Medical School: Diagnosing SSL/TLS and AT-TLS Problems in z/OS Communications Server

Speaker: Gwendolyn J. Dente (gdente@us.ibm.com)
IBM Advanced Technical Support (ATS)
Gaithersburg, Maryland (USA)

Friday, February 8, 2013: 9:30 AM-10:30 AM
(San Francisco Hilton, Golden Gate 3, Lobby Level)
System z Social Media Channels

- Top Facebook pages related to System z:
  - IBM System z
  - IBM Academic Initiative System z
  - IBM Master the Mainframe Contest
  - IBM Destination z
  - Millennial Mainframer
  - IBM Smarter Computing

- Top LinkedIn groups related to System z:
  - System z Advocates
  - SAP on System z
  - IBM Mainframe- Unofficial Group
  - IBM System z Events
  - Mainframe Experts Network
  - System z Linux
  - Enterprise Systems
  - Mainframe Security Gurus

- Twitter profiles related to System z:
  - IBM System z
  - IBM System z Events
  - IBM DB2 on System z
  - Millennial Mainframer
  - Destination z
  - IBM Smarter Computing

- YouTube accounts related to System z:
  - IBM System z
  - Destination z
  - IBM Smarter Computing

- Top System z blogs to check out:
  - Mainframe Insights
  - Smarter Computing
  - Millennial Mainframer
  - Mainframe & Hybrid Computing
  - The Mainframe Blog
  - Mainframe Watch Belgium
  - Mainframe Update
  - Enterprise Systems Media Blog
  - Dancing Dinosaur
  - DB2 for z/OS
  - IBM Destination z
  - DB2utor
Abstract

• For z/OS Communications Server it is rarely necessary to take an SSL trace for diagnosing problems with SSL/TLS or AT-TLS. A simple look in the SyslogD log or even in the messages on the MVS console can reveal what has gone wrong with the secured connection you are testing. Come to this session to see the easy way to diagnose such encrypted session problems.

• The examples are taken from an AT-TLS implementation with z/OS and Policy Agent. However, the basic SSL/TLS return codes and messages could appear in any implementation that uses zOS System SSL. The reference manuals at the back of this presentation show you how to find the correct manuals for discovering the meanings of these SSL Return Codes.
Agenda

- SSL/TLS/AT-TLS Protocol Flow
- Examination of Output from an SSL Trace in z/OS
- Description of Scenario that we are testing
- Error #1: Server cannot find its key ring
- Error #2: Server not authorized to read key ring
- Error #3: Server cannot find its certificate
- Error #4: Server cannot find its certificate or private key
- Error #5: Client configuration specifies a DNS lookup that does not exist
- Error #6: Server Certificate has expired (See Certificate Lab 12895)
- Summary of Common Problems with SSL/TLS and AT-TLS
- Resources for Diagnosing z/OS SSL/TLS and AT-TLS Errors
SSL/TLS/AT-TLS Negotiation Flows

Request Secure Session (and Session ID): Which SSL Version, Ciphers, Compression Algorithms, etc.

Client HELLO Packet

Server HELLO Packet

Response to Secure Session Request (Session ID): Selection of SSL Version, Ciphers, Compression Algorithms, etc.

Server x.509 CERTIFICATE

Optional: Request for Client x.509 CERTIFICATE

Client validates or rejects Server Certificate with Error

Optional: Client x.509 CERTIFICATE sent

Client sends Encrypted Symmetric Key

Record Layer: Encrypted Data Flow

This page shows the approximate flow of the SSL/TLS negotiation for a secure session; this is called the HANDSHAKE LAYER and is negotiated with the HELLO Exchange. It is followed by the RECORD LAYER, in which the production data is encrypted using the Symmetric Key.
The GSKSRVR SSL Trace provides complete and very valuable information on what is happening with an SSL or TLS flow. However, if one does not know the protocols, it can be difficult to read. Frequently it is possible to solve an SSL/TLS problem just by turning on AT-TLS tracing in the AT-TLS policy, which provides more understandable messages about what may have gone awry in an encrypted setup or flow. In addition, the MVS console and the RACF messages there also provide meaningful clues. If the “easier messages” from the trace and MVS logs do not yield the information you need to solve a problem, you can look at the messages on the partner side of the connection. And, yes, there are times when you must produce an SSL/TLS GSKSRVR trace in order to examine the contents of the certificates that have been exchanged during the setup of the secured session. Therefore, in this session, we do not discount the value of an SSL Trace. We just point out that without the necessary background, it can simply be a lot easier to solve problems by using other tools that we show you here.
SSL/TLS Trace Output in z/OS (2)

Deciding which Server Certificate to Send to Client

Examining Ciphers available for connection

Determining secure protocol version to use

Determining the Cipher to use

Deciding which Server Certificate to Send to Client

Examining Ciphers available for connection

Determining secure protocol version to use

Determining the Cipher to use
SSL/TLS Trace Output in z/OS (3)

Sending Server Certificate Msg. to Client

Sending Server HELLO with values negotiated

Decrypting message with Clear Key DES
SSL/TLS Trace Output in z/OS (4)

Sending Server Certificate to Client

Sending CA Certificate to Client
SSL/TLS Trace Output in z/OS (5)

**Sending Server HELLO_DONE to Client**

**Received Encrypted Symmetric Key from Client**

**Using Server Private Key (PKCS) to decrypt the Symmetric Key That the client had encrypted with Server Public Key**

**Record Layer: Encrypted Data Flow**
This page illustrates a “home-grown” exec that strips out TLS wrappers and line formatting, and then converts the certificate into binary so that the Windows formatter can read it...
Error #1: Server Cannot Find its Key Ring

FTP.DATA specifies Server Authentication Only

1. All Key Rings are shared and contain valid and trusted certificates that have not yet expired.
2. Testing between Source and Destination OSA Port addresses:
   TCPPIPT: 192.168.20.91-97

Both FTP Servers are sharing the same Key Ring across all MVS Images. The Client Key Ring is also shared across all clients and MVS images.
Error #1: View of Error Messages

We raised the trace level in the AT-TLS policy from 7 to 255, but perhaps we did not need to. Perhaps the messages on the MVS console are enough to solve this problem.

The server reset the connection – not the client. In addition, the server reported more errors than the client on the MVS console. We must examine the meaning of all these messages and the return codes.
Please reference all the materials listed in the appendix of this presentation. For now, these two documents suffice for our problem determination steps, as they contain the meanings of the SSL and AT-TLS return codes.
The error messages and the return codes all point toward a problem at the Server side of the connection. The problem is related to the Handshake. We know that integral to the handshake is the exchange of negotiation parameters and certificates. The RC of 202 points to a problem that indicates that the server may not be able to find its keyring. And, in fact, that is the problem; There is a typo in the AT-TLS policy where the owner of the Key Ring was spelled wrong. The correct spelling is "FTPD" and not "FTPP." We change the policy and we force System SSL to reinstall the keyring knowledge by raising what is called an INSTANCE Number for AT-TLS.
We discovered that the key ring for the server was incorrectly specified as belonging to a USERID of FTPP. We corrected this to identify the true owner of this Key Ring (FTPD). When our client establishes a secured FTP connection this time, he is successful. The messages that the client sees due to our setup of the Client FTP.DATA file, we even recognize the actual cipher and TLS protocol that we are using for the successful negotiation and establishment.

This is an excerpt from the policy where you see we have increased the policy instance number for the server so that any update or refresh of the PAGENT procedure will cause System SSL to refresh the image of the Key Ring that is in memory for the FTP Server to use:

```plaintext
TTLSEnvironmentAction eAct1~FTPTat192.168.20.9n
{
    HandshakeRole Server
    EnvironmentUserRoleInstance 1 <<<<<<<<<<<<<<<<<<<<<< was 0 before
    <<<<<<<<<<<<<<<<<<<<<<<<
    TTLSEnvironmentAction keyR1
}
```
We were able to solve the problem without looking at the AT-TLS trace output from Policy Agent and the policy. But we show you that this trace would have also provided valuable information to solve the issue. And... this trace is easier to set up and take, with fewer steps, than the SYSTEM SSL Trace. However, there are times when you must provide a System SSL trace, and so it is wise to learn how to set one of these up. (System SSL trace is not a subject of this presentation.)
Explaining the Solution -- Server Key Ring:
Who “Owns” It & How to Find It?

1) Key Ring for a Server or a Client:
Define a USERID and assign an OMVS (UNIX) Identity to it:

```
ADDUSER FTPD    DFLTGRP(OMVSGRP) OMVS(UID(0) HOME('/'))
NOPASSWORD
```

Create a Started Class definition for the Server and associate it with its OMVS Segment (i.e., its USERID or OWNER)

```
RDEFINE STARTED MYFTP:*.*    STDATA(USER(FTPD))
```

Associate an x.509 Server Certificate with its OMVS Segment OWNER

```
RACDCERT ID(FTPD) GENCERT ...
```

2) PERSONAL Certificate for a Client (Assumption: Client is a human user)
Define a USERID and assign an OMVS (UNIX) Identity to it:

```
ADDUSER USER71    DFLTGRP(OMVSGRP) OMVS(UID(707) HOME('/u/user71'))
```

Associate an x.509 Client Certificate with its OMVS Segment OWNER

```
RACDCERT ID(USER71) GENCERT ...
```
Explaining the Solution -- Client Key Ring: Who “Owns” It & How to Find It?

1) Key Ring for a Server or a Client:
Define a USERID and assign an OMVS (UNIX) Identity to it:

```
ADDUSER FTPD   DFLTGRP(OMVSGRP) OMVS(UID(0) HOME('/'))
NOPASSWORD
```

Create a Started Class definition for the Server and associate it with its OMVS Segment (i.e., its USERID or OWNER)

```
RDEFINE  STARTED  MYFTP*.*         STDATA(USER(FTPD))
```

Associate an x.509 Server Certificate with its OMVS Segment OWNER

```
RACDCERT ID(FTPD) GENCERT ...
```

2) PERSONAL Certificate for a Client (Assumption: Client is a human user)
Define a USERID and assign an OMVS (UNIX) Identity to it:

```
ADDUSER USER301   DFLTGRP(OMVSGRP) OMVS(UID(707) HOME('/u/user301'))
...
```

Associate an x.509 Client Certificate with its OMVS Segment OWNER

```
RACDCERT ID(USER301) GENCERT ...
```
ERROR #2: Not Authorized to Read Key Ring

ICH408I USER(USER301) GROUP(USER) NAME(USER301)
IRR.DIGTCERT.LISTRING CL(FACILITY)
INSUFFICIENT ACCESS AUTHORITY
ACCESS INTENT(READ) ACCESS ALLOWED(NONE)

ERROR #2: Solution

- For a Client who does not need to present a client certificate:

  - RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
  - PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(USER301) ACCESS(READ)
  - SETROPTS RACLIST(FACILITY) REFRESH

- For a Server or a Client who needs to present a client certificate:

  - RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
  - RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
  - PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(xxxx) ACCESS(…….)
  - PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(xxxx) ACCESS(……)
  - SETROPTS RACLIST(FACILITY) REFRESH

Here you see that the User could not open the key ring to be able to validate the received Server certificate against the client’s stored copy of the CA certificate that signed the Server Certificate. The solution was simply to authorize the users of the key rings to open and read the key rings. RACF since V1R8 can also permit users to key rings on a more granular basis than what is depicted here. In our example we are permitting the users globally to real any key ring.
Explaining the Solution: RACF Permissions for Working with Certificates and Key Rings

1. Define RACDCERT as an authorized TSO command in IKJTSOxx
2. Define IRR.DIGTCERT.function resources using RDEFINE RACF command
3. To issue RACDCERT, user must have one of the following authorities
   1. The SPECIAL attribute
   2. Sufficient authority to IRR.DIGTCERT.function resources
4. Permit IRR.DIGCERT.function to users

<table>
<thead>
<tr>
<th>Access</th>
<th>Permission</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td></td>
<td>control certificates for this user only</td>
</tr>
<tr>
<td>UPDATE</td>
<td></td>
<td>control certificates for other users too</td>
</tr>
<tr>
<td>CONTROL</td>
<td></td>
<td>control special certificates like CERTAUTH (Certificate Authority) certificates</td>
</tr>
</tbody>
</table>

EXAMPLES:
- PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(TCPIP) ACCESS(READ)
- PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(USER301) ACCESS(READ)
- PERMIT IRR.DIGCERT.LIST CLASS(FACILITY) ID(TCPIP) ACCESS(READ)
- PERMIT IRR.DIGCERT.LIST CLASS(FACILITY) ID(USER301) ACCESS(READ)

A SETROPTS command "refreshes" the controls for the certificate functions.
User executing this command requires the SPECIAL attribute.

RACF since V1R8 can also permit users to key rings on a more granular basis than what is depicted here. In our example we are permitting the users globally to real any key ring.
In this scenario, the FTP Server is unable to find the certificate it is supposed to present to the client during Server Authentication. The label of the certificate stored in RACF is “FTP Server on MVS1-MVS7”. The Key Ring name remains the same for FTPT server, but something has changed on the ring!
The FC2847 error with EDC8121I is difficult to diagnose, since these errno2 codes are not easy to find.

We must examine the meaning of all these SSL messages and the return codes. It appears that the remote end of the connection (The server) reset the connection. The server probably does not "like" what it saw when it tried to establish this connection.
Error #3: Diagnosis (1)

- IP MESSAGES, VOL. 2 (SC31-8784-11) for V1R12
  EZD1287I: Explanation: Application Transparent Transport Layer Security (AT-TLS) detected an error during the specified AT-TLS event

- IP DIAGNOSIS GUIDE (GC31-8782-11), Chapter 29, AT-TLS Return Codes
  RC 6 and RC438: Not documented here!

- Cryptographic Services SYSTEM SECURE SOCKETS LAYER Programming (SC24-5901-10)

At MVS3, RC 6: 6 Key label is not found.
  Explanation: The requested key label is not found in the key database, SAF key ring or z/OS PKCS #11 token.
  User response: Specify a label that exists in the key database, SAF key ring or z/OS PKCS #11 token.

At MVS1, RC 438: 438 Internal error reported by remote partner.
  Explanation: The peer application has detected an internal error while performing an SSL operation and has sent an alert to close the secure connection.
  User response: Check the error log for the remote application to determine the nature of the processing error.

The server reset the connection – not the client. RC438 seen at MVS1 confirms this. RC6 at MVS3 indicates a problem with finding the correct key label in the key database or ring. We need to look at the FTP Server's key ring on MVS3.
Error #3: Diagnosis (2)

Digital ring information for user FTPD:

<table>
<thead>
<tr>
<th>Certificate Label Name</th>
<th>Cert Owner</th>
<th>USAGE</th>
<th>DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVS1 LABS Certificate Authority</td>
<td>CERTAUTH</td>
<td>CERTAUTH</td>
<td>NO</td>
</tr>
</tbody>
</table>

Where is the FTP Server Certificate? It is not on the server's key ring!

We already suspect that there is something wrong with the key ring at MVS3, and so we display it and discover that the FTP Server certificate, which has been requested by the Client to establish secure communications, has been omitted from the appropriate key ring.

The AT-TLS log or trace does not provide us with any more information than we already had; it just confirms that there was a problem with the initial handshake.

We can correct the error as you see with our JCL on the next visual.
Error #3: Solution

Connect FTP Server Cert to Key Ring

RACDCERT ID(FTPD) CONNECT(ID(FTPD)
  - LABEL('FTP Server on MVS1-MVS7')
  - RING(Server_RING) USAGE(PERSONAL) DEFAULT
setropts generic(DIGTCERT) refresh
setropts raclist(DIGTCERT) refresh
racdcert ID(FTPD) listring(Server_RING)

Digital ring information for user FTPD:

Ring: Server_RING
Certificate Label Name  Cert Owner USAGE DEFAULT
-------------------------------- ------------ -------- -------
MVS1 LABS Certificate Authority CERTAUTH CERTAUTH NO
FTP Server on MVS1-MVS7  ID(FTPD) PERSONAL YES

1. Change Policy Instance Number and UPDATE PAGENTT to reinstall changes to Key Ring for FTP Server (“f pagent,update”), or ...
2. Recycle FTP Server to reinstall changes to Key Ring (Disruptive)

RACDCERT connects the missing certificate to the appropriate Key Ring; when we next display the key ring the certificate is finally there. We then need to refresh the FTP Server’s knowledge of a changed key ring and certificate with one of the two steps at the bottom of the page.

On the next visual you will see what needs to be on a key ring. These notes on this page contain the initial introduction to the concepts and the next visual provides more detail.

INTRO: What needs to be on a key ring or associated with a key ring??

A sender's key repository must contain all certificates that represent the full chain of trust for that sender.

A recipient's key repository must contain only the ROOT certificate associated with the sender's full chain of trust.

We show you here three examples of key ring configurations when using Server Authentication only for SSL/TLS or AT-TLS. In general, a TRUSTED Certificate or Certificate Chain and the PRIVATE Key of the End-Entity that owns the certificate is required. But whether or not the PRIVATE KEY need be available depends on whether mutual authentication is required or not. You see in the three examples that the client keyring needs no PRIVATE key associated with it if only Server Authentication is in use.
Read the INTRO to this concept in the notes on the previous page.

In the first example, you see that we have deployed only a server certificate. In this case, the server's key ring or key repository must have access to the server's private key as well as to its own server certificate.

The Client's key ring needs a copy of the self-signed server certificate in order to validate the certificate that the Server sends to it during SSL/TLS or AT-TLS negotiation.

In the second example, you see that we have deployed a server certificate that has been signed by a CA certificate (CA1). In this case, the server's key ring or key repository must have access to the server's private key as well as to its own server certificate. It must also have access to the CA certificate that has signed the server certificate.

The Client's key ring needs a copy of the ROOT CA certificate that has signed the Server certificate in order to validate the certificate that the Server sends to it during SSL/TLS or AT-TLS negotiation.

In the third example, you see that we have deployed a server certificate that has been signed by a CA certificate (CA2), which itself has been signed by another root CA certificate (CA1). In this case, the server's key ring or key repository must have access to the server's private key as well as to its own server certificate. It must also have access to both CA certificates.

The Client's key ring needs a copy of the Server's ROOT CA certificate, but to avoid problems in case the Server does not send the full chain of trust, both CA certificates (Root and Intermediate) that were used to sign the server certificate are included on the keyring.

EXAMPLE:

A local server key ring should contain:

The server certificate for the local End Entity.
The CA certificate that signed the local End Entity certificate.
The CA certificate that signed a remote End Entity certificate, if the negotiation requests Client Authentication.
The CA certificates that may have signed an Intermediate CA certificate that resides on the ring.
In this scenario, the FTP Server is still unable to find the certificate it is supposed to present to the client during Server Authentication. The label of the certificate stored in RACF is “FTP Server on MVS1-MVS7”. The Key Ring name remains the same for FTPT server, but something has changed again on the ring!
Error #4: View of Error Messages

234 Security environment established - ready for negotiation

FC2838 authServerAttls: Start Handshake

FC2847 authServerAttls: ioctl() failed on SIOCTTLSCTL - EDC8121I Connection reset. (errno2=0x77A9733D) <<<<<<<<< Different errno2 from previous example

EZA2897I Authentication negotiation failed

EZA1534I *** Control connection with 192.168.20.93 dies.

RC 428

As before, the FC2847 error with EDC8121I is difficult to diagnose, since these errno2 codes are not easy to find.

We must examine the meaning of all these SSL messages and the return codes again. Once again, it appears that the remote end of the connection (the server) reset the connection. The server probably does not “like” what it saw when it tried to establish this connection.

However, now we have a different TTLS Return Code to look at: RC 428.
Error #4: Diagnosis (1)

IP DIAGNOSIS GUIDE (GC31-8782-11), Chapter 29, AT-TLS Return Codes

RC 428 and RC 438: Not documented here!

• Cryptographic Services SYSTEM SECURE SOCKETS LAYER Programming (SC24-5901-10)

At MVS3 RC 428: Key entry does not contain a private key.
Explanation: The key entry does not contain a private key or the private key is not usable. This error can also occur if the private key is stored in ICSF and ICSF services are not available, ... Certificates that are meant to represent a server or client must be connected to a SAF keyring with a USAGE value of PERSONAL and either be owned by the userid of the application or be SITE certificates.
User response: Ensure that the ICSF started task has been started prior to the application if the private key is stored in ICSF. If executing in FIPS mode, ensure that the certificate being used does not have its private key stored in ICSF.

At MVS1 RC 438: Internal error reported by remote partner.
Explanation: The peer application has detected an internal error while performing an SSL operation and has sent an alert to close the secure connection.
User response: Check the error log for the remote application to determine the nature of the processing error.

The server reset the connection – not the client. RC438 seen at MVS1 confirms this. RC428 at MVS3 indicates a problem with finding the correct private key associated with the certificate that is supposed to be owned by the FTPT1 server. What we will discover is that RC of 428 can also mean simply that the certificate has been incorrectly added to the Key Ring – which is why the server cannot find the private key it needs.
Error #4: Diagnosis (2)

Digital certificate information for user FTPD:

Label: FTP Server on MVS1-MVS7
Certificate ID: 2QTG49fExuPXOKFmaWFmUCWjUDU5eLnYNTI4uhA
Status: TRUST
Start Date: 2012/09/08 00:00:00
End Date: 2015/12/31 23:59:59
Serial Number: >74<
Issuer's Name: >CN=MVS1CA.LABS.IBM.COM.O=MVS1CA.C=US<

( Subject's Name:
  >CN=FTP Server on MVS1-MVS7.OU=WSC.C=US<
Subject's AltNames:
  IP: 192.168.20.0
  Email: FTP at ZOS1
  Domain: WSC.IBM.COM

Key Type: RSA
Key Size: 1024
Private Key: YES

This is not an ICSF key – in some ways the error message is misleading. In other ways it is not: The explanation of the return code does point out that there is no PRIVATE key available to the FTP Server. Therefore,

1. we have either connected the correct certificate and its key INCORRECTLY to the key ring, or
2. we have failed to attach a certificate together with its key – possibly because it was exported or imported to us incorrectly! The display above shows that we DO have a private key and so we focus on bullet #1 Above.

We once again display the FTPD/Server_RING Key Ring to verify that the certificate for the FTP server resides on the correct ring. We also reissue the SETROPTS commands to verify that address space storage has been updated with the correct copy of the key ring. We also re-issue the commands to Policy Agent to have it re-read what is in storage. We test the client again, and obtain the same error.

Then we look at the contents of the Certificate to see if we find anything unusual there – for example, are we actually using ICSF keys when we thought we were not? Are the dates fine -- although this is a long shot, since our error codes are not indicating expired certificates? Is the certificate in TRUST status? What type of Key are we using for the private key? (Not an ICSF key.) Therefore, the error messages indicating that perhaps ICSF was required in order to correct the problem was misleading in our case.

Something else is wrong. What is it?
Error #4: Diagnosis (3)

Digital ring information for user FTPD:

<table>
<thead>
<tr>
<th>Ring:</th>
<th>Cert Owner</th>
<th>USAGE</th>
<th>DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;Server_RING&lt;</td>
<td>CERTAUTH</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>Certificate Label Name</td>
<td>Cert Owner</td>
<td>USAGE</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>MVS1 LABS Certificate Authority</td>
<td>CERTAUTH</td>
<td>CERTAUTH</td>
<td>NO</td>
</tr>
<tr>
<td>FTP Server on MVS1-MVS7</td>
<td>ID(FTPD)</td>
<td>PERSONAL</td>
<td>YES</td>
</tr>
</tbody>
</table>

SDSF DA MVS3     MVS3 PAG 0 CPU 3
COMMAND INPUT ===> PREFIX=FTP* DEST=(ALL) OWNER=* SYSNAME=
NP JOBNAME StepName ProcStep JobID Owner
FTPCCL1 STEP1 STC15829 TCPIP
FTPT1 STEP1 STC16246 TCPIP

The ICSF clue was a false lead. We must look for something else that does not look right. And we find it when we look at the key ring and look at the FTPT1 server. There it is: The server is running with an OMVS segment that identifies TCPIP as the owner; but the Certificate has been attached to the RACF key ring indicating something quite different: There, the certificate owner is identified as “FTPD” and not “TCPIP.”
Explaining the Solution: Who Must Own a Personal Certificate and its Key Pair?

• PERSONAL Certificate for a Server:
  1. ADDUSER FTPD DFLTGRP(OMVSGRP) OMVS(UID(0) HOME("") NOPASSWORD
  2. RDEFINE STARTED MYFTP*.* STDATA(USER(FTPD))
  3. RACDCERT ID(FTPD) GENCERT …

• PERSONAL Certificate for a Client if using Client Authentication (Assumption: Client is a human user)
  1. ADDUSER USER301 DFLTGRP(OMVSGRP) OMVS(UID(707) HOME(" /user301")) …
  2. RACDCERT ID(USER301) GENCERT …

The PUBLIC key of the key pair resides in the Personal Certificate; the PRIVATE key of the key pair is associated with the Personal Certificate and its OMVS Segment Owner.

1) PERSONAL Certificate for a Server:
Define a USERID and assign an OMVS (UNIX) Identity to it:
   ADDUSER FTPD DFLTGRP(OMVSGRP) OMVS(UID(0) HOME("") NOPASSWORD
Create a Started Class definition for the Server and associate it with its OMVS Segment (i.e., its USERID or OWNER)
   RDEFINE STARTED MYFTP*.* STDATA(USER(FTPD))
Associate an x.509 Server Certificate with its OMVS Segment OWNER
   RACDCERT ID(FTPD) GENCERT …

2) PERSONAL Certificate for a Client (Assumption: Client is a human user)
Define a USERID and assign an OMVS (UNIX) Identity to it:
   ADDUSER USER71 DFLTGRP(OMVSGRP) OMVS(UID(707) HOME(" /user71")) …
Associate an x.509 Client Certificate with its OMVS Segment OWNER
   RACDCERT ID(USER71) GENCERT …
The ICSF clue was a false lead. We must look for something else that does not look right. And we find it when we look at the key ring and look at the FTPT1 server. There it is: The server is running with an OMVS segment that identifies TCPIP as the owner; but the Certificate has been attached to the RACF key ring indicating something quite different: There, the certificate owner is identified as “FTPD” and not “TCPIP.”

For the solution we choose to disconnect the certificate from the key ring and then reconnect it while specifying the correct OMVS Segment Userid.
Error #4: Solution (2)

1. Disconnect the certificate from the key ring.
2. CAUTION: Delete the certificate if not being used elsewhere.
   1. Otherwise generate a new certificate and specify the correct OMVS Segment Owner of the started task, FTPT1.
3. Connect new certificate to the keyring.

Or you could instead just choose to associate the started task with the owner of the existing certificate.

This visual describes the steps you could take to correct the problem.
Error #5a: When the FTP Client Desires DNS Validation

0335304B Certificate not valid for host.

Explanation: A server certificate does not contain the current host name as either the common name (CN) element of the subject name or as a DNS entry for the subject alternate name.

User response: Obtain a new certificate containing the desired host name.

1. Some clients will validate a server certificate
   1. Verify that hostname exists in an application-internal listing or with a Resolver Call
      1. (CN) of Subject Name (Certificate "Subject's Name")
      2. Subject Alternate Name (Certificate "Domain")
   2. Application Coding or Certificate or DNS (Host.Local) must be correct
      1. Change any of these
      1. Depends on how the application handles the feature

Here you see an example of an option you can set in the FTP Client.

Not all clients avail themselves of this feature to request validation of a server certificate by checking to see if its entries match anything in the Client's DNS or Host.Local file.

You must know your client before you request a certificate ... or you must plan for all cases and use the ALTNAMES certificate extensions so that, if necessary, you can update a DNS or a Host Local file with a new entry. This is usually less expensive than buying a new certificate. (If you are creating your own certificates expense or time delays are probably not an issue.)

System SSL will validate the host name against the DNS entry in the subject alternate name extension.

The host name in the certificate can be

• a fully-qualified name (e.g., 'dcesec4.endicott.ibm.com'),
• a domain suffix (e.g., '.endicott.ibm.com') or
• a wildcard name beginning with an asterisk (e.g., '*.endicott.ibm.com').

A case-sensitive comparison is performed between the supplied host name and the host name in the certificate. A fully-qualified name must be the same as the supplied host name. A domain suffix matches any host name with the same suffix but does not match the suffix itself. For example, '*.endicott.ibm.com'

The status code in the example is a Certificate Management Services (CMS) Status Codes as documented in the Cryptographic Services System Secure Sockets Layer Programming manual (SC24-5901-07).

This type of feature allows an application to provide a list of hostnames that AT-TLS will compare against the hostname in the certificate. A return code back to the application indicates whether a match was found or not. Based on that return code, the application can decide whether or not to continue the session. Obviously, though, this approach relies on the application to determine the acceptable hostnames, be it through DNS or any other method.
Error #5b: When the TN3270 Client Desires DNS Validation

Personal Communications
TN3270 Client Emulator:
Security Fields (Server Authentication)

0335304B Certificate not valid for host.
Explanation: A server certificate does not contain the current host name as either the common name (CN) element of the subject name or as a DNS entry for the subject alternate name.
User response: Obtain a new certificate containing the desired host name.

1. Some clients will validate a server certificate
   1. Verify that hostname exists in an application-internal listing or with a Resolver Call
   1. (CN) of Subject Name (Certificate "Subject's Name")
   2. Subject Alternate Name (Certificate "Domain")
2. Application Coding or Certificate or DNS (Host.Local) must be correct
   1. Change any of these
   1. Depends on how the application handles the feature

Here you see an example of an option you can set in the TN3270 Server.
Not all clients avail themselves of this feature to request validation of a server certificate by checking to see if its entries match anything in the Client's DNS or Host.Local file.
You must know your client before you request a certificate ... or you must plan for all cases and use the ALTNNAME certificate extensions so that, if necessary, you can update a DNS or a Host Local file with a new entry. This is usually less expensive that buying a new certificate. (If you are creating your own certificates expense or time delays are probably not an issue.)
System SSL will validate the host name against the DNS entry in the subject alternate name extension.
The host name in the certificate can be
   • a fully-qualified name (e.g., 'dcesec4.endicott.ibm.com'),
   • a domain suffix (e.g., '.endicott.ibm.com') or
   • a wildcard name beginning with an asterisk (e.g., '*.endicott.ibm.com').
A case-sensitive comparison is performed between the supplied host name and the host name in the certificate. A fully-qualified name must be the same as the supplied host name. A domain suffix matches any host name with the same suffix but does not match the suffix itself. For example, '*.endicott.ibm.com'
The status code in the example is a Certificate Management Services (CMS) Status Codes as documented in the Cryptographic Services System Secure Sockets Layer Programming manual (SC24-5901-07).
This type of feature allows an application to provide a list of hostnames that AT-TLS will compare against the hostname in the certificate. A return code back to the application indicates whether a match was found or not. Based on that return code, the application can decide whether or not to continue the session. Obviously, though, this approach relies on the application to determine the acceptable hostnames, be it through DNS or any other method.
Error #5: Solution

Best practice: Develop Standards for fields of the x.509 Certificate so that the CN field corresponds to what will be recorded in a DNS or a Local Host file.

1. Change Client Configuration to eliminate the DNS Lookup
2. Add a DNS entry or a Local Host entry or an IPNodes entry that matches what is being validated

This visual shows you the contents of a certificate and advises your company to establish standards for the fields that comprise the x.509 certificate.
Explaining the Solution: Certificate Contents, Name Lookup Requested by Client

You see in the visual above how there is a DNS entry that matches the name in “CN” portion of the distinguished name on the certificate. In some implementations, it is only necessary that there be a matching name, but the IP address if mentioned in the certificate or if used on the connection request need not correspond to the name. With IPSec, you can still have a DNS or local host matching error, but in cases here the IP address must match the endpoint address of the connection partner.

You may also use a local host file or an ipnodes file if the name lookup with such a file is preferred or is being used. Best Practice is to avoid use of the UNIX /etc/hosts file or the MVS hlq.HOSTS.ADDRINFO and hlq.HOSTS.SITEINFO and use the IPNODES file that is identified in the System Resolver Setup File. Together with the COMMONSEARCH parameter in the RESOLVER SETUP file, the IPNODES file is able to be used for both UNIX and MVS searches as well as for IPv4 and IPv6 searches.

(NOTE: The MVS Versions –ADDRINFO and SITEINFO – are created with the MAKESITE command that is executed against the hlq.HOSTS.LOCAL file.)

Example of HOSTS.LOCAL File entry that will produce the appropriate ADDRINFO and SITEINFO files:

HOST : 10.1.1.13: FTPXSRV31 ::::
Error #6: Expired x.509 Certificate and Keys

MVS Console Log at FTP Server LPAR

EZD1287I TTLS Error RC: 401 Initial Handshake 036
LOCAL: 192.168.20.113..21
REMOTE: 192.168.20.111..1038
JOBNAME: FTPT1 RULE: FTPT@192.168.20.113 2
USERID: TCPIP GRPID: 00000002 ENVID: 00000009 CONNID: 000000AD

SYSLOGD Output with AT-TLS Policy Trace Level of 255 at Client LPAR

EZD1284I TTLS Flow GRPID: 00000001 ENVID: 00000006 CONNID: 0000000CA
RC: 401 Call GSK_SECURE_SOCKET_INIT - 7EB3C318

EZD1283I TTLS Event GRPID: 00000101 ENVID: 00000006 CONNID: 0000000CA
RC: 401 Initial Handshake 00000000 7EB9C798

EZD1286I TTLS Error GRPID: 00000001 ENVID: 00000006 CONNID: 0000000CA
LOCAL: 192.168.20.111..1038 REMOTE: 192.168.20.113..21 JOBNAME:
USERID: USER201 USERID: USER301 RULE: FTPTClient@192.168.20.11n-5
RC: 401 Initial Handshake 00000000 7EB9C798

• If you attended this Certificate Lab, you saw the messages above!
  • SHARE San Francisco 12895
  • “Renewing and Rekeying RACF x.509 Digital Certificates”

This visual shows you the SSL Return Code that you would see if your certificate cannot be found due to an invalid status of one type or another.
Error #6: Diagnosis

Cryptographic Services SYSTEM SECURE SOCKETS LAYER Programming (SC24-5901-10)

At MVS1 and MVS3 -- RC 401:

401 Certificate is expired or is not valid yet.

Explanation: The current time is either before the Certificate start time or after the Certificate end time.

User response: Obtain a new Certificate if the Certificate is expired or wait until the Certificate becomes valid if it is not valid yet.

Digital certificate information for user TCPIP:

Label: FTPServer31 EXP
Certificate ID: 2QXjw9fJ18bj1+KFmaWmFLyQMn10BA
Status: TRUST
Start Date: 2008/10/07 00:00:00
End Date: 2011/10/07 23:59:59

During the handshake the FTP Server sent the client its Certificate and this Certificate is not valid. We display the Certificate on the shared RACF database to determine what the validity dates are and discover that the certificate has expired dates.
Error #6: Solution

To renew the expiration dates of a Certificate, follow these steps:

1. Generate a Certificate Request for the Certificate with the invalid dates
   (**RACDCERT GENREQ** command)

2. Generate a new Certificate, keeping the original old date, but extending the new date
   by one year. (**RACDCERT GENCERT** command)

3. Mark the Certificate as TRUSTED – since the old date will cause it to default to
   UNTRUSTED. (**RACDCERT ALTER** command)

- If you attended this Certificate Lab, you executed these steps to complete the solution.
- SHARE San Francisco 12895
  - “Renewing and Rekeying RACF x.509 Digital Certificates”

Here we show you the steps to extend the lifetime of the certificate by changing the
expiration date but retaining the original key pair.
Summary of Common SSL/TLS Problems

- Wrong Owner of Key Ring that Certificates reside on
- Wrong Owner of Personal Certificate
- Missing certificate (either a personal or a CA certificate)
  - Or Correct Certificate Label cannot be found or Incorrect designation as Default Certificate
- Expired Certificate
- Signing authority certificate not available to validate a certificate received during negotiation of secured session
- RACF or Certificate Repository mismanaged: Incorrect Permissions granted.
- Exporting CA Certificate Signing Keys / Private Keys without understanding consequence
  - Failing to export or import with the Private key when it is necessary for the protocol (i.e., not understanding certificate formats)
- Hostname Lookup specified
  - Mismatch in the value returned by the Resolver Function
  - Contents of key ring changed during lifetime of AT-TLS connection
- Encryption Algorithm not Supported by Platform software or hardware
- Security technology: SSL may support it, but IPSec does not
- Etc.
This recaps what you have heard about here. It emphasizes that you need to know a bit about the protocols to be able to perform problem determination, but you may not have to resort to taking a System SSL trace to solve a problem. It could be that simpler diagnostic tools – like logs, messages, and documentation on the meaning of those messages may be enough, as you have seen here.
Resources for Diagnosing z/OS SSL/TLS or AT-TLS Errors

- z/OS Cryptographic Services System Secure Sockets Layer Programming (SC24-5901)
  - also known as: System SSL Programming Guide
- z/OS Communications Server IP Diagnosis Guide (GC31-8782)
- z/OS Communications Server IP Messages: Vol. 1 (EZA) (SC31-8783)
- z/OS Communications Server IP Messages: Vol. 2 (EZB, EZD) (SC31-8784)
- z/OS Communications Server IP Messages: Vol. 3 (EZY) (SC31-8785)
- z/OS Communications Server IP Messages: Vol. 4 (EZZ, SNM) (SC31-8786)
- z/OS Security Server RACF Command Language Reference (SA22-7687)
End of Presentation

Medical School: Diagnosing SSL/TLS and AT-TLS Problems in z/OS CS

Speaker: Gwendolyn J. Dente (gdente@us.ibm.com)
IBM Advanced Technical Support (ATS)
Gaithersburg, Maryland (USA)

Friday, February 8, 2013:
9:30 AM-10:30 AM
(San Francisco Hilton, Golden Gate 3, Lobby Level)