Contents

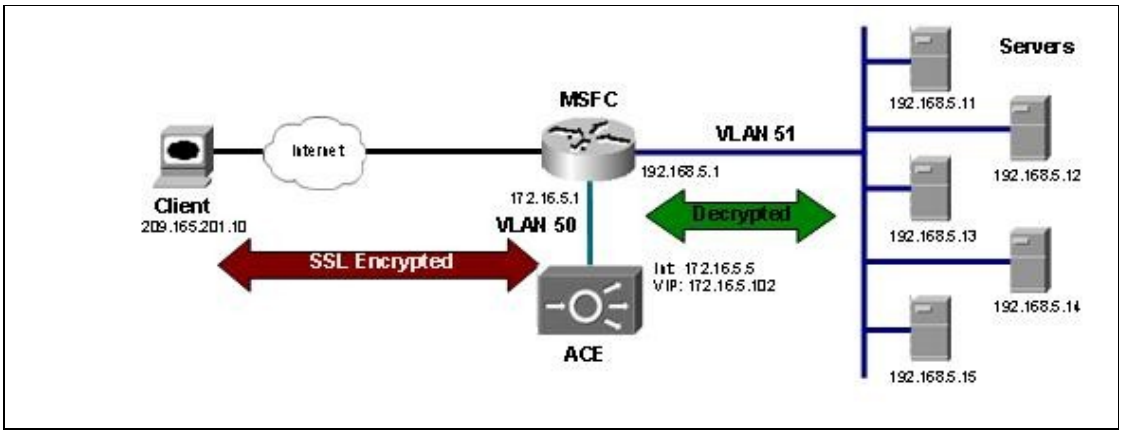
Goal

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 posses 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 One Arm Mode with Source NAT 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). ACE will rewrite the destination IP to that of the rserver, and rewrite the source IP with one from a nat-pool. Once the client request is fully NAT?d it will be sent to the server over the same VLAN which it was originally received. The server will respond to the ACE, based on the source IP of the request. The ACE will receive the response, change the source IP to be the VIP, and send it to the MSFC. The MSFC will forward the response to the client. 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. Once imported, these SSL files are associated with an SSL proxy service that is applied to the VIP to enable SSL termination.

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

```
ACE-1/ACE-1/onearm(config-csr-params)# email acetme@cisco.com

ACE-1/ACE-1/onearm# crypto generate csr PARAMS-1 key.pem
-----BEGIN CERTIFICATE REQUEST-----
MIIB1jCCAT8CAQAwgZUxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlh
MREwDwYDVQQHEwhTYW4gSm9zZTEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczENMAsG
A1UECxMEQURCVTEWMBQGA1UEAxMNd3d3LmNpc2NvLmNvbTEfMB0GCSqGSIb3DQEJ
ARYQYWNldG1lQGNpc2NvLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
sjynhIa/Lz6YCl7gEOlhY3L5sBsegVMC6eYaQt1eaJnfMog8vfY7BlmjupGxnak1
FKIPkfEewKRRve5DDKoFE34oPbOqf1zBtcgzNI9A8UVm1U1P3w5HIWXQ0fyWpvcr
ZIs0nVfKRIr12OIju2QZ2l6JFdX9vbXGdKK3H2wcQVcCAwEAAaAAMA0GCSqGSIb3
DQEBBAUAA4GBACQJFK0bkYV/nKt/MJ5mIWmZYUwgBTy6we20pz1Rnx61C+ulOwnu
596JvlAhP6iDQt2ImITClChpQzXS9xlC5VIdLW5mUYt9KUIMzI9+5eJ0WWHUVtEI
wSkOm7IDptpll7xIfcbtUYupU1wCg3KudNtGmXUTjZ1QCAWwlIknu6R1
-----END CERTIFICATE REQUEST-----
```

At this point, copy and past the certificate request into the web form provided by the CA that will provide the certificate. Once the CA signs the request, it will send back a certificate to be used on the ACE. The CA Thawte (www.thawte.com) will provide a variety of trial test certificates and trust chains for immediate use.

The Configuration section below will assist with importing the certificate and finishing the ACE configuration. Note that in the configuration section, you will only need to import the certificate, since the key was generated on the ACE.

Configuration

The SSL termination configuration begins like the basic Layer 4 load-balancing configuration, by defining a VIP and corresponding server farm and rservers. Although the VIP can be configured with a port of ?any,? the ACE will do a TCP reset on any non-SSL connections. To prevent this, it is recommended that you bind the VIP to a port. In this example, the IP address 172.16.1.100 and port 443 will be used.

```
ACE-1/onearm(config)# class-map match-all 102-vip
ACE-1/onearm(config-cmap)# match virtual-address 172.16.5.102 tcp eq 443
```

When adding the rservers to the server farm, consider the destination of the decrypted traffic. In almost every case, encrypted SSL traffic is received on an SSL-specific port, and the decrypted traffic needs to be sent to another port on the real servers. The Cisco ACE uses the rserver port defined in the server farm to properly translate (using Port Address Translation [PAT]) the destination for the decrypted connection.

```
ACE-1/onearm# show run serverfarm
Generating configuration....

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

The VIP and server farm in this example allow the Cisco ACE to accept connections to the VIP on port 443 and forward them to a real server on port 80. Note that if the port is not provided, the VIP port will be preserved.

Most SSL termination configurations begin by importing the certificate and key onto the Cisco ACE. The 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/onearm# crypto import ftp 172.25.91.127 cisco cert.pem cert.pem
ACE-1/onearm# 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 ter
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/execmds.h

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/onearm# 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/onearm# 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/onearm(config)# ssl-proxy service proxy-1
ACE-1/onearm(config-ssl-proxy)# cert cert.pem
ACE-1/onearm(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/onearm(config)# policy-map multi-match client-vips
ACE-1/onearm(config-pmap)# class 102-vip
ACE-1/onearm(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/onearm# show crypto files
ACE-1/onearm# show crypto certificate all
```

```
ACE-1/onearm# show crypto key all
ACE-1/onearm# show crypto session
ACE-1/onearm# show crypto hardware
ACE-1/onearm# 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 192.168.5.11
  inservice
rserver host lnx2
  ip address 192.168.5.12
  inservice
rserver host lnx3
  ip address 192.168.5.13
  inservice
rserver host lnx4
  ip address 192.168.5.14
  inservice
rserver host lnx5
  ip address 192.168.5.15
  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 172.16.5.102 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
    nat dynamic 5 vlan 50

interface vlan 50
  description "Client-Server VLAN"
  ip address 172.16.5.5 255.255.255.0
  access-group input everyone
  service-policy input client-vips
  service-policy input remote-access
  nat-pool 5 172.16.5.200 172.16.5.209 netmask 255.255.255.0 pat
  no shutdown

ip route 0.0.0.0 0.0.0.0 172.16.5.1
```

Related Information

Technical Support & Documentation - Cisco Systems