ACE-1/routed(config-csr-params)# country US
ACE-1/routed(config-csr-params)# state California
ACE-1/routed(config-csr-params)# serial-number 1
ACE-1/routed(config-csr-params)# locality San Jose
ACE-1/routed(config-csr-params)# organization-name Cisco Systems
ACE-1/routed(config-csr-params)# organization-unit ADBU
ACE-1/routed(config-csr-params)# email acetme@cisco.com
```


easiest way to accomplish this is by placing the two files onto a secure FTP (SFTP) or FTP server so they can be transferred to the ACE.

```
ACE-1/routed# crypto import ftp 172.25.91.127 cisco cert.pem cert.pem
ACE-1/routed# crypto import ftp 172.25.91.127 cisco intermediate.pem intermediate.pem
ACE-1/routed# crypto import ftp 172.25.91.127 cisco key.pem key.pem
```

Note: If the files are in privacy enhanced mail (PEM) format, you can cut and paste to import the SSL file using the terminal parameter. For information on how to use the ?crypto import? command, visit the following URL:

http://cisco.com/en/US/docs/interfaces_modules/services_modules/ace/v3.00_A2/command/reference/execcmds.html#wp1

Once the SSL files have been imported, they should be checked to ensure that they were uploaded properly and to verify that they match. If the two files do not match, the RSA key cannot be exchanged and the ACE will not be able to properly terminate client connections.

```
ACE-1/routed# show crypto files
Filename                               File   File   Expor   Key/
                                      Size  Type   table   Cert
-----
cert.pem                               1354  PEM    Yes     CERT
key.pem                                887   PEM    Yes     KEY
```

```
ACE-1/routed# crypto verify key.pem cert.pem
Keypair in key.pem matches certificate in cert.pem.
```

After the SSL files have been verified, the Cisco ACE can be configured with an SSL proxy service, which is a logical grouping of the certificates, key, and SSL parameters used to define the characteristics of SSL termination on the ACE.

```
ACE-1/routed(config)# ssl-proxy service proxy-1
ACE-1/routed(config-ssl-proxy)# cert cert.pem
ACE-1/routed(config-ssl-proxy)# key key.pem
```

Within the ACE, all SSL termination is fully integrated. Therefore, there is no need to configure internal VLANs or IPs to handle decrypted traffic. All that is required to enable SSL termination is to attach the SSL proxy service configured above to a VIP in a service policy.

```
ACE-1/routed(config)# policy-map multi-match client-vips
ACE-1/routed(config-pmap)# class 102-vip
ACE-1/routed(config-pmap-c)# ssl-proxy server proxy-1
```

At this point the ACE should be configured with a working SSL termination configuration. Make a test connection to the VIP address using HTTPS in a web browser, and you should see a response from one of the real servers.

Related show Commands

This section provides information you can use to confirm your configuration is working properly.

Certain show commands are supported by the [Output Interpreter Tool \(registered customers only\)](#), which allows you to view an analysis of show command output.

```
ACE-1/routed# show crypto files
```

```
ACE-1/routed# show crypto certificate all
ACE-1/routed# show crypto key all
ACE-1/routed# show crypto session
ACE-1/routed# show crypto hardware
ACE-1/routed# show service-policy <name> detail
```

Comments

Once the configuration is complete, check to make sure the VIP address can be accessed via HTTPS in a web browser. If any certificate errors are shown, this indicates a problem with the certificate, not with the Cisco ACE configuration. The above commands can be used to verify that SSL sessions are being terminated successfully.

When a client's web browser connects to an SSL server on any device, the browser and server negotiate which encryption cipher to use for the session. The list and order of ciphers presented by the ACE in a default configuration are as follows.

1. CM_SSL_RSA_WITH_RC4_128_MD5
2. CM_SSL_RSA_WITH_RC4_128_SHA
3. CM_SSL_RSA_WITH_DES_CBC_SHA
4. CM_SSL_RSA_WITH_3DES_EDE_CBC_SHA
5. CM_SSL_RSA_WITH_AES_128_CBC_SHA
6. CM_SSL_RSA_WITH_AES_256_CBC_SHA
7. CM_SSL_RSA_EXPORT_WITH_RC4_40_MD5
8. CM_SSL_RSA_EXPORT1024_WITH_RC4_56_MD5
9. CM_SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
10. CM_SSL_RSA_EXPORT1024_WITH_DES_CBC_SHA
11. CM_SSL_RSA_EXPORT1024_WITH_RC4_56_SHA

If this list is not desirable or the order needs to be changed, an SSL parameter map can be configured to make such changes. The documentation on this feature can be found at the following URL:

http://cisco.com/en/US/docs/interfaces_modules/services_modules/ace/v3.00_A2/configuration/ssl/guide/terminat.html#w

show running-config

```
logging enable

access-list everyone line 8 extended permit ip any any
access-list everyone line 16 extended permit icmp any any

rserver host lnx1
  ip address 192.168.1.11
  inservice
rserver host lnx2
  ip address 192.168.1.12
  inservice
rserver host lnx3
  ip address 192.168.1.13
  inservice
rserver host lnx4
  ip address 192.168.1.14
  inservice
rserver host lnx5
  ip address 192.168.1.15
  inservice

ssl-proxy service proxy-1
  key key.pem
```

```
cert cert.pem

serverfarm host web
  rserver lnx1 80
    inservice
  rserver lnx2 80
    inservice
  rserver lnx3 80
    inservice
  rserver lnx4 80
    inservice
  rserver lnx5 80
    inservice

class-map match-all 102-vip
  2 match virtual-address 172.16.1.102 tcp eq https

policy-map type management first-match remote-access
  class class-default
    permit

policy-map type loadbalance http first-match slb
  class class-default
    serverfarm web

policy-map multi-match client-vips
  class 102-vip
    loadbalance vip inservice
    loadbalance policy slb
    ssl-proxy server proxy-1

interface vlan 20
  description "Client Side"
  ip address 172.16.1.5 255.255.255.0
  access-group input everyone
  service-policy input client-vips
  no shutdown
interface vlan 40
  description "Default gateway of real servers"
  ip address 192.168.1.1 255.255.255.0
  service-policy input remote-access
  no shutdown

ip route 0.0.0.0 0.0.0.0 172.16.1.1
```

Related Information

[Technical Support & Documentation - Cisco Systems](#)