This article describes how to troubleshoot the MAPI 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

- 1 MAPI Accelerator
- 2 Encrypted MAPI Acceleration
  - 2.1 Summary
  - 2.2 Feature Information
  - 2.3 Troubleshooting Methodology
    - 2.3.1 Step 1 - Verify Encryption Service Identity configuration and key retrieval success
    - 2.3.2 Step 2 - In 5.0.3 a new diagnostic command was introduced to check some of the required settings.
    - 2.3.3 Step 3- Manually verify the WAE settings that are not checked by the diagnostic command above.
  - 2.4 Data Analysis
  - 2.5 Common Problems
    - 2.5.1 Problem 1: The Encryption service identity configured on the Core WAE does not have the correct permissions in AD.
    - 2.5.2 Resolution 1: Consult the configuration guide and verify the object in AD has the correct permissions. "Replicating Directory Changes" and "Replicating Directory Changes All" must both be set to allow.
    - 2.5.3 Problem 2: There is a time skew between the Core WAE and the KDC it attempts to retrieve the key from
    - 2.5.4 Resolution 2: Use ntpdate on all WAEs (especially the Core) to sync the clock with the KDC. Then point to the enterprise NTP server (prefereably the same as the KDC).

- ◊ 2.5.5 Problem 3: The domain you defined for your Encryption service does not match the domain your Exchange server is in.
- ◊ 2.5.6 Resolution 3: If your Core WAE services multiple Exchange servers in different domains you must configure an Encryption service Identity for each domain the Exchange servers reside in.
- ◊ 2.5.7 Problem 4: If WANSecure fails your connections can drop to TG
- ◊ 2.5.8 Resolution 4: Remove peer cert verify configuration from both WAEs and restart the encrytpion service on the Core WAE(s).
- ◊ 2.5.9 Problem 5: If NTLM is used by the Outlook client the connection will be pushed down to Generic AO.
- ◊ 2.5.10 Resolution 5: The customer must enable / require Kerberos authentication in their Exchange environment. NTLM is NOT supported (as of 5.1)

• 3 MAPI AO Logging

MAPI Accelerator

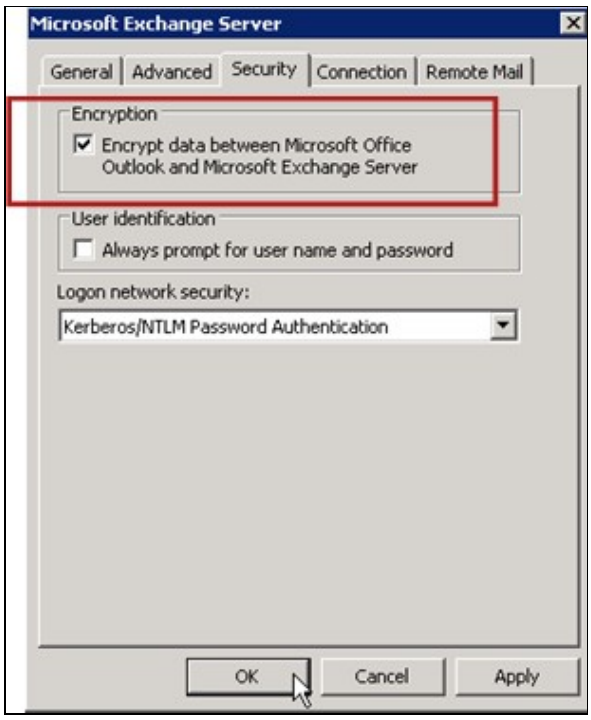
The MAPI accelerator optimizes Microsoft Outlook Exchange e-mail traffic. Exchange uses the EMSMDB protocol, which is layered on MS-RPC, which in turn uses either TCP or HTTP (unsupported) as the low level transport.

The MAPI AO supports Microsoft Outlook 2000 through 2007 clients for both cached and noncached mode traffic. Secure connections that use message authentication (signing) or encryption are not accelerated by the MAPI AO. Such connections and connections from older clients are handed off to the generic AO for TFO optimizations. Additionally, Outlook Web Access (OWA) and Exchange-Exchange connections are not supported.

Note: Microsoft Outlook 2007 has encryption enabled by default. You must disable encryption to benefit from the MAPI application accelerator. In Outlook, choose **Tools > E-mail Accounts**, choose **View or Change Existing E-mail Accounts**, and then click **Next**. Choose the Exchange account, and then click **Change**. Click **More Settings**, and then click the **Security** tab. Uncheck the **Encrypt data between Microsoft Office Outlook and Microsoft Exchange Server** check box, as shown in Figure 1.

Alternatively, you can disable encryption for all users of an Exchange Server by using a Group Policy.

Figure 1. Disabling Encryption in Outlook 2007



In the following cases, the MAPI AO does not handle a connection:

- Encrypted connection (handed off to the generic AO)
- Unsupported client (handed off to the generic AO)
- Unrecoverable parsing error. All TCP connections between the client and server service are disconnected. When the client reconnects, all connections are handed off to the generic AO.
- Client attempts to establish a new association group on the connection when the WAE is overloaded.
- Client establishes a connection when the WAE is overloaded and MAPI reserved connection resources are not available.

The Outlook client and server interact in a session over a group of TCP connections called an association group. Within an association group, object accesses can span across any connection and connections are dynamically created and released as needed. A client can have more than one association group open at the same time to different servers or the same server. (Public folders are deployed on different servers from the mail store.)

It is essential that all MAPI connections within an association group go through the same pair of WAEs in the branch and data center. If some connections within an association group do not go through the MAPI AO on these WAEs, the MAPI AO would not see the transactions performed on those connections and the connections are said to "escape" the association group. For this reason, the MAPI AO should not be deployed on serially clustered inline WAEs that form a high availability group.

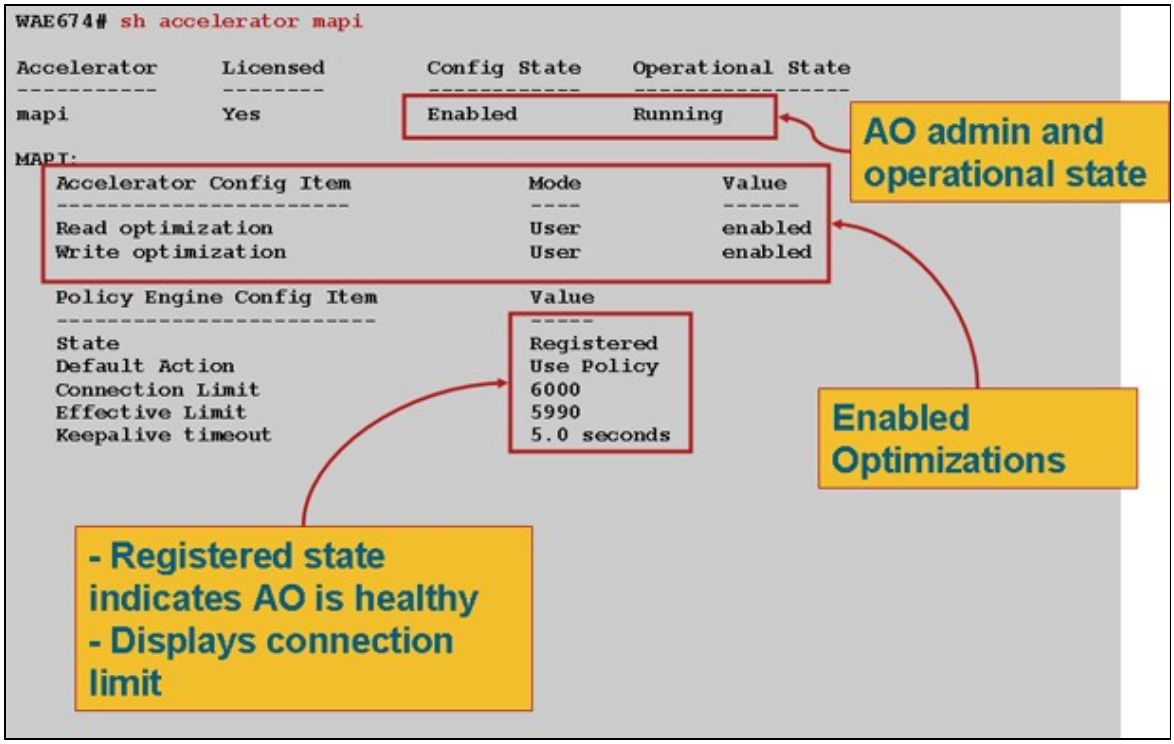
The symptoms of MAPI connections that escape their WAE association group are Outlook error symptoms such as duplicate messages or Outlook stops responding.

During a TFO overload condition, new connections for an existing association group would be passed through and escape the MAPI AO, so the MAPI AO reserves a number of connection resources in advance to minimize the impact of an overload condition. For more details about reserved MAPI connections and their impact on device overload, see the section "MAPI Application Accelerator Reserved Connections Impact on Overload" in the Troubleshooting Overload Conditions article.

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 MAPI accelerator operation and the EPM application accelerator must be enabled.

Next, verify the status specific to the MAPI AO by using the **show accelerator mapi** command, as shown in Figure 2. You want to see that the MAPI 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.

Figure 2. Verifying the MAPI Accelerator Status



Use the **show statistics accelerator epm** command to verify that the EPM AO is functional. Check that the Total Handled Connections, Total Requests Successfully Parsed, and Total Responses Successfully Parsed counters increase when a client is started.

Use the **show running-config** command to verify that the MAPI and EPM traffic policies are properly configured. You want to see **accelerate mapi** for the Email-and-Messaging application action and you want to see the MS-EndPortMapper classifier and traffic policy defined, as follows:

```
WAE674# sh run | include mapi
map adaptor EPM mapi
 name Email-and-Messaging All action optimize full accelerate mapi

WAE674# sh run | begin MS-EndPointMapper
...skipping
classifier MS-EndPointMapper
  match dst port eq 135
exit

WAE674# sh run | include MS-EndPointMapper
  classifier MS-EndPortMapper
    name Other classifier MS-EndPortMapper action optimize DRE no compression none accelerate MS-p
```

Use the **show policy-engine application dynamic** command to verify that dynamic match rules exist, as follows:

- Look for a rule with User ID: EPM and Map Name: uuida4f1db00-ca47-1067-b31f-00dd010662da.
- The Flows field indicates the total number of active connections to the Exchange service.
- For each MAPI client you should see a separate entry with the User ID: MAPI.

Use the **show statistics connection optimized mapi** command to check that the WAAS device is establishing optimized MAPI connections. Verify that "M" appears in the Accel column for MAPI connections, which indicates that the MAPI AO was used, as follows:

```
WAE674# show stat conn opt mapi

Current Active Optimized Flows:                       2
 Current Active Optimized TCP Plus Flows:          1
 Current Active Optimized TCP Only Flows:          1
 Current Active Optimized TCP Preposition Flows:   0
Current Active Auto-Discovery Flows:                  0
Current Reserved Flows:                               12              <---------- Added in 4.1.5
Current Active Pass-Through Flows:                    0
Historical Flows:                                     161


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

ConnID  Source IP:Port         Dest IP:Port        PeerID           Accel RR
342     10.56.94.101:4506    10.10.100.100:1456  0:1a:64:d3:2f:b8   TMDL  61.0%   <-----Look for "
```

Note: In version 4.1.5, the Current Reserved Flows counter was added in the output. This counter refers to the number of reserved MAPI connection resources on the WAE that are currently unused but set aside for future MAPI connections. For more details about reserved MAPI connections and their impact on device overload, see the section "MAPI Application Accelerator Reserved Connections Impact on Overload" in the Troubleshooting Overload Conditions article.

If you observe connections with "TGDL" in the Accel column, these connections were pushed down to the generic AO and optimized with transport optimizations only. If these are connections that you expected to be handled by the MAPI AO, it may be because they are encrypted MAPI connections. To check on the number of encrypted MAPI connections that have been requested, use the **show statistics accelerator mapi** command as follows:

```
wae# sh stat accel mapi

MAPI:
Global Statistics
-----------------
Time Accelerator was started:                                    Thu Nov  5 19:45:19 2009
Time Statistics were Last Reset/Cleared:                         Thu Nov  5 19:45:19 2009
Total Handled Connections:                                       8615
Total Optimized Connections:                                     8614
Total Connections Handed-off with Compression Policies Unchanged:  0
Total Dropped Connections:                                       1
Current Active Connections:                                      20
Current Pending Connections:                                     0
Maximum Active Connections:                                      512
Number of Synch Get Buffer Requests:                             1052
Minimum Synch Get Buffer Size (bytes):                           31680
Maximum Synch Get Buffer Size (bytes):                           31680
Average Synch Get Buffer Size (bytes):                           31680
```

```
Number of Read Stream Requests:                                  3844
Minimum Read Stream Buffer Size (bytes):                         19
Maximum Read Stream Buffer Size (bytes):                         31744
Average Read Stream Buffer Size (bytes):                         14556
Minimum Accumulated Read Ahead Data Size (bytes):                0
Maximum Accumulated Read Ahead Data Size (bytes):                1172480
Average Accumulated Read Ahead Data Size (bytes):                594385
Local Response Count:                                            20827
Average Local Response Time (usec):                              250895
Remote Response Count:                                           70486
Average Remote Response Time (usec):                             277036
Current 2000 Accelerated Sessions:                               0
Current 2003 Accelerated Sessions:                               1
Current 2007 Accelerated Sessions:                               0
Secured Connections:                                             1               <-----Encrypted
Lower than 2000 Sessions:                                        0
Higher than 2007 Sessions:                                       0
```

You can find the IP addresses of clients requesting encrypted MAPI connections in the syslog by searching for messages like the following:

```
2009 Jan 5 13:11:54 WAE512 mapi_ao: %WAAS-MAPIAO-3-132104: (929480) Encrypted connection. Client i
```

You can view the MAPI connection statistics by using the **show statistics connection optimized mapi detail** command as follows:

```
WAE674# show stat conn opt mapi detail
Connection Id:            1830
 Peer Id:                 00:14:5e:84:24:5f
 Connection Type:         EXTERNAL CLIENT
 Start Time:              Thu Jun 25 06:32:27 2009
 Source IP Address:       10.10.10.10
 Source Port Number:      3774
 Destination IP Address:  10.10.100.101
 Destination Port Number: 1146
 Application Name:        Email-and-Messaging                          <-----Should see Email-a
 Classifier Name:         **Map Default**
 Map Name:                uuida4f1db00-ca47-1067-b31f-00dd010662da      <-----Should see this UU
 Directed Mode:           FALSE
 Preposition Flow:        FALSE
 Policy Details:
        Configured:       TCP_OPTIMIZE + DRE + LZ
           Derived:       TCP_OPTIMIZE + DRE + LZ
              Peer:       TCP_OPTIMIZE + DRE + LZ
        Negotiated:       TCP_OPTIMIZE + DRE + LZ
           Applied:       TCP_OPTIMIZE + DRE + LZ
 Accelerator Details:
             Configured:  MAPI                                         <-----Should see MAPI co
                Derived:  MAPI
                Applied:  MAPI                                         <-----Should see MAPI ap
                   Hist:  None
                                        Original            Optimized
                                ------------------- -------------------
 Bytes Read:                                  4612                1973
 Bytes Written:                               4086                2096
. . .
```

Local and remote response counts and average response times are shown in this output:

```
 . . .
```

```
MAPI : 1830

 Time Statistics were Last Reset/Cleared:                          Thu Jun 25
06:32:27 2009
 Total Bytes Read:                                                46123985
 Total Bytes Written:                                             40864046
Number of Synch Get Buffer Requests:                              0
Minimum Synch Get Buffer Size (bytes):                            0
Maximum Synch Get Buffer Size (bytes):                            0
Average Synch Get Buffer Size (bytes):                            0
Number of Read Stream Requests:                                   0
Minimum Read Stream Buffer Size (bytes):                          0
Maximum Read Stream Buffer Size (bytes):                          0
Average Read Stream Buffer Size (bytes):                          0
Minimum Accumulated Read Ahead Data Size (bytes):                 0
Maximum Accumulated Read Ahead Data Size (bytes):                 0
Average Accumulated Read Ahead Data Size (bytes):                 0
Local Response Count:                                             0              <----------
Average Local Response Time (usec):                               0              <----------
Remote Response Count:                                            19             <----------
Average Remote Response Time (usec):                              89005          <----------
. . .
```

Encrypted MAPI Acceleration

Summary

As of WAAS 5.0.1 the MAPI accelerator can now accelerate encrypted MAPI traffic. This feature will be enabled by default in the 5.0.3 release. However, in order successfully accelerate encrypted MAPI traffic there are number of requirements in both the WAAS and Microsoft AD environment. This guide will help you verify and troubleshoot eMAPI functionality.

Feature Information

eMAPI will be enabled by default in 5.0.3 and will require the following to successfully accelerate encrypted traffic.

1) CMS secure store must be initialized and open on all Core WAEs

2) The WAEs must be able to resolve the FQDN of the Exchange server(s) and Kerberos KDC (Active Directory Controller)

3) The WAE's clocks must be in sync with the KDC

4) SSL acclerator, WAN Secure, and eMAPI must be enabled on all WAEs in the path from Outlook to Exchange

5) The WAEs in the path must have the correct policy-map configuration

6) The Core WAE(s) must have one or more Encrypted Services Domain Identies configured (User or Machine account)

7) If a machine account is used this WAE must be joined to the AD domain.

8) Then with either the Machine or User account use case, those objects in Active directory need to be given specific permissions. "Replicating Directory Changes" and "Replicating Directory Changes All" must both be set to allow.

The recommended way to do this is via a Universal Security group (e.g. assign the permissions to the group and then add the WAAS devices and/or usernames specified in the Encryption service to this group). See the attached guide for screenshots of AD configuration and WAAS CM GUI.

Troubleshooting Methodology

Step 1 - Verify Encryption Service Identity configuration and key retrieval success

While the diagnostics command (step 2 below) verifies the existence of an Encryption service it does not verify whether key retrieval will be successful. Hence we do not know by just running that diagnostic command if the proper permissions were given to the object in Active Directory (either Machine or User account).

Summary of what needs to be done to configure and verify Encryption service will succeed key retrieval

User account :

1. create AD user

2. create AD group and set "Replicating Directory Changes" and "Replicating Directory Changes All" to ALLOW

3. add the user to the group created

4. define user account domain identity in encryption services

5. run get key diagnostic cli

windows-domain diagnostics encryption-service get-key <exchange server FQDN> <domain name>

Note that you should use the actual/real exchange server name configured on the server and not a NLB/VIP type FQDN which might resolves to multiple exchange servers.

6. if key retrieval worked - done

Example of success:

pdi-7541-dc#windows-domain diagnostics encryption-service get-key pdidc-exchange1.pdidc.cisco.com pdidc.cisco.com

SPN pdidc-exchange1.pdidc.cisco.com, Domain name: pdidc.cisco.com

Key retrieval is in progress.

pdi-7541-dc#windows-domain diagnostics encryption-service get-key pdidc-exchange1.pdidc.cisco.com pdidc.cisco.com

SPN pdidc-exchange1.pdidc.cisco.com, Domain name: pdidc.cisco.com

Key for pdidc-exchange1.pdidc.cisco.com resides in memory key cache

Machine account

1. join core WAE device(s) to AD domain

2.create AD group and set "Replicating Directory Changes" and "Replicating Directory Changes All" to ALLOW

3. add machine account(s) to group created

4. configure encryption services to use machine account

5. Give sometime to get the Group Policy to be applied to the joined machine or force the application of group policy from the AD. gpupdate /force.

6. run get key diagnostic cli

windows-domain diagnostics encryption-service get-key <exchange server FQDN> <domain name>

Note that you should use the actual/real exchange server name configured on the server and not a NLB/VIP type FQDN which might resolves to multiple exchange servers.

7. if key retrieval worked - done

For more details and screen shots on Encrytpion service and AD config see the attached guide.

Step 2 - In 5.0.3 a new diagnostic command was introduced to check some of the required settings.

?accelerator mapi verify encryption settings

1.CLI does various validity checks. It output is summary of ability to accelerate encrypted MAPI traffic as edge or core.

2.Checks the various components? status/config attributes for Encryption Service to work properly.

3.When a configuration issue is found it will output what is missing and the CLI or actions to fix it.

4.It give the summary out as Edge device and Core device. Device which can be both edge and core should have EMAPI operational for both edge and core.

Below is a sample output from an incorrectly configured WAE:

```
Core#accelerator mapi verify encryption-settings
[EDGE:]
 Verifying Mapi Accelerator State
 --------------------------------
        Status: FAILED
        Accelerator     Config State    Operational State
        -----------     ------------    -----------------
        mapi            Disabled        Shutdown
        >>Mapi Accelerator should be Enabled
```

```
        >>Mapi Accelerator should be in Running state

 Verifying SSL Accelerator State
 -------------------------------
        Status: FAILED
        >>Accelerator   Config State    Operational State
        -----------     ------------    -----------------
        ssl             Disabled        Shutdown
        >>SSL Accelerator should be Enabled
        >>SSL Accelerator should be in Running state

 Verifying Wan-secure State
 --------------------------
        Status: FAILED
        >>Accelerator   Config State    Operational State
        -----------     ------------    -----------------
        wan-secure      Disabled        Shutdown
        >>Wan-secure should be Enabled
        >>Wan-secure should be in Running state

Verifying Mapi Wan-secure mode Setting
 ---------------------------------
        Status: FAILED
        Accelerator Config Item                 Mode            Value
        -----------------------                 ----            ------
        WanSecure Mode                          User            Not Applicable
        >>Mapi wan-secure setting should be auto/always
 Verifying NTP State
 -------------------
       Status: FAILED
       >>NTP status should be enabled and configured

Summary [EDGE]:
 ===============
      Device has to be properly configured for one or more components

[CORE:]
Verifying encryption-service State
 ----------------------------------
        Status: FAILED
        Service                 Config State    Operational State
        -----------             ------------    -----------------
        Encryption-service      Disabled        Shutdown
        >>Encryption Service should be Enabled
       >>Encryption Service status should be in 'Running' state

Verifying Encryption-service Identity Settings
 ----------------------------------------------
        Status: FAILED
        >>No active Encryption-service Identity is configured.
        >>Please configure an active Windows Domain Encryption Service Identity.

 Summary [CORE]: Applicable only on CORE WAEs
 ============================================
        Device has to be properly configured for one or more components
```

Below is the output from a Core WAE that is configured correctly:

```
Core#acc mapi verify encryption-settings [EDGE:]
```

Step 2 - In 5.0.3 a new diagnostic command was introduced to check some of the required settings

```
Verifying Mapi Accelerator State
--------------------------------
       Status: OK
Verifying SSL Accelerator State
-------------------------------
       Status: OK
Verifying Wan-secure State
--------------------------
       Status: OK
Verifying Mapi encryption Settings
----------------------------------
       Status: OK
Verifying Mapi Wan-secure mode Setting
----------------------------------
       Status: OK
Verifying NTP State
-------------------
       Status: OK
Summary [EDGE]:
===============
       Device has proper configuration to accelerate encrypted traffic

[CORE:]

Verifying encryption-service State
----------------------------------
       Status: OK
Verifying Encryption-service Identity Settings
----------------------------------------------
       Status: OK
Summary [CORE]: Applicable only on CORE WAEs
============================================
       Device has proper configuration to accelerate encrypted traffic
```

Step 3- Manually verify the WAE settings that are not checked by the diagnostic command above.

1) The above command while it chekcs for the existence of NTP configured, it does not actually verify the times are in sync between the WAE and KDC. It is very important the times are in sync between the Core and KDC for key retrieval to be successful.

If the manual check reveals they are out of sync a simple way to force the clock of the WAE to be in sync would be the ntpdate command (**ntpdate <KDC ip>**). Then point the WAEs to the enterprise NTP server.

2) Verify **dnslookup** succeeds on all WAEs for the Exchange servers' FQDN and the KDCs' FQDN

3) Verify the class-map and policy-map is configured correctly on all WAEs in the path.

pdi-7541-dc#sh class-map type waas MAPI

Class-map type waas match-any MAPI

Match tcp destination epm mapi (0 flow-matches)

pdi-7541-dc#show policy-map type waas Policy-map type waas

WAAS-GLOBAL (6084690 total)

Class MAPI (0 flow-matches)
optimize full accelerate mapi application Email-and-Messaging

4) Verify the CMS secure store is open and initialized on all WAEs "show cms secure store"

Data Analysis

Besides analyzing the output of the diagnostic command and the manual show commands you may need to review the sysreport.

Specifically you'll want to review the mapiao-errorlog, sr-errorlog (core WAE only), and wsao-errorlog files.

There will be hints in each log depending on the scenario which will lead you to the reason connections drop to Generic AO.

As a reference here is sample output showing various working components

This output is from sr-errorlog and shows validation of Machine Account Encryption Service Identity

Note: This only confirms the Core WAE joined the domain and the machine account exists.

```
07/03/2012 19:12:07.278(Local)(6249 1.5) NTCE (278902) Adding Identity MacchineAcctWAAS to map act
07/03/2012 19:12:07.279(Local)(6249 1.5) NTCE (279018) Adding identity(MacchineAcctWAAS) to Map [S
07/03/2012 19:12:07.279(Local)(6249 1.5) NTCE (279282) Activate Id: MacchineAcctWAAS [SRMain.cpp:2
07/03/2012 19:12:07.279(Local)(6249 1.5) NTCE (279306) Identity MacchineAcctWAAS found in the Map
07/03/2012 19:12:07.279(Local)(6249 1.5) NTCE (279321) Authentication for ID: MacchineAcctWAAS [SR
07/03/2012 19:12:07.330(Local)(6249 1.5) NTCE (330581) Authentication success, tkt validity startt
07/03/2012 19:12:07.330(Local)(6249 1.5) NTCE (330599)
ID_TAG :MacchineAcctWAAS
Name : pdi-7541-dc
Domain : PDIDC.CISCO.COM
Realm : PDIDC.CISCO.COM
CLI_GUID :
SITE_GUID :
CONF_GUID :
Status:ENABLED
Black_Listed:NO
AUTH_STATUS: SUCCESS
ACCT_TYPE:Machine [SRIdentityObject.cpp:85]
07/03/2012 19:12:07.331(Local)(6249 1.5) NTCE (331685) DN Info found for domain PDIDC.CISCO.COM [S
07/03/2012 19:12:07.347(Local)(6249 1.5) NTCE (347680) Import cred successfull for pn: pdi-7541-dc
```

This output is from the Core sr-errorlog again and shows successful key retrieval from KDC.

```
10/23/2012 15:46:55.673(Local)(3780 1.2) NTCE (673766) Key Not Found in cache, initiating retrieva
10/23/2012 15:46:55.673(Local)(3780 1.2) NTCE (673811) Queued InitiateKeyRetrieval task [SRServer.
Key retrieval is in Progress [SRServer.cpp:322]
```

```
10/23/2012 15:46:55.673(Local)(3780 0.0) NTCE (673818) Initiating key retrieval [SRServer.cpp:271]
10/23/2012 15:46:55.673(Local)(3780 1.2) NTCE (673827) initiating key retrieval in progress [SRDat
10/23/2012 15:46:55.673(Local)(3780 1.2) NTCE (673834) Sending ack for result 2, item name /cfg/gl
[SRDataServer.cpp:444]
10/23/2012 15:46:55.673(Local)(3780 0.0) NTCE (673922) Match found for DN: pdidc.cisco.com is ID:M
10/23/2012 15:46:55.673(Local)(3780 0.0) NTCE (673937) Identity MacchineAcctWAAS found in the Map
10/23/2012 15:46:55.673(Local)(3780 0.0) NTCE (673950) DN Info found for domain pdidc.cisco.com [S
10/23/2012 15:46:55.674(Local)(3780 0.0) NTCE (674011) DRS_SPN: E3514235-4B06-11D1-AB04-00C04FC2DC
PDI-7541-DC@PDIDC.CISCO.COM [GssCli.cpp:51]
10/23/2012 15:46:55.674(Local)(3780 0.0) NTCE (674020) CREATED srkr obj(0x50aa00) for spn (exchang
10/23/2012 15:46:55.674(Local)(3780 1.3) NTCE (674421) Import cred successfull for pn: PDI-7541-DC
10/23/2012 15:46:55.676(Local)(3780 1.3) NTCE (676280) session(0x50aa00) Complete TGT stage of GSS
10/23/2012 15:46:55.676(Local)(3780 0.1) NTCE (676415) SRKR: Success in posting connect to service
10/23/2012 15:46:55.676(Local)(3780 0.0) NTCE (676607) Connected to server. [IoOperation.cpp:389]
10/23/2012 15:46:55.677(Local)(3780 0.0) NTCE (677736) SRKR: Success in posting connect to service
10/23/2012 15:46:55.678(Local)(3780 0.1) NTCE (678001) Connected to server. [IoOperation.cpp:389]
10/23/2012 15:46:55.679(Local)(3780 0.1) NTCE (679500) Cleaning up credential cache for PDI-7541-D
10/23/2012 15:46:55.680(Local)(3780 0.1) NTCE (680011) Parsing DRSBIND Response [AppApiDrsBind.cpp
10/23/2012 15:46:55.680(Local)(3780 0.1) NTCE (680030) DRSBind Success, Status:00000000 [AppApiDrs
10/23/2012 15:46:55.685(Local)(3780 0.1) NTCE (685502) session(0x50aa00) Successful in Key Retriev
[SRKeyRetriever.cpp:269]
10/23/2012 15:46:55.685(Local)(3780 0.1) NTCE (685583) Send Key response to the Client for spn: ex
[SRKeyMgr.cpp:296]
10/23/2012 15:46:55.685(Local)(3780 0.1) NTCE (685594) Deleting spn: exchangeMDB/pdidc-exchange1.p
```

This output is from the mapiao-errorlog file on the Edge WAE for a successful eMAPI connection

```
'''10/23/2012 17:56:23.080(Local)(8311 0.1) NTCE (80175) (fl=2433) Edge TCP connection initiated (
Flavor: 0 [EdgeTcpConnectionDceRpcLayer.cpp:43]
10/23/2012 17:56:23.080(Local)(8311 0.1) NTCE (80199) Edge TCP connection initiated (-1409268656),
[EdgeTcpConnectionDceRpcLayer.cpp:48]
10/23/2012 17:56:23.108(Local)(8311 0.0) NTCE (108825) (fl=2433) Bind Request from client with AGI
AuthType: SPNEGO AuthCtxId: 0 WsPlumb:1
[EdgeTcpConnectionDceRpcLayer.cpp:1277]'''
10/23/2012 17:56:23.109(Local)(8311 0.0) NTCE (109935) CheckAndDoAoshReplumbing perform replumbing
10/23/2012 17:56:23.109(Local)(8311 0.0) NTCE (109949) (fl=2433) AOSH Replumbing was performed ret
10/23/2012 17:56:23.109(Local)(8311 0.0) NTCE (109956) CheckAndPlumb WanSecure(14) ret:= [1,0] WsP
10/23/2012 17:56:23.312(Local)(8311 0.1) NTCE (312687) (fl=2433) Connection multiplexing enabled b
10/23/2012 17:56:23.312(Local)(8311 0.1) NTCE (312700) (fl=2433) Header signing enabled by server
10/23/2012 17:56:23.312(Local)(8311 0.1) NTCE (312719) (fl=2433) OnNewConnection - Client IP 14.11
nAssociationGroup=0x11de4,conn_fd=26,
bWasConnectionFromReservedPool=0, bIsNewMapiSession=1 [ConnectionReservationManager.cpp:255]
'''10/23/2012 17:56:23.366(Local)(8311 0.1) NTCE (366789) (fl=2433) Received security context from
10/23/2012 17:56:23.367(Local)(8311 0.1) NTCE (367157) (fl=2433) Security Layer moved to ESTB stat
10/23/2012 17:56:23.368(Local)(8311 0.1) NTCE (368029) (fl=2433) Informational:: Send APC set to W
10/23/2012 17:56:23.368(Local)(8311 0.1) NTCE (368041) (fl=2433) Sec-Params [CtxId, AL, AT, ACT, D
[FlowIOBuffers.cpp:477]
10/23/2012 17:56:23.369(Local)(8311 0.0) NTCE (369128) (fl=2433) CEdgeTcpConnectionEmsMdbLayer::Co
Product Minor:0, Build Major:6117,
Build Minor:5001 Client ip 14.110.3.117 Client port 58352 Dest ip 14.110.3.99 Dest port 27744 [Edg
10/23/2012 17:56:23.868(Local)(8311 0.1) ERRO (868390) (fl=2433) ContextHandle.IsNull() [EdgeTcpCo
10/23/2012 17:56:23.890(Local)(8311 0.0) NTCE (890891) (fl=2433) CEdgeTcpConnectionEmsMdbLayer::Co
Product Minor:0, Build Major:6117,
Build Minor:5001 Client ip 14.110.3.117 Client port 58352 Dest ip 14.110.3.99 Dest port 27744 [Edg
```

Here is the corresponding Core WAE output from mapiao-errorlog for the same TCP conneciton

```
'''10/23/2012 17:56:54.092(Local)(6408 0.0) NTCE (92814) (fl=21) Core TCP connection initiated (11
lavor: 0 [CoreTcpConnectionDceRpcLayer.cpp:99]
10/23/2012 17:56:54.092(Local)(6408 0.0) NTCE (92832) Core TCP connection initiated (11892640), Co
[CoreTcpConnectionDceRpcLayer.cpp:104]'''
10/23/2012 17:56:54.175(Local)(6408 0.0) NTCE (175035) SrlibCache Cache eviction starting: static
id*, aosh_work*) [SrlibCache.cpp:453]
10/23/2012 17:56:54.175(Local)(6408 0.0) NTCE (175068) last_cleanup_time (1344411860), evict_in_pr
cpp:464]
10/23/2012 17:56:54.175(Local)(6408 0.0) NTCE (175121) SendNextCmd isDuringSend 0, WriteQueue sz 1
10/23/2012 17:56:54.175(Local)(6408 0.0) NTCE (175132) SendNextCmd: Sending request: exchangeMDB/P
[bClose 0] [SrlibClientTransport.cpp:168]
10/23/2012 17:56:54.185(Local)(6408 0.1) NTCE (185576) OnReadComplete len 4 status 0 isDuringRead
cpp:127]
10/23/2012 17:56:54.185(Local)(6408 0.1) NTCE (185587) Parse header, msg body len 152 [SrlibTransp
10/23/2012 17:56:54.185(Local)(6408 0.1) NTCE (185592) ReadNextMsg isDuringRead 0, isDuringHeaderR
10/23/2012 17:56:54.185(Local)(6408 0.1) NTCE (185623) OnReadComplete len 148 status 0 isDuringRea
t.cpp:127]
'''10/23/2012 17:56:54.185(Local)(6408 0.1) NTCE (185688) Insert new KrbKey: exchangeMDB/PDIDC-EXC
rlibCache.cpp:735]
'''10/23/2012 17:56:54.185(Local)(6408 0.1) NTCE (185747) ReadNextMsg isDuringRead 0, isDuringHead
'''10/23/2012 17:56:54.261(Local)(6408 0.1) NTCE (261575) (fl=21) Successfully created memory keyt
.com0nxrPblND [GssServer.cpp:468]
10/23/2012 17:56:54.261(Local)(6408 0.1) NTCE (261613) (fl=21) Successfully added entry in memory
10/23/2012 17:56:54.261(Local)(6408 0.1) NTCE (261858) (fl=21) Successfully acquired credentials.
```

Common Problems

Below are some common reasons which results in eMAPI connection Hand-off to Generic AO (TG).

Problem 1: The Encryption service identity configured on the Core WAE does not have the correct permissions in AD.

Output from sr-errolog on Core WAE

```
09/25/2012 18:47:54.147(Local)(9063 0.1) ERRO (147570) session(0x517fa0) Failed to Retrieve Key fr
'''09/25/2012 18:47:54.147(Local)(9063 0.1) ERRO (147592) Key retrieval failed with Status 16 [SRK
''''''09/25/2012 18:47:54.147(Local)(9063 0.1) ERRO (147623) Identity "WAASMacAct" has been blackl
''''''09/25/2012 18:47:54.147(Local)(9063 0.1) ERRO (147631) Key retrieval failed due to permissio
'''09/25/2012 18:47:54.147(Local)(9063 0.1) ERRO (147636) Identity: WAASMacAct will be black liste
09/25/2012 18:47:54.147(Local)(9063 0.1) NTCE (147657) Calling KrbKeyResponse  key handler in srli
09/25/2012 18:47:54.147(Local)(9063 0.1) NTCE (147722) Queued send reponse buffer to client task [
09/25/2012 18:47:54.147(Local)(9063 0.1) NTCE (147730) KrbKeyResponse, sent to client session obje
09/25/2012 18:47:54.147(Local)(9063 0.0) NTCE (147733) SendNextCmd isDuringSend 0, WriteQueue size
09/25/2012 18:47:54.147(Local)(9063 0.1) NTCE (147740) Send Key response to the Client
```

Resolution 1: Consult the configuration guide and verify the object in AD has the correct permissions. "Replicating Directory Changes" and "Replicating Directory Changes All" must both be set to allow.

http://www.cisco.com/en/US/docs/app_ntwk_services/waas/waas/v511/configuration/guide/policy.html#wp1256547

Problem 2: There is a time skew between the Core WAE and the KDC it attempts to retrieve the key from

Output from sr-errolog on Core WAE

```
10/23/2012 01:31:33.507(Local)(1832 0.1) NTCE (507836) Initiating key retrieval [SRServer.cpp:271]
10/23/2012 01:31:33.507(Local)(1832 0.1) NTCE (507878) Match found for DN: pdidc.cisco.com is ID:M
10/23/2012 01:31:33.507(Local)(1832 0.1) NTCE (507888) Identity MacchineAcctWAAS found in the Map
10/23/2012 01:31:33.507(Local)(1832 0.1) NTCE (507901) DN Info found for domain pdidc.cisco.com [S
10/23/2012 01:31:33.507(Local)(1832 0.1) NTCE (507923) DRS_SPN: E3514235-4B06-11D1-AB04-00C04FC2DC
PDI-7541-DC@PDIDC.CISCO.COM [GssCli.cpp:51]
10/23/2012 01:31:33.507(Local)(1832 0.1) NTCE (507933) CREATED srkr obj(0x2aaaac0008c0) for spn (e
10/23/2012 01:31:33.508(Local)(1832 1.6) NTCE (508252) Import cred successfull for pn: PDI-7541-DC
10/23/2012 01:31:33.511(Local)(1832 1.6) ERRO (511151) CreateSecurityContext: gss_init_sec_context
'''10/23/2012 01:31:33.511(Local)(1832 1.6) ERRO (511170) GSS_MAJOR ERROR:851968 msg_cnt:0, Miscel
10/23/2012 01:31:33.511(Local)(1832 1.6) ERRO (511177) GSS_MINOR ERROR:2529624064 msg_cnt:0, Clock
10/23/2012 01:31:33.511(Local)(1832 1.6) ERRO (511182) gsskrb5_get_subkey failed: 851968,22, [GssC
10/23/2012 01:31:33.511(Local)(1832 1.6) ERRO (511188) session(0x2aaaac0008c0) Error: Invalid secu
[SRKeyRetriever.cpp:386]
10/23/2012 01:31:33.511(Local)(1832 1.6) ERRO (511193) session(0x2aaaac0008c0) Failed to Retrieve
[SRKeyRetriever.cpp:267]'''
10/23/2012 01:31:33.511(Local)(1832 0.0) ERRO (511213) Key retrieval failed with Status 1 [SRKeyMg
```

Resolution 2: Use ntpdate on all WAEs (especially the Core) to sync the clock with the KDC. Then point to the enterprise NTP server (prefereably the same as the KDC).

Problem 3: The domain you defined for your Encryption service does not match the domain your Exchange server is in.

Output from sr-errolog on Core WAE

```
10/23/2012 18:41:21.918(Local)(3780 1.5) NTCE (918788) Key retrieval is in Progress [SRServer.cpp:
10/23/2012 18:41:21.918(Local)(3780 1.5) NTCE (918793) initiating key retrieval in progress [SRDat
10/23/2012 18:41:21.918(Local)(3780 0.0) NTCE (918790) Initiating key retrieval [SRServer.cpp:271]
10/23/2012 18:41:21.918(Local)(3780 1.5) NTCE (918798) Sending ack for result 2, item name /cfg/gl
10/23/2012 18:41:21.918(Local)(3780 0.0) ERRO (918813) Failed to find Identity match for domain ci
10/23/2012 18:41:21.918(Local)(3780 0.0) NTCE (918821) Failed to find identity match for domain [S
10/23/2012 18:41:21.918(Local)(3780 0.0) NTCE (918832) Send Key response to the Client for spn: ex
```

Resolution 3: If your Core WAE services multiple Exchange servers in different domains you must configure an Encryption service Identity for each domain the Exchange servers reside in.

Note, there is NO support for sub-domain include at this time. So if you have

myexchange.sub-domain.domain.com , the Encryption service Identity must be in sub-domain.domain.com; it CAN NOT be in the parent domain.

Problem 4: If WANSecure fails your connections can drop to TG

eMAPI connections can be handed over to Generic AO because WAN secure plumb failed. WAN Secure plumb failed because cert verify failed. Peer cert verify will failed because the default self-signed peer cert is being used or the cert has legitimately failed OCSP check.

Core WAE settings

```
crypto pki global-settings

  ocsp url http://pdidc.cisco.com/ocsp
revocation-check ocsp-cert-url
exit

!

crypto ssl services host-service peering

  peer-cert-verify
exit

!

WAN Secure:

  Accelerator Config Item             Mode           Value
------------------------             ----           ------
SSL AO                               User           enabled
Secure store                         User           enabled
Peer SSL version                     User           default
Peer cipher list                     User           default
Peer cert                            User           default
Peer cert verify                     User           enabled
```

This will result in the following mapiao-errorlog and wsao-errorlog entries:

The hint here is the first highlighted line "disconnected more than four consecutive times"

Mapiao-errorlog on client side WAE:

```
'''10/08/2012 20:02:15.025(Local)(24333 0.0) NTCE (25621) (fl=267542) Client 10.16.1.201 disconnec
[EdgeTcpConnectionDceRpcLayer.cpp:1443]
'''10/08/2012 20:02:15.025(Local)(24333 0.0) NTCE (25634) (fl=267542) CEdgeIOBuffers:: StartHandOv
[EdgeIOBuffers.cpp:826]
10/08/2012 20:02:15.025(Local)(24333 0.0) NTCE (25644) (fl=267542) CEdgeIOBuffers::CheckSendHandOv
fragment of call id 0, current call id is 2 [EdgeIOBuffers.cpp:324]
10/08/2012 20:02:15.048(Local)(24333 0.1) NTCE (48753) (fl=267542) Connection multiplexing enabled
10/08/2012 20:02:15.048(Local)(24333 0.1) NTCE (48771) (fl=267542) Header signing enabled by serve
10/08/2012 20:02:15.048(Local)(24333 0.1) NTCE (48779) (fl=267542) CEdgeIOBuffers:: StartHandOverP
```

Resolution 3: If your Core WAE services multiple Exchange servers in different domains you must configure an

Wsao-errorlog on client side WAE:

```
'''10/08/2012 20:04:34.430(Local)(5939 4.0) ERRO (430001) certificate verification failed 'self si
'''10/08/2012 20:04:34.430(Local)(5939 4.0) ERRO (430047) ssl_read failed: 'SSL_ERROR_SSL' [open_s
10/08/2012 20:04:34.430(Local)(5939 4.0) ERRO (430055) openssl errors: error:14090086: SSL routine
[open_ssl.cpp:1220]
```

Resolution 4: Remove peer cert verify configuration from both WAEs and restart the encrytpion service on the Core WAE(s).

```
pdi-7541-dc(config)#crypto ssl services host-service peering

pdi-7541-dc(config-ssl-peering)#no peer-cert-verify

pdi-7541-dc(config)#no windows-domain encryption-service enable

pdi-7541-dc(config)#windows-domain encryption-service enable
```

Problem 5: If NTLM is used by the Outlook client the connection will be pushed down to Generic AO.

You will see the following in the mapiao-errorlog on the client side WAE:

```
'''waas-edge#find-patter match ntlm mapiao-errorlog.current
'''
09/21/2012 20:30:32.154(Local)(8930 0.1) NTCE (154827) (fl=83271) Bind Request from client with AG
PRIVACY '''AuthType:NTLM '''AuthCtxId: 153817840 WsPlumb: 2 [EdgeTcpConnectionDceRpcLayer.cpp:1277
09/21/2012 20:30:32.154(Local)(8930 0.1) NTCE (154861) (fl=83271) '''Unsupported''' '''Auth Type :
(8930 0.0) NTCE (157628) (fl=83283) Bind Request from client with AGID 0x0, callId 2, to dest-ip 1
WsPlumb: 2 [EdgeTcpConnectionDceRpcLayer.cpp:1277]
```

Resolution 5: The customer must enable / require Kerberos authentication in their Exchange environment. NTLM is NOT supported (as of 5.1)

Be aware there is a Microsoft tech brief that calls out a fall back to NTLM when a CAS is used.

The scenario where Kerberos does not function is specific to Exchange 2010, and is in the following scenario:

Multiple Exchange Client Access Servers (CAS) in an organization/domain.
These CAS servers are clustered together using any method - using Microsoft's built-in Client-array function, or a 3rd party load balancer.

In the scenario above, Kerberos does not work - and clients will fall-back to NTLM by defult. I believe this is due to the fact that clients have to AUTH to the CAS server vs. the Mailbox server, as they did in previous Exchange releases.

In Exchange 2010 RTM, there is no fix for this! Kerberos in the above scenario will never function pre-Exchange 2010-SP1.

In SP1, Kerberos can be enabled in these environments, but it's a manual process. See the article here:
http://technet.microsoft.com/en-us/library/ff808313.aspx

MAPI AO Logging

- The following log files are available for troubleshooting MAPI AO issues:
- Transaction log files: /local1/logs/tfo/working.log (and /local1/logs/tfo/tfo_log_*.txt)

Debug log files: /local1/errorlog/mapiao-errorlog.current (and mapiao-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 19:12:35 2009 :2289 :10.10.10.10 :3740 :10.10.100.101 :1146 :OT :END :EXTERNAL CLIENT :
Wed Jul 15 19:12:35 2009 :2289 :10.10.10.10 :3740 :10.10.100.101 :1146 :SODRE :END :730 :605 :556
Wed Jul 15 19:12:35 2009 :2290 :10.10.10.10 :3738 :10.10.100.101 :1146 :OT :END :EXTERNAL CLIENT :
Wed Jul 15 19:12:35 2009 :2290 :10.10.10.10 :3738 :10.10.100.101 :1146 :SODRE :END :4550 :15854 :6
Wed Jul 15 19:12:35 2009 :2284 :10.10.10.10 :3739 :10.10.100.101 :1146 :OT :END :EXTERNAL CLIENT :
```

To set up and enable debug logging of the MAPI 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 MAPI AO debugging are as follows:

```
WAE674# debug accelerator mapi ?
all enable all MAPI accelerator debugs
Common-flow enable MAPI Common flow debugs
DCERPC-layer enable MAPI DCERPC-layer flow debugs
EMSMDB-layer enable MAPI EMSMDB-layer flow debugs
IO enable MAPI IO debugs
ROP-layer enable MAPI ROP-layer debugs
ROP-parser enable MAPI ROP-parser debugs
RPC-parser enable MAPI RPC-parser debugs
shell enable MAPI shell debugs
Transport enable MAPI transport debugs
Utilities enable MAPI utilities debugs
```

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

```
WAE674# debug accelerator mapi Common-flow
WAE674# type-tail errorlog/mapiao-errorlog.current follow
```