This article describes how to troubleshoot the SSL AO.

Guide Contents
Main Article
Understanding the WAAS Architecture and Traffic Flow
Preliminary WAAS Troubleshooting
Troubleshooting Optimization
Troubleshooting Application Acceleration
Troubleshooting the CIFS AO
Troubleshooting the HTTP AO
Troubleshooting the EPM AO
Troubleshooting the MAPI AO
Troubleshooting the NFS AO
Troubleshooting the SSL AO
Troubleshooting the Video AO
Troubleshooting the Generic AO
Troubleshooting Overload Conditions
Troubleshooting WCCP
Troubleshooting AppNav
Troubleshooting Disk and Hardware Problems
Troubleshooting Serial Inline Clusters
Troubleshooting vWAAS
Troubleshooting WAAS Express
Troubleshooting NAM Integration

Contents

SSL Accelerator Overview

The SSL accelerator (available in 4.1.3 and later) optimizes encrypted Secure Sockets Layer (SSL) and Transport Layer Security (TLS) traffic. The SSL accelerator provides traffic encryption and decryption within WAAS to enable end-to-end traffic optimization. The SSL accelerator also provides secure management of the encryption certificates and keys.

In a WAAS network, the data center WAE acts as a trusted intermediary node for SSL requests by the client. The private key and server certificate are stored on the data center WAE. The data center WAE participates in the SSL handshake to derive the session key, which it distributes securely in-band to the branch WAE, allowing the branch WAE to decrypt client traffic, optimize it, reencrypt it, and send it over the WAN to the data center WAE. The data center WAE maintains a separate SSL session with the origin server.

The following services are relevant for SSL/TLS optimization:

- Accelerated Service ? A configuration entity that describes acceleration characteristics to be applied for a SSL server or set of servers. Specifies the certificate and private key to be used while posing as a trusted intermediary, ciphers to be used, SSL version allowed, and certificate verification settings.
- Peering Service ? A configuration entity that describes acceleration characteristics to be applied for in-band SSL connections between branch and data-center WAEs. This service is used for transferring session key information from data-center to branch WAEs for optimizing SSL connections.
- Central Manager Admin Service ? Not used directly by the SSL accelerator, but to be used by an administrator for the configuration management of SSL accelerated services. Also used to upload certificates and private keys to be used in SSL accelerated services.
- Central Manager Management Service ? Not used directly by the SSL accelerator, but used for communication between application accelerator devices and the Central Manager. This service is used for configuration management, secure store encryption key retrieval, and device status updates.

The Central Manager secure store is essential for the SSL AO to operate because it stores secure encryption keys for all WAEs. After each Central Manager reload, the administrator needs to reopen the secure store by providing the passphrase with the **cms secure-store open** command. A WAE automatically retrieves its secure store encryption key from the Central Manager whenever the WAE reboots, so no action is required on the WAE after a reload.

If clients are using an HTTP proxy solution, the initial connection is handled by the HTTP AO, which recognizes it as an SSL tunnel request to port 443. The HTTP AO looks for a matching SSL accelerated service defined on the data center WAE and when it finds a match, hands off the connection to the SSL AO. However, the traffic that the HTTP AO hands off to the SSL AO for an HTTPS proxy gets reported as part of the web application statistics, not in the SSL application. If the HTTP AO does not find a match, the connection is optimized as per static HTTPS (SSL) policy configuration.

The SSL AO can use self-signed certificates rather than CA-signed certificates, which can be helpful in deploying proof of concept (POC) systems and in troubleshooting SSL issues. By using self-signed certificates, you can quickly deploy a WAAS system without having to import the origin server certificates, and you can eliminate certificates as a potential source of problems. You can configure a self-signed certificate in the Central Manager when creating an SSL Accelerated Service. However, when you use a self-signed certificate, the client browser will display a security alert that the certificate is untrusted (because it is not signed by a well-known CA). To avoid this security warning, install the certificate in the Trusted Root Certification Authorities store on the client browser. (On Internet Explorer, on the security warning, click **View Certificate**, then on the Certificate dialog click **Install Certificate** and complete the Certificate Import Wizard.)

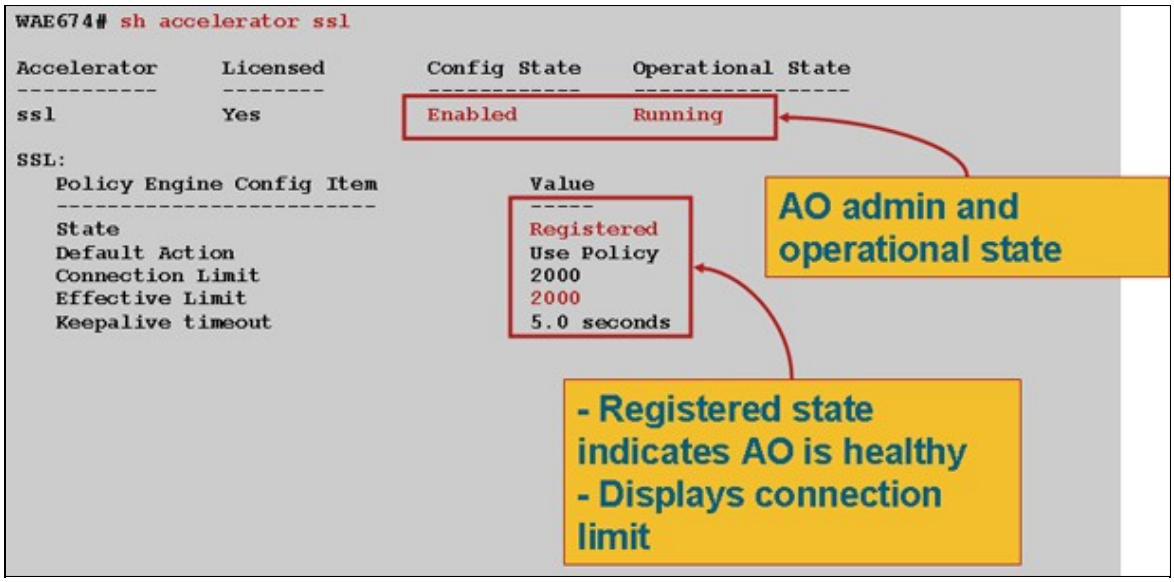
Configuring the SSL Management Services is optional, and allows you to change the SSL version and cipher list used for Central Manager communications to WAEs and to the browser (for administrative access). If you configure ciphers that are not supported by your browser, you will lose the connection to the Central Manager. In this case, use the **crypto ssl management-service** configuration command from the CLI to set the SSL management service settings back to the default.

Troubleshooting the SSL AO

You can verify the general AO configuration and status with the **show accelerator** and **show license** commands, as described in the Troubleshooting Application Acceleration article. The Enterprise license is required for SSL accelerator operation.

Next, verify the status that is specific to the SSL AO on both the data center and branch WAEs by using the **show accelerator ssl** command, as shown in Figure 1. You want to see that the SSL AO is Enabled, Running, and Registered, and that the connection limit is displayed. If the Config State is Enabled but the Operational State is Shutdown, it indicates a licensing problem. If the Operational State is Disabled, it may be because the WAE cannot retrieve the SSL keys from the Central Manager secure store, either because the secure store is not open or the Central Manager is unreachable. Use the **show cms info** and **ping** commands to confirm that the Central Manager is reachable.

Figure 1. Verifying the SSL Accelerator Status



If you see an Operational State of Gen Crypto Params, wait until the status becomes Running, which may take a few minutes following a reboot. If you see a state of Retrieving Keys from CM for more than a few minutes, it could indicate that the CMS service on the Central Manager is not running, that there is no network connectivity to the Central Manager, that the WAAS versions on the WAE and Central Manager are incompatible, or that the Central Manager secure store is not open.

You can verify that the Central Manager secure store is initialized and open by using the **show cms secure-store** command as follows:

```
cm# show cms secure-store
secure-store is initialized and open.
```

If the secure store is not initialized or open, you will see critical alarms such as mstore_key_failure and secure-store. You can open the secure store with the **cms secure-store open** command or from the Central

Manager, choose **Admin > Secure Store**.

Tip: Document the secure store password to avoid having to reset the secure store if you forget the password.

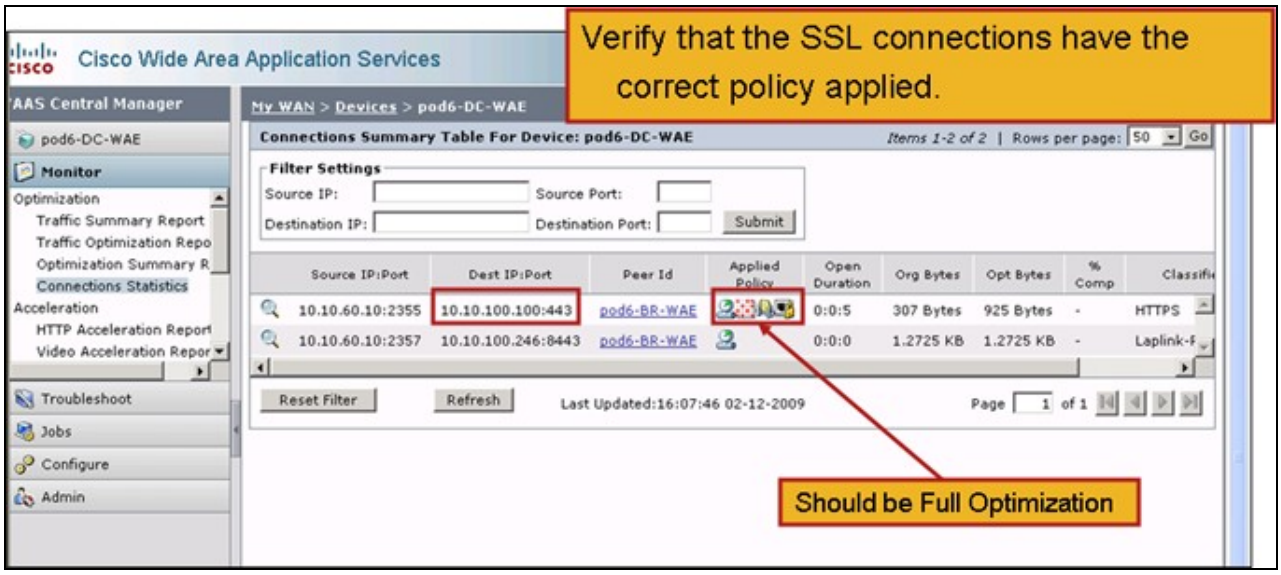
If there is a problem with the disk encryption on a WAE, that can also prevent the SSL AO from operating. Use the **show disk details** command to verify that disk encryption is enabled and check if the CONTENT and SPOOL partitions are mounted. If these partitions are mounted, it indicates that the disk encryption keys were successfully retrieved from the Central Manager and encrypted data can be written and read from the disks. If the **show disk details** command shows "System is initializing," that indicates the encryption keys have not yet been retrieved from the Central Manager and the disks have not yet been mounted. The WAE will not provide acceleration services in this state. If the WAE is unable to retrieve disk encryption keys from the Central Manager, it will raise an alarm.

You can verify that the SSL accelerated service is configured and its status is "Enabled" on the data center WAE (in the Central Manager, choose the device, then choose **Configure > Acceleration > SSL Accelerated Services**). A configured and enabled accelerated service may be rendered inactive by the SSL accelerator due to the following conditions:

- The certificate configured in the accelerated service been deleted from the WAE. Use the **show running-config** command to determine the certificate being used in the accelerated service, then use the **show crypto certificates** and **show crypto certificate-details** commands to confirm that the certificate is present secure store. If the certificate is missing, reimport the certificate.
- The accelerated service certificate has expired. Use the **show crypto certificates** and **show crypto certificate-details** commands to check the certificate expiry date.
- The accelerated service certificate has a valid date starting in the future. Use the **show crypto certificates** and **show crypto certificate-details** commands and check the validity section of the command output. Also, ensure that the WAE clock and timezone information is accurate.

You can verify that SSL connections have the correct policy applied, that is, they have full optimization with SSL acceleration, as shown in Figure 2. In the Central Manager, choose the WAE device, then choose **Monitor > Optimization > Connections Statistics**.

Figure 2. Verifying the Correct Policy on SSL Connections



Use the **show running-config** command to verify that the HTTPS traffic policy is properly configured. You want to see **optimize DRE no compression none** for the SSL application action and you want to see

appropriate match conditions listed for the HTTPS classifier, as follows:

```
WAE674# sh run | include HTTPS
   classifier HTTPS
      name SSL classifier HTTPS action optimize DRE no compression none              <----------

WAE674# sh run | begin HTTPS

...skipping
   classifier HTTPS
      match dst port eq 443                                                          <----------
   exit
```

An active accelerated service inserts dynamic policies corresponding to the server IP:port, server name:port, or server domain:port configured within the accelerated service. These policies can be inspected using the **show policy-engine application dynamic** command. The Dst field in each displayed policy indicates the server IP and port matching the accelerated service. For the wildcard domain (for example, server-domain *.webex.com port 443), the Dst field will be 'Any:443'. For the server-name configuration, forward DNS lookup is performed when the accelerated service is activated and all the IP addresses returned in the DNS response will be inserted in the policy engine. This command is useful to catch situations where an accelerated service is marked "inservice" but the accelerated service is rendered inactive because of some other error. For example, all accelerated services are dependent on the peering service, and if the peering service is inactive because of a missing/deleted certificate, then an accelerated service will also be marked as inactive although it appears to be "inservice" in the show running-config output. You can verify that the SSL dynamic policy is active on the data center WAE by using the **show policy-engine application dynamic** command. You can verify the peering service status by using the **show crypto ssl services host-service peering** command.

An SSL AO accelerated service configuration can have four types of server entries:

- Static IP (server-ip)--available in version 4.1.3 and later
- Catch All (server-ip any)--available in 4.1.7 and later
- Hostname (server-name)--available in 4.2.1 and later
- Wildcard domain (server-domain)--available in 4.2.1 and later

Once the connection is received by the SSL AO, it decides which accelerated service should be used for optimization. The static IP configuration is given the highest preference, followed by server name, server domain, and then the server ip any. If none of the configured and activated accelerated services match with the server IP for the connection, the connection is pushed down to the generic AO. The cookie inserted into the policy engine by the SSL AO is used to determine which accelerated service and what type of server entry is matched for a particular connection. This policy engine cookie is a 32-bit number and is meaningful only to the SSL AO. The higher bits are used to indicate different server entry types and the lower bits indicate the accelerated service index, as follows:

SSL Policy Engine Cookie Values

Cookie Value	Server Entry Type	Comments
0x8*xxxxxxx*	Server IP address	Static IP address configuration
0x4*xxxxxxx*	Server hostname	Data center WAE performs a forward DNS lookup for the hostname and it adds the IP addresses that are returned into the dynamic policy configuration.

		Refreshed every 10 minutes by default.
0x2FFFFFFF	Server domain name	Data center WAE performs a reverse DNS lookup on the destination host IP address to determine if it matches with the domain. If it matches, then SSL traffic is accelerated, and if it does not match, the traffic is handled according to the static HTTPS policy.
0x1*xxxxxxx*	Server Any	All SSL connections are accelerated using this accelerated service configuration

Example 1: Accelerated Service with server-ip Configuration:

```
WAE(config)#crypto ssl services accelerated-service asvc-ip
WAE(config-ssl-accelerated)#description "Server IP acceleration"
WAE(config-ssl-accelerated)#server-cert-key server.p12
WAE(config-ssl-accelerated)#server-ip 171.70.150.5 port 443
WAE(config-ssl-accelerated)#inservice
```

The corresponding policy engine entry is added as follows:

```
WAE# sh policy-engine application dynamic
Dynamic Match Freelist Information:
  Allocated: 32768  In Use: 3  Max In Use: 5  Allocations: 1751

< snip >

Individual Dynamic Match Information:
  Number:     1   Type: Any->Host (6)  User Id: SSL (4)           <----------------
    Src: ANY:ANY  Dst: 171.70.150.5:443                          <----------------
    Map Name: basic
    Flags: SSL
    Seconds: 0  Remaining: - NA -  DM Index: 32764
    Hits: 25  Flows: - NA -  Cookie: 0x80000001                  <----------------
```

Example 2: Accelerated Service with server-name Configuration:

This configuration allows easy deployment for optimization of enterprise SSL applications. It is adaptable to DNS configuration changes and reduces IT administrative tasks.

```
WAE(config)#crypto ssl services accelerated-service asvc-name
WAE(config-ssl-accelerated)#description "Server name acceleration"
WAE(config-ssl-accelerated)#server-cert-key server.p12
WAE(config-ssl-accelerated)#server-name www.google.com port 443
WAE(config-ssl-accelerated)#inservice
```

The corresponding policy engine entry is added as follows:

```
WAE# sh policy-engine application dynamic
Dynamic Match Freelist Information:
  Allocated: 32768  In Use: 3  Max In Use: 5  Allocations: 1751

< snip >

Individual Dynamic Match Information:
 Number:     1   Type: Any->Host (6)  User Id: SSL (4)           <----------------
   Src: ANY:ANY  Dst: 74.125.19.104:443                         <----------------
   Map Name: basic
   Flags: SSL
   Seconds: 0  Remaining: - NA -  DM Index: 32762
   Hits: 0  Flows: - NA -  Cookie: 0x40000002                   <----------------
```

```
   DM Ref Index: - NA -  DM Ref Cnt: 0
 Number:     2   Type: Any->Host (6)  User Id: SSL (4)            <----------------
   Src: ANY:ANY  Dst: 74.125.19.147:443                           <----------------
   Map Name: basic
   Flags: SSL
   Seconds: 0  Remaining: - NA -  DM Index: 32763
   Hits: 0  Flows: - NA -  Cookie: 0x40000002                     <----------------
   DM Ref Index: - NA -  DM Ref Cnt: 0
 Number:     3   Type: Any->Host (6)  User Id: SSL (4)            <----------------
   Src: ANY:ANY  Dst: 74.125.19.103:443                           <----------------
   Map Name: basic
   Flags: SSL
   Seconds: 0  Remaining: - NA -  DM Index: 32764
   Hits: 0  Flows: - NA -  Cookie: 0x40000002                     <----------------
   DM Ref Index: - NA -  DM Ref Cnt: 0
 Number:     4   Type: Any->Host (6)  User Id: SSL (4)            <----------------
   Src: ANY:ANY  Dst: 74.125.19.99:443                            <----------------
   Map Name: basic
   Flags: SSL
   Seconds: 0  Remaining: - NA -  DM Index: 32765
   Hits: 0  Flows: - NA -  Cookie: 0x40000002                     <----------------
   DM Ref Index: - NA -  DM Ref Cnt: 0
```

Example 3: Accelerated Service with server-domain Configuration:

This configuration allows WAAS devices to configure a single wildcard domain that avoids the need to know IP addresses for all the servers. The data center WAE uses reverse DNS (rDNS) to match traffic belonging to the configured domain. Configuring a wildcard domain avoids configuring multiple IP addresses, making the solution scalable and applicable for SaaS architecture.

```
WAE(config)#crypto ssl services accelerated-service asvc-domain
WAE(config-ssl-accelerated)#description "Server domain acceleration"
WAE(config-ssl-accelerated)#server-cert-key server.p12
WAE(config-ssl-accelerated)#server-name *.webex.com port 443
WAE(config-ssl-accelerated)#inservice
```

The corresponding policy engine entry is added as follows:

```
WAE# sh policy-engine application dynamic
Dynamic Match Freelist Information:
  Allocated: 32768  In Use: 3  Max In Use: 5  Allocations: 1751

< snip >

Individual Dynamic Match Information:
 Number:     1   Type: Any->Host (6)  User Id: SSL (4)            <----------------
   Src: ANY:ANY  Dst: ANY:443                                     <----------------
   Map Name: basic
   Flags: SSL
   Seconds: 0  Remaining: - NA -  DM Index: 32762
   Hits: 0  Flows: - NA -  Cookie: 0x2FFFFFFF                     <----------------
   DM Ref Index: - NA -  DM Ref Cnt: 0
```

Example 4: Accelerated Service with server-ip any Configuration:

This configuration provides a catch-all mechanism. When an accelerated service with **server-ip any port 443** is made active, it allows all connections on port 443 to be optimized by the SSL AO. This configuration can be used during POCs to optimize all traffic on a particular port.

```
WAE(config)#crypto ssl services accelerated-service asvc-ipany
```

```
WAE(config-ssl-accelerated)#description "Server ipany acceleration"
WAE(config-ssl-accelerated)#server-cert-key server.p12
WAE(config-ssl-accelerated)#server-ip any port 443
WAE(config-ssl-accelerated)#inservice
```

The corresponding policy engine entry is added as follows:

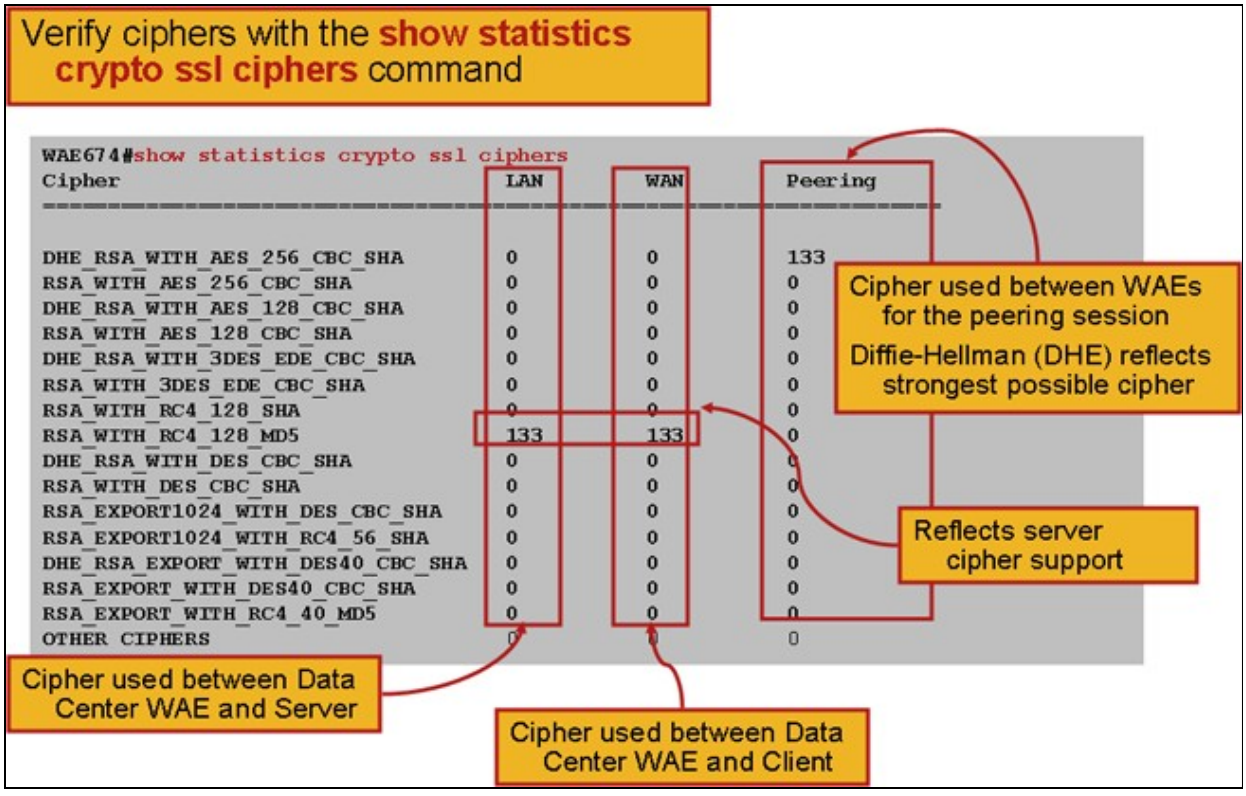
```
WAE# sh policy-engine application dynamic
Dynamic Match Freelist Information:
  Allocated: 32768  In Use: 3  Max In Use: 5  Allocations: 1751

< snip >

Individual Dynamic Match Information:
 Number:     1   Type: Any->Host (6)  User Id: SSL (4)            <----------------
   Src: ANY:ANY  Dst: ANY:443                                     <----------------
   Map Name: basic
   Flags: SSL
   Seconds: 0  Remaining: - NA -  DM Index: 32762
   Hits: 0  Flows: - NA -  Cookie: 0x10000004                     <----------------
   DM Ref Index: - NA -  DM Ref Cnt: 0
```

You can verify the ciphers being used with the **show statistics crypto ssl ciphers** commands, as shown in Figure 3.

Figure 3. Verifying Ciphers



You can verify that these ciphers match those configured on the origin server. **Note**: Ciphers that include DHE are not supported by Microsoft IIS servers.

On an Apache server, you can verify the SSL version and cipher details in the httpd.conf file. These fields may also be in a separate file (sslmod.conf) referenced from httpd.conf. Look for the SSLProtocol and

SSLCipherSuite fields as follows:

```
SSLProtocol -all +TLSv1 +SSLv3
SSLCipherSuite HIGH:MEDIUM:!aNULL:+SHA1:+MD5:+HIGH:+MEDIUM
. . .
SSLCertificateFile /etc/httpd/ssl/server.crt
SSLCertificateKeyFile /etc/httpd/ssl/server.key
```

To verify the certificate issuer on an Apache server, use the openssl command to read the certificate as follows:

```
> openssl x509 -in cert.pem -noout -issuer -issuer_hash
issuer= / C=US/ST=California/L=San Jose/O=CISCO/CN=tools.cisco.com/emailAddress=webmaster@cisco.co
```

In the browser, you can view a certificate and its details to determine the certificate chain, version, encryption key type, issuer common name (CN), and subject/site CN. In Internet Explorer, click the padlock icon, click **View Certificate**, and then look at the Details and Certification Path tabs for this information.

Most browsers require that client certificates be in the PKCS12 format rather than the X509 PEM format. To export the X509 PEM format to PKCS12 format, use the openssl command as follows on an Apache server:

```
> openssl pkcs12 -export -in cert.pem -inkey key.pem -out cred.p12
Enter Export Password:
Verifying - Enter Export Password:
```

If the private keys are encrypted, the passphrase is required for export. The export password is used again for importing credentials to the WAAS device.

Use the **show statistics accelerator ssl** command to see the SSL AO statistics.

```
WAE7326# show statistics accelerator ssl
SSL:

  Global Statistics
  -----------------
  Time Accelerator was started:                       Mon Nov 10     15:28:47 2008
  Time Statistics were Last Reset/Cleared:            Mon Nov 10    15:28:47 2008
  Total Handled Connections:                                          17                <---------
  Total Optimized Connections:                                        17                <---------
  Total Connections Handed-off with Compression Policies Unchanged:   0                 <---------
  Total Dropped Connections:                                          0                 <---------
  Current Active Connections:                                         0
  Current Pending Connections:                                        0
  Maximum Active Connections:                                         3
  Total LAN Bytes Read:                                               25277124          <---------
  Total Reads on LAN:                                                 5798              <---------
  Total LAN Bytes Written:                                            6398              <---------
  Total Writes on LAN:                                                51                <---------
  Total WAN Bytes Read:                                               43989             <---------
  Total Reads on WAN:                                                 2533              <---------
  Total WAN Bytes Written:                                            10829055          <---------
  Total Writes on WAN:                                                3072              <---------
. . .
```

Failed sessions and certificate verifications statistics can be useful for troubleshooting and are more easily retrieved by using the following filter on the **show statistics accelerator ssl** command:

```
WAE# show statistics accelerator ssl | inc Failed
  Total Failed Handshakes:                                          47
```

```
  Total Failed Certificate Verifications:                          28
  Failed certificate verifications due to invalid certificates:    28
  Failed Certificate Verifications based on OCSP Check:            0
  Failed Certificate Verifications (non OCSP):                     28
  Total Failed Certificate Verifications due to Other Errors:      0
  Total Failed OCSP Requests:                                      0
  Total Failed OCSP Requests due to Other Errors:                  0
  Total Failed OCSP Requests due to Connection Errors:             0
  Total Failed OCSP Requests due to Connection Timeouts:           0
  Total Failed OCSP Requests due to Insufficient Resources:        0
```

DNS related statistics can be useful for troubleshooting server name and wildcard domain configuration. To retrieve these statistics use the **show statistics accelerator ssl** command, as follows:

```
WAE# show statistics accelerator ssl
 . . .
 Number of forward DNS lookups issued:                           18
 Number of forward DNS lookups failed:                           0
 Number of flows with matching host names:                       8
 Number of reverse DNS lookups issued:                           46
 Number of reverse DNS lookups failed:                           4
 Number of reverse DNS lookups cancelled:                        0
 Number of flows with matching domain names:                     40
 Number of flows with matching any IP rule:                      6
 . . .
 Pipe-through due to domain name mismatch:                       6
 . . .
```

SSL rehandshake related statistics can be useful for troubleshooting and can be retrieved using the following filter on the **show statistics accelerator ssl** command:

```
WAE# show statistics accelerator ssl | inc renegotiation
  Total renegotiations requested by server:                        0
  Total SSL renegotiations attempted:                              0
  Total number of failed renegotiations:                           0
  Flows dropped due to renegotiation timeout:                      0
```

Use the **show statistics connection optimized ssl** command to check that the WAAS device is establishing optimized SSL connections. Verify that "TDLS" appears in the Accel column for a connection. "S" indicates that the SSL AO was used as follows:

```
WAE674# sh stat conn opt ssl
Current Active Optimized Flows:                       3
   Current Active Optimized TCP Plus Flows:           3
   Current Active Optimized TCP Only Flows:           0
   Current Active Optimized TCP Preposition Flows:    1
Current Active Auto-Discovery Flows:                  0
Current Active Pass-Through Flows:                    0
Historical Flows:                                     100

D:DRE,L:LZ,T:TCP Optimization,
A:AOIM,C:CIFS,E:EPM,G:GENERIC,H:HTTP,M:MAPI,N:NFS,S:SSL,V:VIDEO

ConnID  Local IP:Port         Remote IP:Port        PeerID             Accelerator
342     10.56.94.101:3406    10.10.100.100:443      0:1a:64:d3:2f:b8   TDLS                <-----
```

You can check connection statistics for closed connections by using the **show statistics connection closed ssl** command.

If the connections are not getting optimized, check if WCCP/PBR is properly configured and working, and check for asymmetric routing.

You can view the SSL connection statistics by using the **show statistics connection optimized ssl detail** command, where you will see the dynamic policy that results from the configured SSL accelerated service. **Note**: The configured policy is TFO optimization only, but full optimization is applied as a result of the configured SSL service.

```
WAE674# sh stat connection optimized ssl detail
Connection Id:            1633
   Peer Id:                  00:14:5e:84:24:5f
   Connection Type:          EXTERNAL CLIENT
   Start Time:               Wed Jul 15 06:35:48 2009
   Source IP Address:        10.10.10.10
   Source Port Number:       2199
   Destination IP Address:   10.10.100.100
   Destination Port Number:  443
   Application Name:         SSL
   Classifier Name:          HTTPS
   Map Name:                 basic
   Directed Mode:            FALSE
   Preposition Flow:         FALSE
   Policy Details:
          Configured:        TCP_OPTIMIZE                                    <------TFO only is
             Derived:        TCP_OPTIMIZE + DRE + LZ
                Peer:        TCP_OPTIMIZE
          Negotiated:        TCP_OPTIMIZE + DRE + LZ
             Applied:        TCP_OPTIMIZE + DRE + LZ                         <------Full optimi
   Accelerator Details:
               Configured:   None
                  Derived:   None
                  Applied:   SSL                                             <------SSL acceler
                     Hist:   None

                                        Original            Optimized
                                  ------------------- -------------------
   Bytes Read:                               1318                 584
   Bytes Written:                             208                1950
. . .
```

Later in this output, extended SSL session level details are shown as follows:

```
. . .
SSL : 1633

  Time Statistics were Last Reset/Cleared:                  Tue Jul 10 18:23:20 2009
  Total Bytes Read:                                         0          0
  Total Bytes Written:                                      0          0
  Memory address:                                           0x8117738
  LAN bytes read:                                           1318
  Number of reads on LAN fd:                                4
  LAN bytes written out:                                    208
  Number of writes on LAN fd:                               2
  WAN bytes read:                                           584
  Number of reads on WAN fd:                                23
  WAN bytes written out:                                    1950
  Number of writes on WAN fd:                               7
  LAN handshake bytes read:                                 1318
  LAN handshake bytes written out:                          208
  WAN handshake bytes read:                                 542
  WAN handshake bytes written out:                          1424
  AO bytes read:                                            0
```

```
Number of reads on AO fd:                                    0
AO bytes written out:                                        0
Number of writes on AO fd:                                   0
DRE bytes read:                                              10
Number of reads on DRE fd:                                   1
DRE bytes written out:                                       10
Number of writes on DRE fd:                                  1
Number of renegotiations requested by server:                0
Number of SSL renegotiations performed:                      0
Flow state:                                                  0x00080000
LAN work items:                                              1
LAN conn state:                                              READ
LAN SSL state:                                               SSLOK (0x3)
WAN work items:                                              0
WAN conn state:                                              READ
WAN SSL state:                                               SSLOK (0x3)
W2W work items:                                              1
W2W conn state:                                              READ
W2W SSL state:                                               SSLOK (0x3)
AO  work items:                                              1
AO  conn state:                                              READ
DRE work items:                                              1
DRE conn state:                                              READ
Hostname in HTTP CONNECT:                                                  <-----Added
IP Address in HTTP CONNECT:                                                <-----Added
TCP Port in HTTP CONNECT:                                                  <-----Added
```

Troubleshooting HTTP AO to SSL AO Handoff Connections

If a client must go through a proxy to reach an HTTPS server, the client?s request first goes as an HTTP CONNECT message to the proxy (with the actual HTTPS server IP address embedded in the CONNECT message). At this point, the HTTP AO handles this connection on the peer WAEs. The proxy creates a tunnel between the client and server port and relays subsequent data between the client and that server IP address and port. The proxy responds back to the client with a ?200 OK? message and hands off the connection to the SSL AO because the client intends to talk to the server over SSL. The client then initiates an SSL handshake with the SSL server over the TCP connection (tunnel) that was set up by the proxy.

Check the following things when troubleshooting issues with handed-off connections:

- Check the output of the **show statistics accelerator http** command to confirm that a connection was handled by the HTTP AO and then handed off to the SSL AO. Look at the Total Handled Connections and Total Connections Handed-off to SSL counters. If there are any issues, verify the following:
    - The HTTP AO is enabled and in the running state on the peer WAEs.
    - The SSL accelerated service is configured with the port used by the client in the CONNECT URL (or implied port 443 if HTTPS is being used). Often the proxy port is different from the CONNECT URL port and this proxy port should not be configured in the SSL accelerated service. However, the proxy port should be included in the traffic classifier that is mapped to the HTTP AO.
- Check the output of the **show statistics accelerator http** command to confirm that this connection was handled and optimized by the SSL AO. Look at the Total Handled Connections and Total Optimized Connections counters. If the statistics counters are not correct, perform basic SSL troubleshooting as discussed in the previous section.
- On the data center WAE, verify that the **show statistics connection optimized detail** command output shows the actual SSL server?s hostname, IP address, and TCP port. If these fields are not set correctly, check the following:
    - Verify that the client browser proxy settings are correct.

- Verify that the DNS server is configured on the data center WAE and is reachable. You can configure a DNS server on the WAE with the **ip name-server *A.B.C.D*** command.

Troubleshooting Server Certificate Verification

Server certificate verification requires that you import the correct CA certificate to the data center WAE.

To troubleshoot server certificate verification follow these steps:

1. Inspect the server certificate and retrieve the Issuer name. This Issuer name within the server certificate must match the subject name within the matching CA certificate. If you have PEM encoded certificates, you can use the following **openssl** command on a server with openssl installed:

```
> openssl x509 ?in cert-file-name ?noout ?text
```

2. Ensure that the matching crypto pki ca configuration exists on the data center WAE by using **show running-config** command. For a CA certificate to be used by the WAE in the verification process, a crypto pki ca configuration item is required for each CA certificate imported. For example, if a CA certificate company1.ca is imported, then the following configuration must be made on the data center WAE:

```
crypto pki ca company1
   ca-certificate company1.ca
   exit
```

Note: If a CA certificate is imported using the Central Manager GUI, the Central Manager automatically adds the above crypto pki ca configuration to include the imported CA certificate. However, if the CA certificate is imported via the CLI, then you will need to manually add the above configuration.

3. If the certificate being verified includes a certificate chain, then ensure that the certificate chain is coherent, and the topmost issuer?s CA certificate is imported on the WAE. Use the **openssl verify** command to verify the certificate separately first.

4. If verification still fails, then examine the SSL accelerator debug log. Use the following commands to enable debug logging:

```
wae# config
wae(config)# logging disk priority debug
wae(config)# logging disk enable
wae(config)# exit
wae# undebug all
wae# debug accelerator ssl verify
wae# debug tfo connection all
```

5. Initiate a test connection and then examine the /local/local1/errorlog/sslao-errorlog.current log file. This file should indicate the issuer name that was included in the server certificate. Ensure that this issuer name exactly matches the subject name of the CA certificate.

If there are any other internal errors in the logs, it may be helpful to enable additional debug options.

6. Even if the Issuer name and Subject names match, the CA certificate may not be the correct one. In such cases, if the server certificate is issued by a well-known CA, then a browser can be used to directly (without WAAS) reach the server. When the browser sets up the connection, the certificate can be examined by clicking the Lock icon that appears on the bottom right of the browser window or within the browser?s address bar. The certificate details may indicate the appropriate CA certificate matching this server certificate. Check the Serial Number field within the CA certificate. This serial number should match the

serial number of the certificate that is being imported on the data center WAE.

7. If you have OCSP revocation checking enabled, disable it and check that certificate verification by itself works. For help troubleshooting OCSP settings see the "Troubleshooting OCSP Revocation Checking" section.

Troubleshooting Client Certificate Verification

Verification of the client certificate may be enabled on the origin server and/or on the data center WAE. When WAAS is used to accelerate SSL traffic, the client certificate received by the origin server is the certificate indicated in the machine-cert-key specified in the **crypto ssl services global-settings** command on the data center WAE or the data center WAE machine self signed certificate, if the machine-cert-key is not configured. As a result, if client certificate verification is failing on the origin server, it may be because the data center WAE machine-certificate is not verifiable on the origin server.

If client certificate verification on the data center WAE is not working, it is likely because the CA certificate matching the client certificate is not imported on the data center WAE. See the "Troubleshooting Server Certificate Verification" section for instructions how to check if you have the correct CA certificate imported on the WAE.

Troubleshooting Peer WAE Certificate Verification

To troubleshoot peer certificate verification issues follow these steps:

1. Verify that the certificate being verified is a CA signed certificate. A self signed certificate by one WAE is not verifiable by another WAE. WAEs by default are loaded with self signed certificates. A self signed certificate must be configured using the **crypto ssl services global-settings machine-cert-key** command.

2. Verify that the correct CA certificate is loaded on the device that is verifying the certificate. For example, if peer-cert-verify is configured on the data center WAE, then it is essential for the branch WAE certificate to be CA-signed and the same signing CA?s certificate should be imported on the data center WAE. Do not forget to create a CA using the **crypto pki ca** command to use the imported certificate, if you are importing the certificate manually through the CLI. When imported by the Central Manager GUI, the Central Manager automatically creates a matching crypto pki ca configuration.

3. If verification of the peer WAE still fails, check the debug logs as described in the "SSL AO Logging" section.

Troubleshooting OCSP Revocation Checking

If the system is having trouble making successful SSL connections with Online Certificate Status Protocol (OCSP) revocation checking enabled, follow these troubleshooting steps:

1. Ensure that the OCSP responder service is running on the responder server.
2. Ensure good connectivity between the WAE and the responder. Use the **ping** and **telnet** commands (to the appropriate port) from the WAE to check.
3. Confirm that the certificate being validated is indeed valid. The expiry date and correct responder URL are typically areas where there are issues.
4. Verify that the certificate for OCSP responses is imported on the WAE. Responses from an OCSP responder are also signed and the CA certificate matching the OCSP responses must reside on the WAE.
5. Check the **show statistics accelerator ssl** command output to check for OCSP statistics and check

the counters corresponding to OCSP failures.

6. If the OCSP HTTP connection is going through an HTTP proxy, try disabling the proxy to see if it helps. If it does help, then check that the proxy configuration is not causing the connection failure. If the proxy configuration is fine, then there may be some HTTP header peculiarity which may be causing some incompatibility with the proxy. Capture a packet trace for further investigation.
7. If all else fails, you may have to capture a packet trace of the outgoing OCSP request for further debugging. You can use the **tcpdump** or **tethereal** commands as described in the section "Capturing and Analyzing Packets" in the Preliminary WAAS Troubleshooting article.

The URL used by the data center WAE to reach an OCSP responder is derived in one of two ways:

- The static OCSP URL configured by the **crypto pki global-settings** configuration command
- The OCSP URL specified in the certificate being checked

If the URL is derived from the certificate being checked, then it is essential to ensure that the URL is reachable. Enable the SSL accelerator OCSP debug logs to determine the URL and then check for connectivity to the responder. See the next section for details on using debug logs.

Troubleshooting DNS Configuration

If the system is having trouble optimizing SSL connections with server name and server domain configurations, follow these troubleshooting steps:

1. Ensure that the DNS server configured on the WAE is reachable and can resolve names. Use the following command to check the configured DNS server:

```
WAE# sh running-config | include name-server
ip name-server 2.53.4.3

Try to perform DNS or reverse DNS lookup on the WAE using the following commands:

WAE# dnslookup www.cisco.com
The specified host/domain name is unknown !
```

This response indicates the name cannot be resolved by the configured name servers.

Try ping/traceoute for the configured name servers to check their reachability and the round trip time.

```
WAE# ping 2.53.4.3
PING 2.53.4.3 (2.53.4.3) 56(84) bytes of data.
--- 2.53.4.3 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4008ms

WAE# traceroute 2.53.4.3
traceroute to 2.53.4.3 (2.53.4.3), 30 hops max, 38 byte packets
1  2.53.4.33 (2.53.4.33)  0.604 ms  0.288 ms  0.405 ms
2  * * *
3  * * *
4  * * *
5  * * *
```

2. If the DNS server is reachable and it can resolve names and still the SSL connections are not getting optimized, make sure the accelerated service configuring the specified domain or hostname is active and there are no alarms for the SSL AO. Use following commands:

```
WAE#  show alarms
```

```
 Critical Alarms:
 ----------------
       Alarm ID                 Module/Submodule                Instance
  ---------------            --------------------          ---------------
   1 accl_svc_inactive         sslao/ASVC/asvc-host         accl_svc_inactive
   2 accl_svc_inactive         sslao/ASVC/asvc-domain       accl_svc_inactive

Major Alarms:
-------------
None

Minor Alarms:
-------------
None
```

The presence of the "accl_svc_inactive" alarm is an indication that there is some discrepancy in the accelerated service configuration and there might be one or more accelerated services having overlapping configuration for server entries. Check the accelerated service configuration and make sure the configuration is correct. Use the following command to verify the configuration:

```
 WAE# show crypto ssl accelerated service
 Accelerated Service     Config State    Oper State      Cookie
 -------------------     ------------    -----------     ------
 asvc-ip                  ACTIVE          ACTIVE          0
 asvc-host                ACTIVE          INACTIVE        1
 asvc-domain              ACTIVE          INACTIVE        2
```

To check details about a particular accelerated service use the following command:

```
WAE# show crypto ssl accelerated service asvc-host
 Name: asvc-host
   Config state: ACTIVE, Oper state: INACTIVE, Cookie: 0x3, Error vector: 0x0
   No server IP addresses are configured
   The following server host names are configured:
     lnxserv.shilpa.com port 443
         Host 'lnxserv.shilpa.com' resolves to following IPs:
         --none--
   No server domain names are configured
```

One reason that the operational state of the accelerated service might be INACTIVE is a DNS failure. For example, if there is a server hostname in the accelerated service configuration and the WAE cannot resolve the server IP address, then it cannot configure the appropriate dynamic policy.

3. If the statistics counter for ?Pipe-through due to non-matching domain name? is increasing, it is an indication that the SSL connection is for a server that is configured for optimization. Check the policy engine entries using following command:

```
WAE#sh policy-engine application dynamic
       Number:     1   Type: Any->Host (6)  User Id: SSL (4)
       Src: ANY:ANY  Dst: 2.53.4.2:443
       Map Name: basic
       Flags: TIME_LMT DENY
       Seconds: 10  Remaining: 5  DM Index: 32767
       Hits: 1  Flows: - NA -  Cookie: 0x2EEEEEEE
       DM Ref Index: - NA -  DM Ref Cnt: 0
```

Check the connection status using the **show statistics connection** command. The first connection should show an Accelerator of TSGDL and the subsequent connections, until the lifetime of the TIME_DENY policy entry, should be TDL.

4. If the DNS server is across the WAN with respect to the data center WAE, or if the reverse DNS response time is too long, then some connections may be dropped. This depends on the client timeout and the rDNS response time. In this case, the counter for ?Number of reverse DNS lookups cancelled? increases and the connection is dropped. This situation is an indication that the DNS server is not responsive or very slow and/or NSCD on WAAS is not working. The NSCD status can be checked using the **show alarms** command. The probability of this happening is very low since in most deployments, the DNS server is expected to be on the same LAN as the data center WAE.

Troubleshooting HTTP to SSL AO Chaining

NOTE: HTTP to SSL AO chaining was introduced in WAAS version 4.3.1. This section is not applicable to earlier WAAS versions.

Chaining allows an AO to insert another AO at any time during the lifetime of a flow and both AOs can apply their AO-specific optimization independently on the flow. AO chaining is different from the AO handoff feature provided by WAAS in pre-4.3.1 releases because with AO chaining the first AO continues to optimize the flow.

The SSL AO handles two types of connections:

- Byte-0 SSL: The SSL AO receives the connection first and completes the SSL handshake. It parses the initial part of the payload to check for an HTTP method. If the payload indicates HTTP, it inserts the HTTP AO; if not, it applies the regular TSDL optimization.
- Proxy CONNECT: The HTTP AO receives the connection first. It identifies the CONNECT header method in the client?s request and inserts the SSL AO after the proxy confirms with a 200 OK message.

The SSL AO uses a lightweight HTTP parser that detects the following HTTP methods: GET, HEAD, POST, PUT, OPTIONS, TRACE, COPY, LOCK, POLL, BCOPY, BMOVE, MKCOL, DELETE, SEARCH, UNLOCK, BDELETE, PROPFIND, BPROPFIND, PROPPATCH, SUBSCRIBE, BPROPPATCH, UNSUBSCRIBE, and X__MS_ENUMATTS. You can use the **debug accelerator ssl parser** command to debug issues related to the parser. You can use the **show stat accel ssl payload http/other** command to view statistics of traffic classified based on the payload type.

Troubleshooting tips:

1. Make sure the HTTPS feature is enabled in the HTTP AO configuration as this is owned by the HTTP AO. For details, see the Troubleshooting the HTTP AO article.
2. Check the connection state using the **show stat connection** command. If correctly optimized, it should show THSDL indicating TCP, HTTP, SSL and DRE-LZ optimization. If any of these optimizations are missing, debug further on that optimizer (SSL, HTTP, and so forth). For example, if the connection state shows THDL, it means SSL optimization was not applied on the connection. Details on debugging issues related to the SSL AO follow.
3. Make sure the SSL AO is enabled and is in the running state (see the section "Troubleshooting the SSL AO").
4. Make sure there are no alarms by using the **show alarms** command.
5. If SSL traffic is not being optimized, make sure the server IP address, host-name, or domain-name and port number is added as part of the accelerated service.
6. Make sure the accelerated service is in the ACTIVE state by using the **show crypto ssl services accelerated-service** ***ASVC-name*** command (see the "Troubleshooting DNS Configuration" section).
7. Make sure the policy engine has an entry for this server and port by using the **show policy-engine application dynamic** command.

8. If the destination server is using SSL on a non-default port (the default is 443), make sure this is reflected in the policy engine configuration. The Central Manager relies on this information for reporting SSL traffic data.
9. Make sure the configured host-name resolves to a valid IP address by using the **show crypto ssl services accelerated-service** ***ASVC-name*** command. If no IP address is found, check if the name server is configured correctly. Also check the output of the **dnslookup** ***IP-address*** command.

```
wae# sh run no-policy
. . .
crypto ssl services accelerated-service sslc
  version all
  server-cert-key test.p12
  server-ip 2.75.167.2 port 4433
  server-ip any port 443
  server-name mail.yahoo.com port 443
  server-name mail.google.com port 443
  inservice

wae# sh crypto ssl services accelerated-service sslc
Name: sslc
Config state: ACTIVE, Oper state: ACTIVE, Cookie: 0x0, Error vector: 0x0

   The following server IP addresses are configured:
     2.75.167.2 port 4433
     any port 443

   The following server host names are configured:
     mail.yahoo.com port 443
         Host 'mail.yahoo.com' resolves to following IPs:
         66.163.169.186

     mail.google.com port 443
         Host 'mail.google.com' resolves to following IPs:
         74.125.19.17
         74.125.19.18
         74.125.19.19
         74.125.19.83

wae# dnslookup mail.yahoo.com
Official hostname: login.lga1.b.yahoo.com
         address: 66.163.169.186
Aliases: mail.yahoo.com
Aliases: login.yahoo.com
Aliases: login-global.lgg1.b.yahoo.com

wae# dnslookup mail.google.com
Official hostname: googlemail.l.google.com
         address: 74.125.19.83
         address: 74.125.19.17
         address: 74.125.19.19
         address: 74.125.19.18
Aliases: mail.google.com
```

SSL AO Logging

The following log files are available for troubleshooting SSL AO issues:

- Transaction log files: /local1/logs/tfo/working.log (and /local1/logs/tfo/tfo_log_*.txt)
- Debug log files: /local1/errorlog/sslao-errorlog.current (and sslao-errorlog.*)

For easier debugging, you should first set up an ACL to restrict packets to one host.

```
WAE674(config)# ip access-list extended 150 permit tcp host 10.10.10.10 any
WAE674(config)# ip access-list extended 150 permit tcp any host 10.10.10.10
```

To enable transaction logging, use the **transaction-logs** configuration command as follows:

```
wae(config)# transaction-logs flow enable
wae(config)# transaction-logs flow access-list 150
```

You can view the end of a transaction log file by using the **type-tail** command as follows:

```
wae# type-tail tfo_log_10.10.11.230_20090715_130000.txt
Wed Jul 15 14:35:48 2009 :1633 :10.10.10.10 :2199 :10.10.100.100 :443 :OT :START  :EXTERNAL CLIENT
 :SSL :HTTPS :F :(TFO) (DRE,LZ,TFO) (TFO) (DRE,LZ,TFO) (DRE,LZ,TFO) :<None> :(None) (None) (SSL) :
Wed Jul 15 14:36:06 2009 :1633 :10.10.10.10 :2199 :10.10.100.100 :443 :SODRE :END :165 :15978764 :
Wed Jul 15 14:36:06 2009 :1633 :10.10.10.10 :2199 :10.10.100.100 :443 :OT :END :EXTERNAL CLIENT :(
```

To set up and enable debug logging of the SSL AO, use the following commands.

NOTE: Debug logging is CPU intensive and can generate a large amount of output. Use it judiciously and sparingly in a production environment.

You can enable detailed logging to the disk as follows:

```
WAE674(config)# logging disk enable
WAE674(config)# logging disk priority detail
```

You can enable debug logging for connections in the ACL as follows:

```
WAE674# debug connection access-list 150
```

The options for SSL AO debugging are as follows:

```
WAE674# debug accelerator ssl ?
 accelerated-svc   enable accelerated service debugs
 alarm             enable SSL AO alarm debugs
 all               enable all SSL accelerator debugs
 am                enable auth manager debugs
 am-generic-svc    enable am generic service debugs
 bio               enable bio layer debugs
 ca                enable cert auth module debugs
 ca-pool           enable cert auth pool debugs
 cipherlist        enable cipherlist debugs
 client-to-server  enable client-to-server datapath debugs
 dataserver        enable dataserver debugs
 flow-shutdown     enable flow shutdown debugs
 generic           enable generic debugs
 ocsp              enable ocsp debugs
 oom-manager       enable oom-manager debugs
 openssl-internal  enable opnessl internal debugs
 peering-svc       enable peering service debugs
 session-cache     enable session cache debugs
 shell             enable SSL shell debugs
 sm-alert          enable session manager alert debugs
 sm-generic        enable session manager generic debugs
 sm-io             enable session manager i/o debugs
 sm-pipethrough    enable sm pipethrough debugs
 synchronization   enable synchronization debugs
```

```
 verify            enable certificate verification debugs
 waas-to-waas      enable waas-to-waas datapath debugs
```

You can enable debug logging for SSL connections and then display the end of the debug error log as follows:

```
WAE674# debug accelerator ssl all
WAE674# debug connection all
Enabling debug messages for all connections.
Are you sure you want to do this? (y/n) [n]y
WAE674# type-tail errorlog/sslao-errorlog.current follow
```

Troubleshooting Certificate Expiry Alarms on NME and SRE Modules

The SSL AO generates alarms when the self-signed machine certificate has expired (or is within 30 days of expiration) and a custom global machine certificate is not configured on the WAAS device. The WAAS software generates factory self-signed certificates with an expiration date of 5 years from the first startup of the WAAS device.

The clock in all WAAS NME and SRE modules is set to January 1, 2006 during first startup, even though the NME or SRE module is more recent. This causes the self-signed certificate to expire on January 1, 2011 and the device generates certificate expiration alarms.

If you are not using the default factory certificate as the global certificate, and instead are using a custom certificate for the SSL AO, you will not experience this unexpected expiration and you can update the custom certificate whenever it expires. Also, if you have updated the NME or SME module with a new software image and have synchronized the clock to a more recent date, you may not experience this issue.

The symptom of certificate expiration is one of the following alarms (shown here in the output of the **show alarms** command):

```
Major Alarms:
-------------
       Alarm ID                 Module/Submodule                Instance
  ---------------             --------------------           ---------------
  1 cert_near_expiration      sslao/SGS/gsetting           cert_near_expiration
```

or

```
       Alarm ID                 Module/Submodule                Instance
  ---------------             --------------------           ---------------
  1 cert_expired              sslao/SGS/gsetting           cert_expired
```

The Central Manager GUI reports the following alarm: "Certificate__waas-self__.p12 is near expiration it is configured as machine cert in global settings"

You can use one of the following solutions to resolve this problem:

- Configure a different certificate for global settings:

```
SRE# crypto generate self-signed-cert waas-self.p12 rsa modulus 1024
SRE# config
SRE(config)# crypto ssl services global-settings machine-cert-key waas-self.p12
```

- Update the self-signed factory certificate with a later expiration date. This solution requires a script that you can obtain by contacting Cisco TAC.

NOTE: This issue is fixed by the resolution of caveat CSCte05426, released in WAAS software versions 4.1.7b, 4.2.3c, and 4.3.3. The certification expiration date is changed to 2037.