An Introduction to AT-TLS for FTP and TN3270

Chelsea Jean-Mary
Mike Rich

Senior Technical Advisor:
Linda Harrison

Trademarks

The following are Registered Trademarks of the International Business Machines Corporation in the United States and/or other countries.

- IBM
- z/OS

The following are trademarks or registered trademarks of other companies.

- Microsoft is a registered trademark of Microsoft Corporation in the United States and other countries.

All other products may be trademarks or registered trademarks of their respective companies.

Refer to www.ibm.com/legal for further legal information.

Introduction

Chelsea Jean-Mary
Client Technical Specialist
Washington Systems Center

Mike Rich
CyberSecurity Specialist, Guardium for z/OS
IBM Z Security

Contents

Background

SSL/TLS

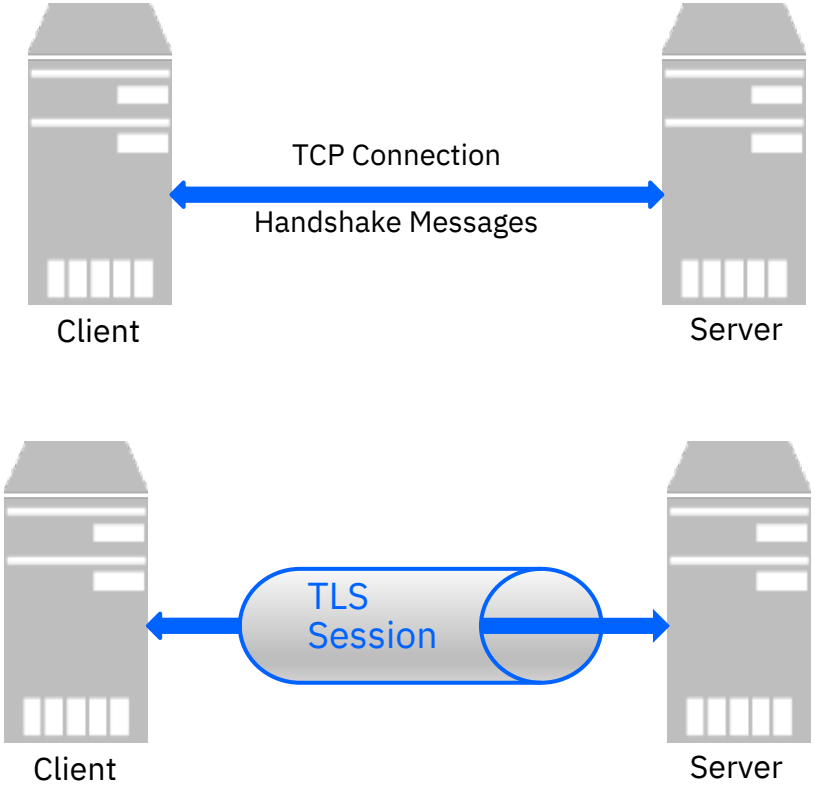
- Who, What, Where, When, Why, and How?
- SSL is TLS's predecessor
- Cryptographic protocols designed to provide communications security over a computer network

Benefits:

- ✓ Encryption
- ✓ Authentication

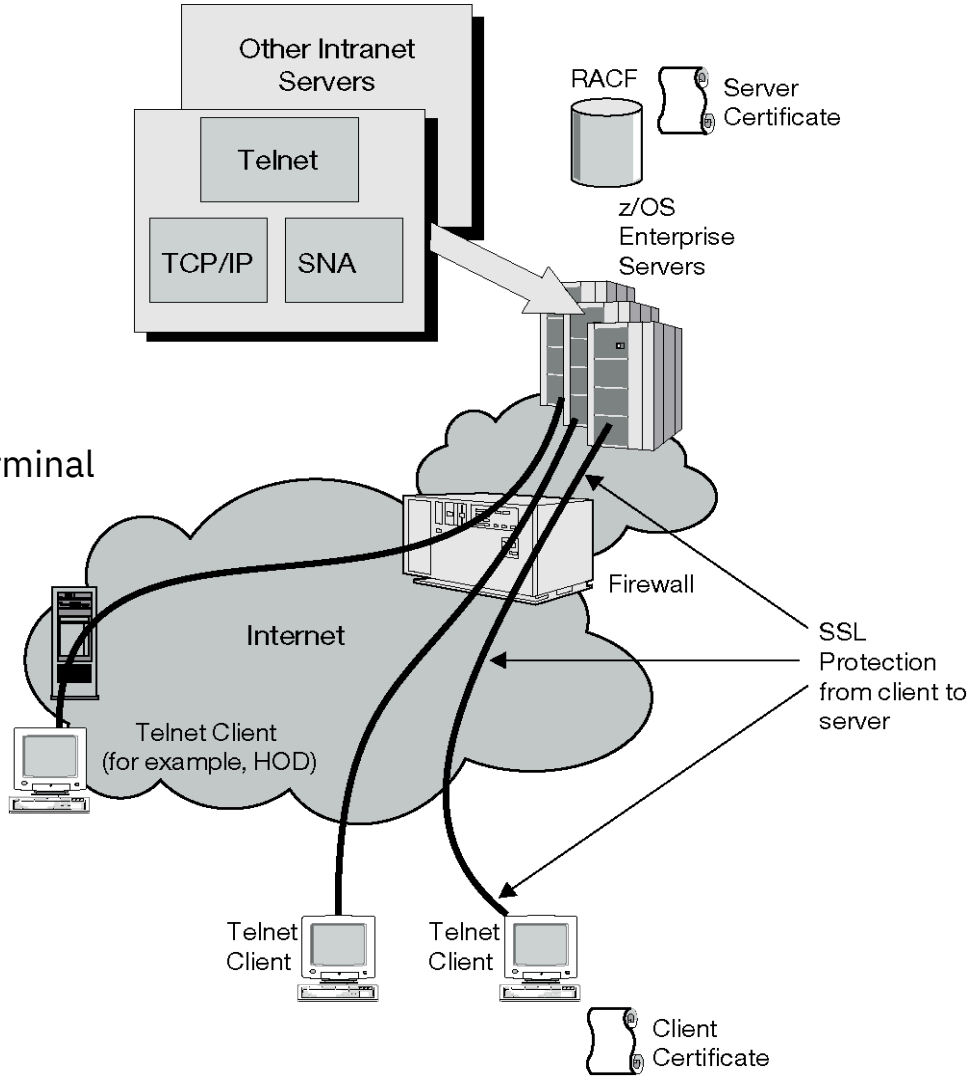
Goal/Result:

- ✓ Trusted End-to-End Communication



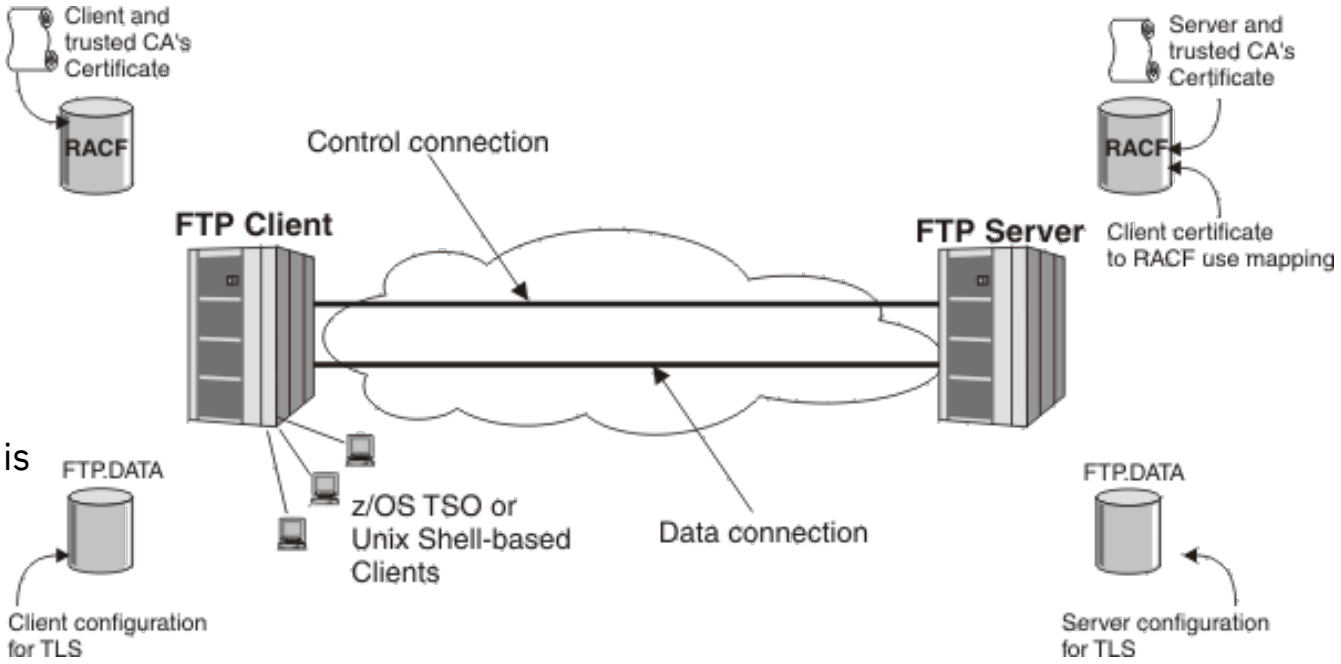
SSL on z/OS: TN3270

- Telnet Client Connection
- RACF Authentication
- Z Server Certificate
- A Secure SSL Connection is made
  - Secure transmission between TN3270 Terminal and z/OS



TLS on z/OS: FTP

- Conditional vs. Nonconditional mode
- Both have their TLS Configuration
- Option RACF Authentication
- The control connection is made to establish the handshake
- Then a secure data connection is made for transmission over the wire



Application Transparent Transport Layer Security (AT-TLS)

What is Application Transparent Transport Layer Security (AT-TLS)?

- *Creates a secure session on behalf of applications*
- Instead of implementing TLS in every application that requires a secure connection, AT-TLS provides encryption and decryption of data based on policy statements that are coded in the Policy Agent.
- The application sends and receives cleartext (unencrypted data) as usual while AT-TLS encrypts and decrypts data at the TCP transport layer.
- Basically, establishment of TLS connection does not require application involvement.

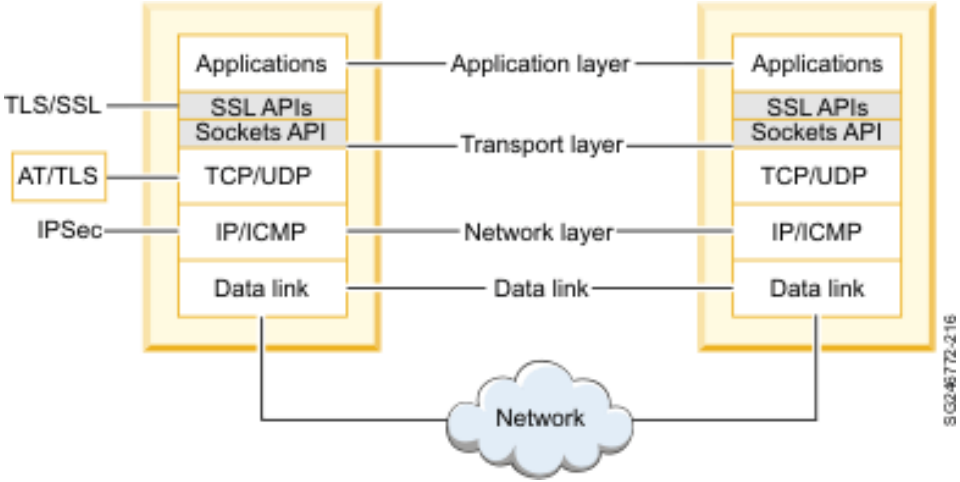


Figure: AT-TLS in the IP layer model

What is Application Transparent Transport Layer Security (AT-TLS)?

- Three types of applications that use AT-TLS:
  - Basic – unaware of AT-TLS
  - Aware – aware of AT-TLS and can query information
  - Controlling – aware of AT-TLS and needs to control the secure session

Why use AT-TLS?

- Cost reduction
  - Reduces development costs for exploiting TLS on applications
- Single, system-wide AT-TLS policy
  - Compared to an application-specific policy
- Ongoing performance improvements
  - As features are added, application can use them by changing AT-TLS policy, not code

How do you use AT-TLS on z/OS?

There are many options required for enabling AT-TLS on z/OS:

- Set the TTLS parameter on the TCPCONFIG statement in the PROFILE.TCPIP dataset
- Define the EZB.INITSTACK resource profile in the SERVAUTH class to control whether an application can access the TCP/IP stack before required policies have been installed.

The next few slides outline how to enable AT-TLS for FTP and telnet.

Enabling AT-TLS for FTP

- Customizing security parameters of the FTP client/server program requires the use of a configuration file known as FTP.DATA. The following parameters are set in the ”Security Options” section of the FTP.DATA file:

FTP.DATA

TLSMECHANISM	ATTLS
EXTENSIONS	AUTH_TLS
SECURE_CTRLCONN	Private
SECURE_DATACONN	Private
SECURE_ FTP	Required
SECURE_MECHANISM	TLS

- Execute the following **ftp** command in Unix System Services to establish a secure connection:

  ftp –t TLS ipaddress

Enabling AT-TLS for telnet

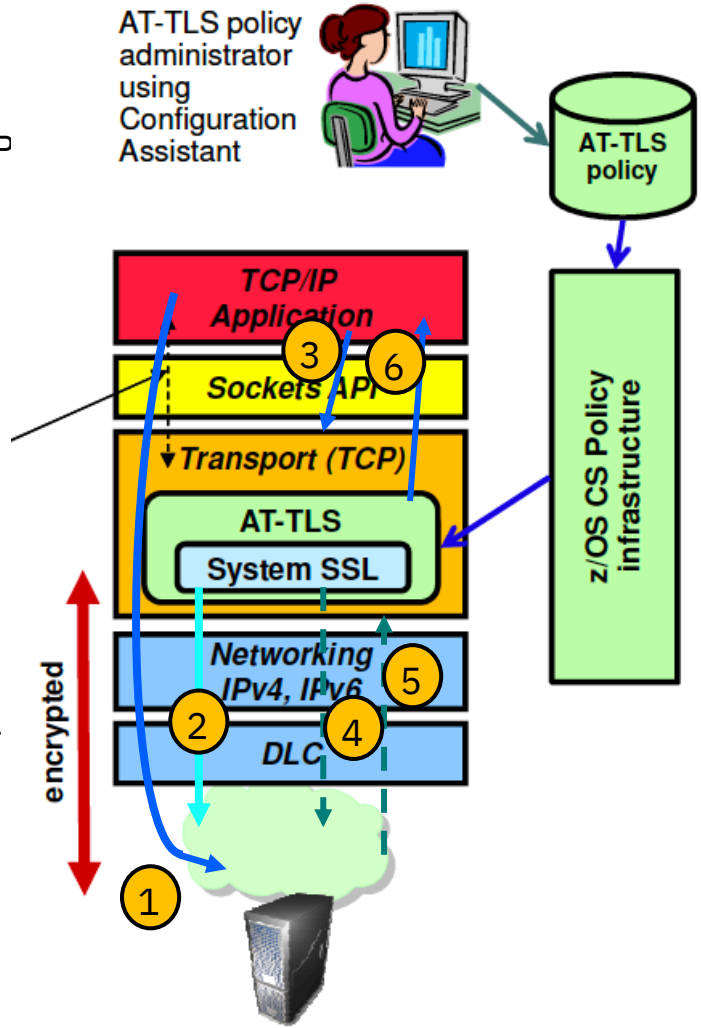
- The startup JCL has a PROFILE DD statement that points to a profile data set that contains parameters to control the TN3270E server. Within this profile data set are statements blocks used to define TN3270E server behavior:
  - Set the TTLSPORT parameter statement to define the port number that Telnet will listen on for secure connection requests from a client that point to the AT-TLS port
  - Set the CONNTYPE parameter to Secure in the TN3270.PROFILE dataset

- Personal Communications (PCOMM) Security Setup:
  - Enable Security and Microsoft CryptoAPI (MSCAPI)

How It Works (z/OS as client)

Setup: AT-TLS policy is configured and deployed for the TCP application and the TCP application is started.

1. z/OS client connects to server and connection is established
2. TCP layer invokes System SSL to perform the TLS handshake under identity of the client application
3. z/OS client sends data to server
4. TCP layer invokes System SSL to encrypt queued data and then sends it to server
5. Server sends encrypted data. TCP layer invokes System SSL to decrypt it
6. TCP delivers inbound data to z/OS client in the clear



Unencrypted (cleartext) flows

SSL/TLS handshake flows

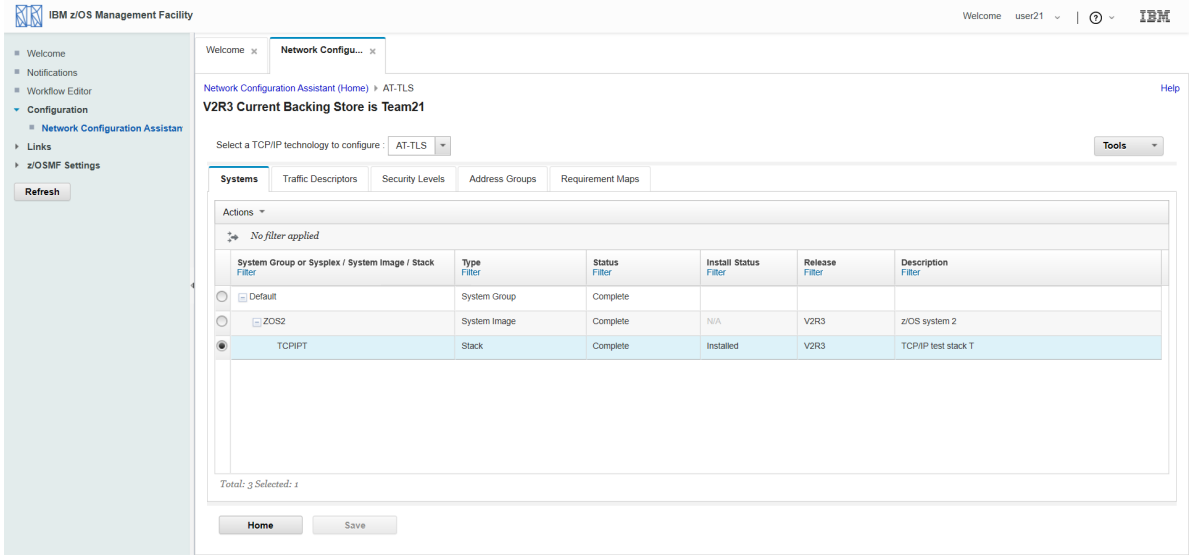
SSL/TLS-secured (encrypted) flows

Configuration Assistant

Configuration Assistant

- AT-TLS Policies
- Create and Manage TCP/IP profiles
- Network Security Services (NSS)
- Quality of Service (QoS)
- Intrusion Detection Service (IDS)

And more…



Policy Agent

Services:

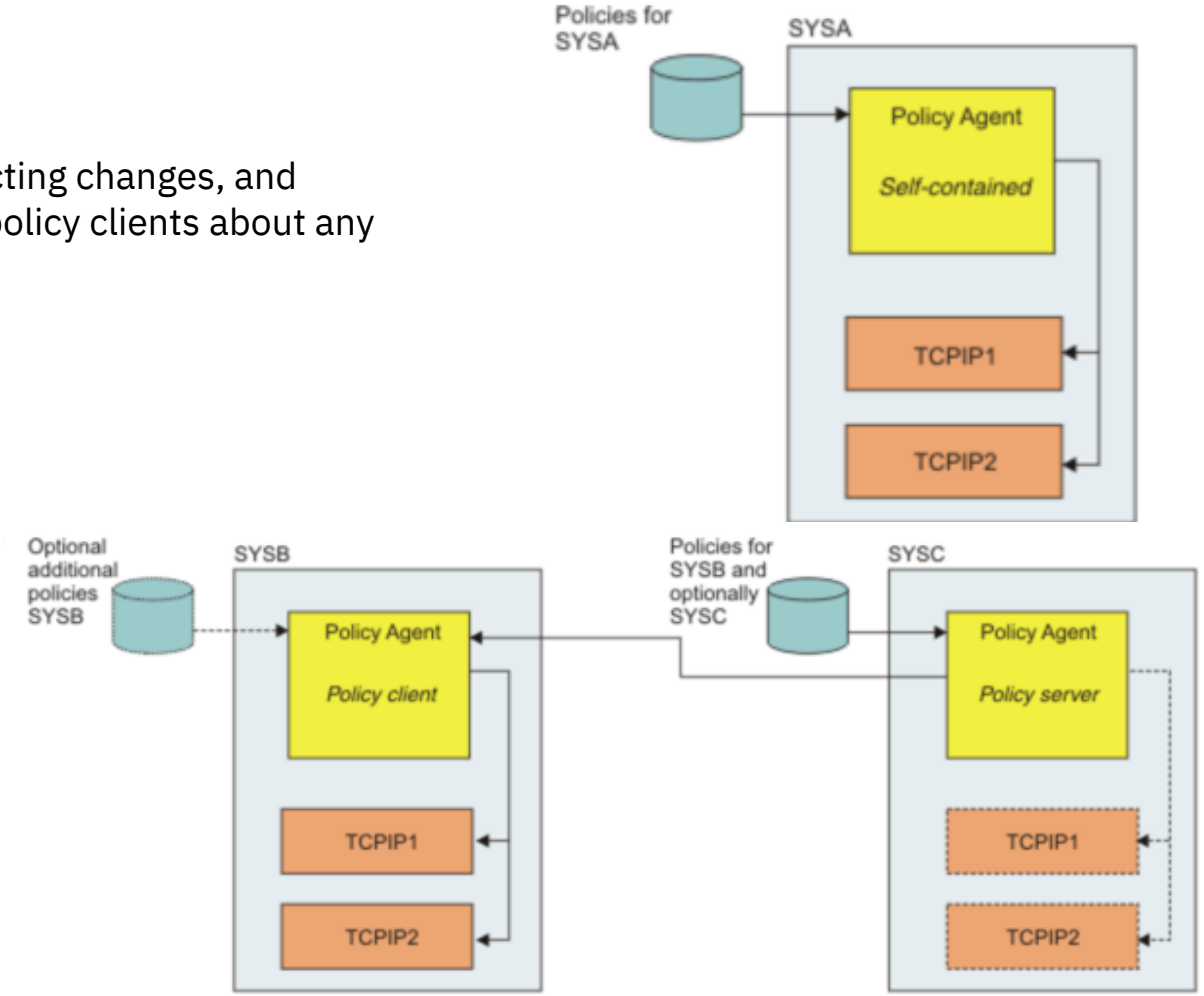
- Monitoring the policy files, detecting changes, and informing the TCP/IP stacks or policy clients about any changes.

Roles:

- Policy Decision Point (PDP)
- Policy Client
- Centralized Policy Server

More...

Figure 1. Policy Agent roles



General Category, versus Protocol, versus Command, versus Parameter

IP security – Any security feature that protects IP traffic

IPSec - The protocol suite.

ipsec - The command associated with an IP filter action, or the z/OS UNIX System Services command.

IPSEC - The statement in the TCP/IP profile

IPSECURITY - The parameter on the IPCONFIG statement in the TCP/IP profile

Commands

ipsec

- Use the z/OS UNIX System Services ipsec command to display information about active filters and Security Associations, and to control aspects of Security Association negotiation

pasearch

- Use the pasearch command to display Policy Agent information that is defined in the Policy Agent configuration files including IP security and other types of policies

MODIFY

- Policy Agent rereads the Policy Agent configuration files

Netstat

- IPSECURITY status display for a particular stack (Netstat CONFIG/-f)

Errors

Common AT-TLS errors

- Server cannot find its Key Ring
- Sever not authorized to read key ring
- Server cannot find its certificate or private key
- Client configuration specifies a DNS lookup that does not exist
- Server certificate has expired

How to look up errors

- NETSTAT command - helps in problem determination and status checking of connections
- Increase log level of SYSLOGD.
- Increase debug level for TN3270.
- Run a CTRACE with option TCP or a packet trace
- Set debug traces using the TTLSConnectionAction statement
- SSL trace may be run using System SSL

Error messages

- EZD1286I – issued to syslogd to report any errors that occur on a connection when the trace level 2 (Error) is set

- EZD1287I – issued to TCP/IP job log to report any errors that occur on an AT-TLS connection when the trace level 1 (Error) is set

- Return codes -
  - 5001 – 5999 describe AT-TLS can be corrected by the user
  - 6001 – 6999 describe internal AT-TLS errors. Contact IBM with error information and syslog information (if available)

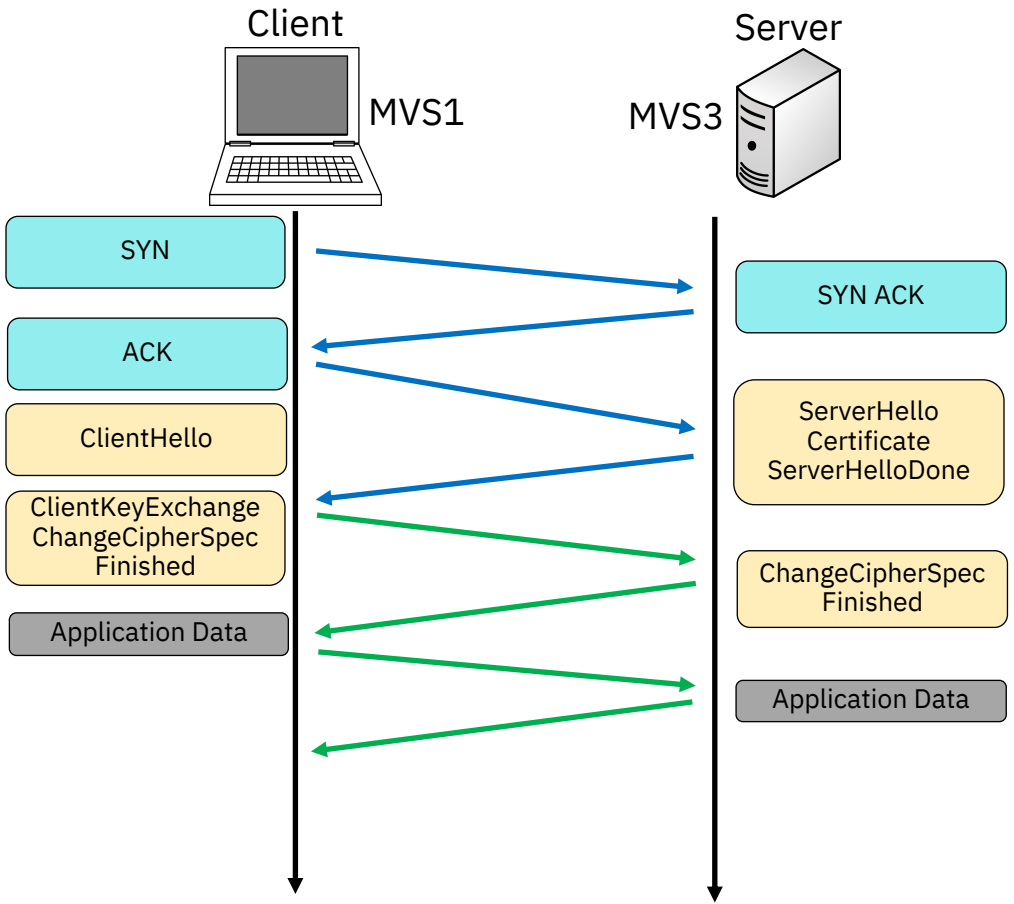
AT-TLS Return Codes are documented in two different places:

- z/OS Communications Server V2R4 IP Diagnosis Guide, GC27-3652-40, Chapter 27 Diagnosing Application Transparent Transport Layer Security (AT-TLS)

- z/OS V2R4 Cryptographic Services System Secure Sockets Layer Programming, SC14-7495-40, Chapter 13 Messages and code

Error Example #1: Server Cannot Find its Key Ring

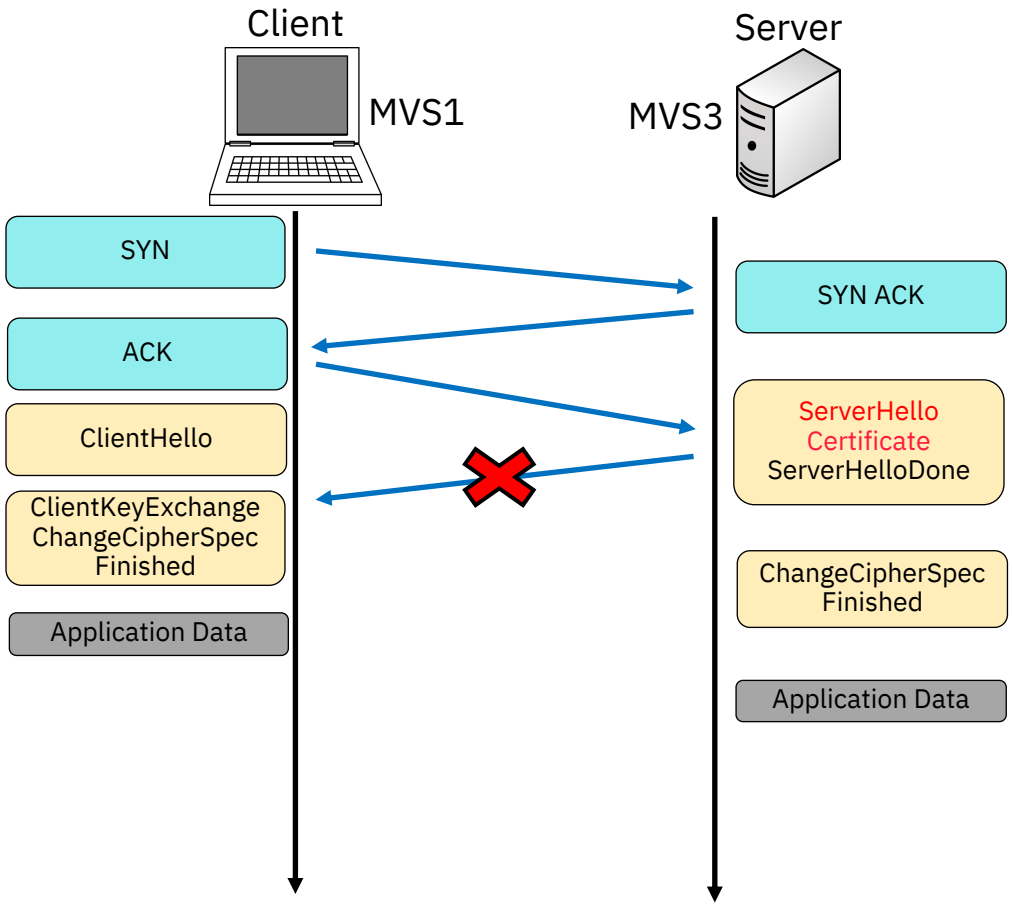
The following flow is seen when a client connects to an AT-TLS secured port.



Error Example #1: Server Cannot Find its Key Ring

The following flow is seen when a client connects to an AT-TLS secured port.

In this example, the FTP server cannot find its key ring.



Error Example #1: Server Cannot Find its Key Ring

The following shows the error messages. The server reset the connection, n

```
EZA1701I >>> AUTH TLS
234 Security environment established - ready for
FC2838 authServerAttls: Start Handshake
FC2847 authServerAttls: ioctl() failed
reset.  (errno2=0x74520442)
EZA2897I Authentication negotiation fai
```

AT MVS3 (SERVER):	EZD1287I TTLS Error RC: 202	Environment
AT MVS3 (SERVER):	EZD1287I TTLS Error RC: 202	Environment Link
AT MVS1 (CLIENT):	EZD1287I TTLS Error RC: 406	Initial Handshake
AT MVS3 (SERVER):	EZD1287I TTLS Error RC: 5006	Initial Handshake

The error messages point to a problem at the Server side related to the Handshake.

RC 202 indicates the server may not be able to find its keyring.

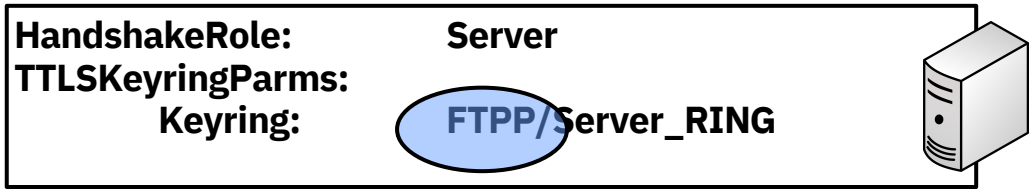
"The key ring cannot be opened because the user does not have permission."

"An I/O error occurred on the socket. This occurs if the TCP socket is closed."

"The connection is using a TTLSEnvironmentAction statement that failed to initialize a System SSL environment."

Error Example #1: Server Cannot Find its Key Ring

The following shows the pasearch output at MVS3 (server).



There is a typo in the AT-TLS policy where the owner of the Key Ring is spelled wrong.

It should be “FTPD” not “FTPP”.

Error Example #1: Server Cannot Find its Key Ring

Solution:

1. Change the policy.
2. Change Policy Instance number and UPDATE PAGENTT to reinstall changes to the Key ring for FTP Server.
3. Recycle FTP Server to reinstall changes to Key Ring.

```
TTLSKeyringParms          keyR1
{
        Keyring           FTPD/Server_RING <<<<<<<<<
```

```
TTLSEnvironmentAction eAct1~FTPTat192.168.20.9n
{
        HandshakeRole             Server
        EnvironmentUserInstance 1 <<<<<<<<<< was 0 before
        TTLSKeyringParmsRef       keyR1
}
```

Error Example #1: Server Cannot Find its Key Ring

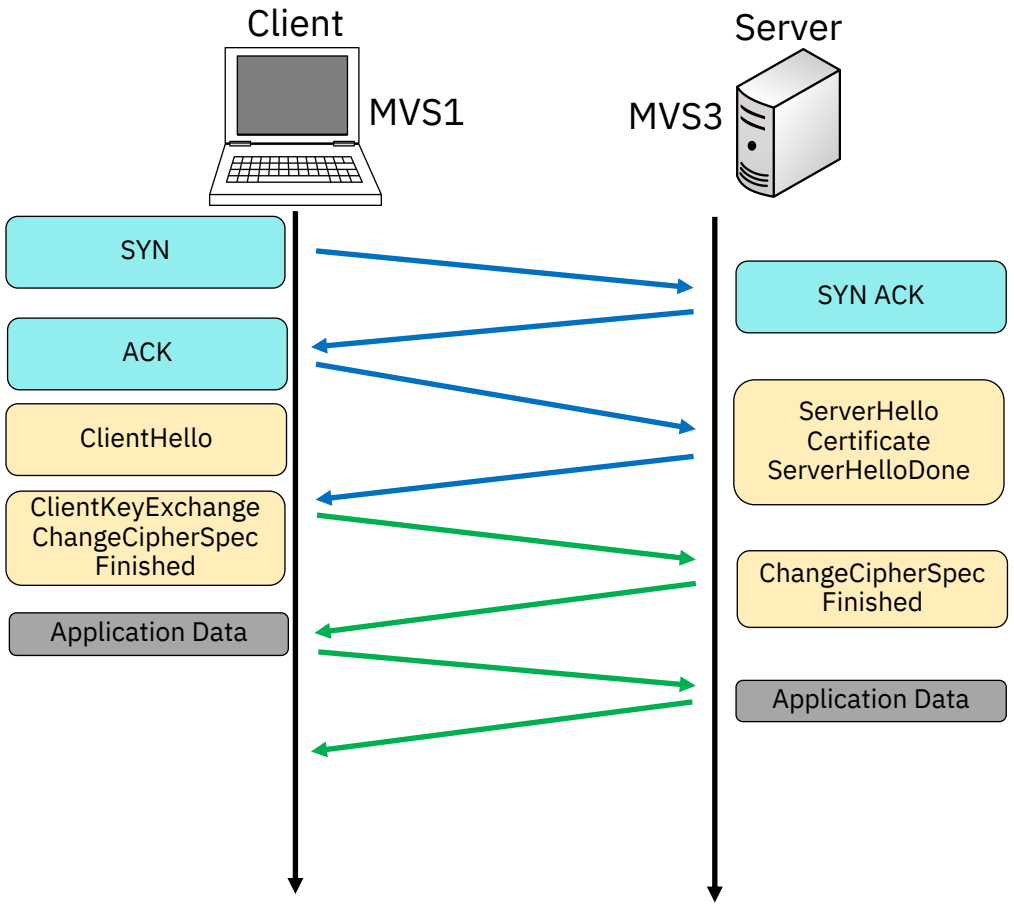
The following are the messages now seen by the client.

```
234 Security environment established - ready for negotiation
FC2838 authServerAttls: Start Handshake
FC2869 authServerAttls: FIPS140 not enabled
FC2890 authServerAttls: Using TLSv1.1 protocol
FC2904 authServerAttls: SSL cipher: 0A
FU1486 getCtrlConnCertAttls: Request certificate, size 673
EZA2895I Authentication negotiation succeeded
FC1777 setdlevel: entered
FC1938 setpbsz: entered
EZA1701I >>> PBSZ 0
200 Protection buffer size accepted
EZA1701I >>> PROT P
200 Data connection protection set to private
EZA2906I Data connection protection is private
EZA1459I NAME (192.168.20.93:USER301):
```

Notice that we can recognize the actual cipher and TLS protocol being used for successful negotiation and establishment.

Error Example #2: Server Cannot Find its Certificate

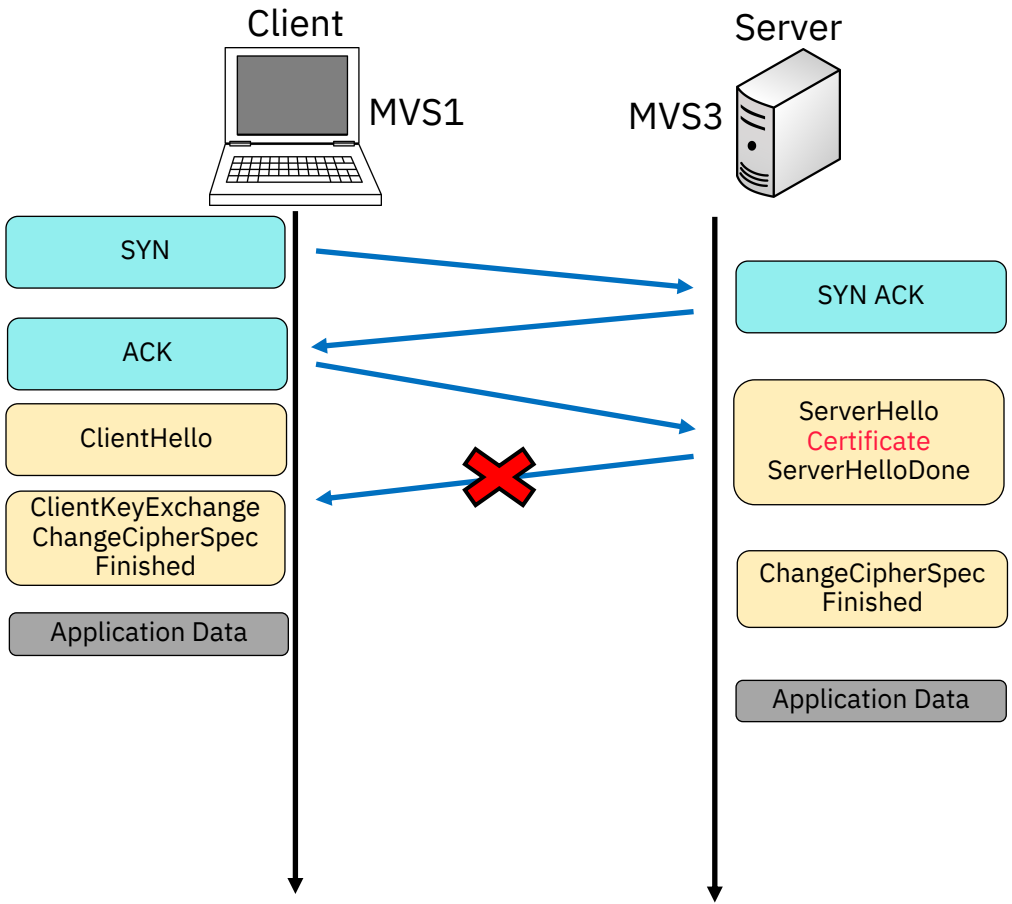
The following flow is seen when a client connects to an AT-TLS secured port.



Error Example #2: Server Cannot Find its Certificate

The following flow is seen when a client connects to an AT-TLS secured port.

In this case, the FTP server fails to provide a certificate.



Error Example #2: Server Cannot Find its Certificate

The following shows the error messages. The server reset the connection, not the client.

```
234 Security environment established - ready for negotiation

FC2838 authServerAttls: Start Handshake

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

EZA2897I   Authentication negotiation failed

EZA1534I *** Control connection with 192.168.20.93 dies.
```

AT MVS3 (SERVER) :	EZD1287I TTLS Error RC: 6	Initial Handshake
AT MVS1 (CLIENT) :	EZD1287I TTLS Error RC: 438	Initial Handshake

Error Example #2: Server Cannot Find its Certificate

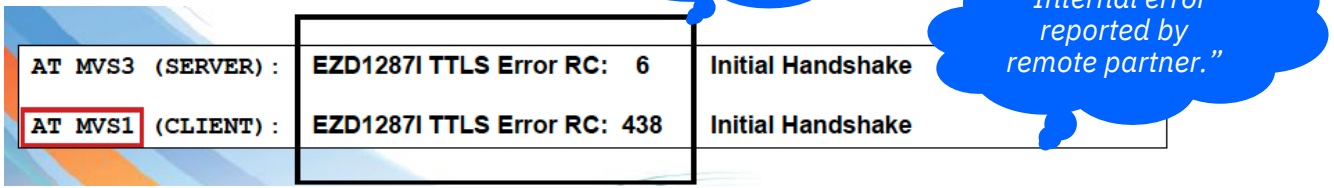
The following shows the error messages. The server reset the connection, not th

MVS (server) indicates a problem with finding the correct key label in the key database or ring.

```
234 Security environment established - ready for negotiation

FC2838 authServerAttls: Start Handshake

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

EZA2897I  Authentication negotiation failed

EZA1534I *** Control connection with 192.168.20.93 di
```

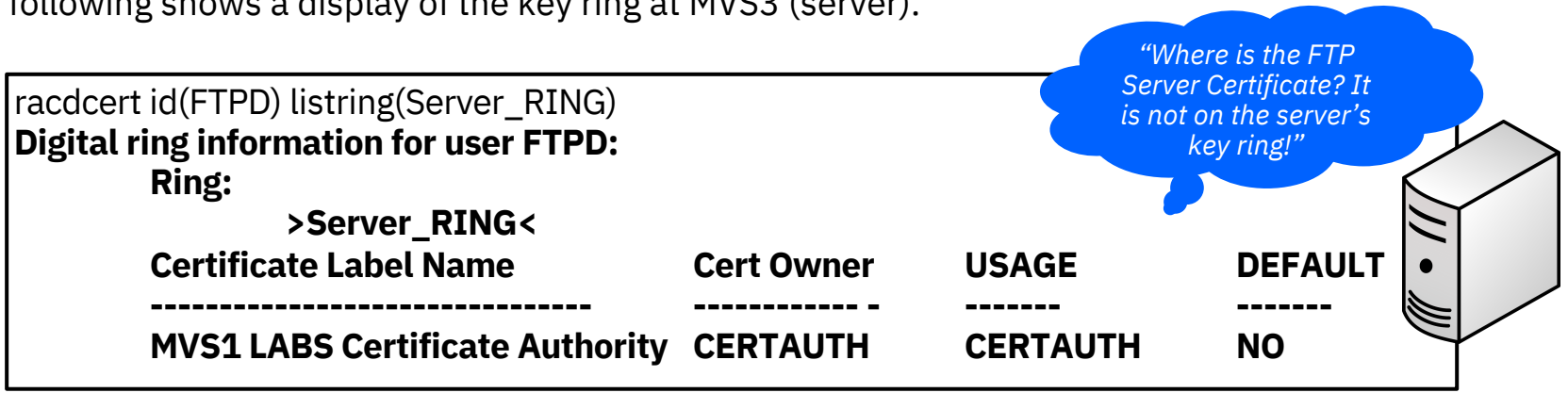
"Key label is not found."

"Internal error reported by remote partner."



Error Example #2: Server Cannot Find its Certificate

The following shows a display of the key ring at MVS3 (server).

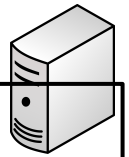


Error Example #2: Server Cannot Find its Certificate

Solution:

1. Connect FTP Server Certificate to Key Ring.
2. Change Policy Instance number and UPDATE PAGENTT to reinstall changes to the Key ring for FTP Server.
3. Recycle FTP Server to reinstall changes 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
```

IBM z/OS Encryption Readiness Technology (zERT) Network Analyzer

- A new plug-in for z/OSMF that network security admins can use to import, query and analyze data
- Collects security attributes of IP and Enterprise Extender traffic protected using TLS/SSL, SSH and IPSec protocols
- Discover more information about:
  - What traffic is being protected and what's not
  - How traffic is protected
  - Who on z/OS system is consuming traffic

Appendix

Perform the following steps if you run into problems:

1. Issue pasearch –t to see all AT-TLS policies that are active in Policy Agent.

2. Issue Netstat TTLS Conn *connid* or Netstat –x Conn *connid*. To determine whether the stack mapped a connection to AT-TLS policy and, if so, to which policy was mapped. Ensure that your AT-TLS policies are correctly defined.

3. In cases where AT-TLS connections do not map to any policy, verify that TCPCONFIG TTLS has been specified. Netstat configuration shows the current setting of AT-TLS.

4. If an error message was issued by AT-TLS, review the syslogd files for message EZD1286I or the TCP/IP job log for message EZD1287I.

Perform the following steps if you run into problems:

5. If error is recreatable, turn on an AT-TLS trace for the connection. Turn on the trace by coding a TTLSRule specific to the failing connection. Include a TTLSConnectionAction statement that has the Trace statement set to 255 (All).

6. If the problem cannot be resolved from the trace, perform a packet trace or a CTRACE with option TCP to provide addt'l debugging information and contact IBM service.

7. If System SSL tracing is needed, enable the GSKSRVR CTRACE with option Level=255. The JOBNAME specification needs to be the TCP/IP stack name. The GSK_TRACE and GSK_TRACE_FILE environment variables cannot be used to capture System SSL tracing when using AT-TLS.

Error messages – startup errors

- If message EZZ4248E is written to the console and not released, one of the following might have occurred:
  - Policy Agent has not been started.
  - Policy Agent configuration does not contain a TCPImage statement for this stack, or the stack policy configuration does not contain any local or remote AT-TLS policies.
  - Policy Agent is not permitted to create a socket with this stack. Ensure that the SERVAUTH class is active. Ensure that the EZB.INITSTACK.mvsname.tcpname resource profile is defined and that Policy Agent is permitted to it. If the EZB.STACKACCESS.mvsname.tcpname resource profile is defined, ensure that Policy Agent is permitted to it.
- If message EZD1288I is displayed, the AT-TLS group is configured with FIPS140 on, but ICSF is not active. Start ICSF before starting AT-TLS groups configured with FIPS140 On.

Error messages – startup errors

- If applications started after the stack fail to create a socket (errno EAGAIN, errno2 JrTcpNotActive), the stack is probably being configured for AT-TLS, and the application has been started before AT-TLS policy has been installed. If this is a required network infrastructure application, permit it to the EZB.INITSTACK.mvsname.tcpname resource profile in the SERVAUTH class. If it is not a required network infrastructure application, either start it after message EZZ4248E is released or modify the application to wait a short time and try again when the errno is EAGAIN.
- If message EZD1287I TTLS Error RC: 5020 Group Init is displayed, the TCP/IP stack was not able to load the System SSL DLL required for AT-TLS processing.

Sample AT-TLS Trace

Example trace of a generic server processing a secure connection.

Trace level 255 was used to generate this trace.

The standard syslogd prefix information has been removed from this trace.

```
11:10:25 TCPCS3   EZD1281I TTLS Map   CONNID: 00000025 LOCAL: 9.42.104.156..21 REMOTE: 9.27.154.171..1271
                           JOBNAME: FTPD2 USERID: FTPD TYPE: InBound STATUS: Enabled RULE: ftp_serv_21
                           ACTIONS: grp_act1 env_act_serv **N/A**  1
11:10:28 TCPCS3   EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000000 CONNID: 00000025  RC:    0
  Connection Init
11:10:28 TCPCS3   EZD1282I TTLS Start GRPID: 00000001 ENVID: 00000001 CONNID: 00000000 Environment Create
                           ACTIONS: grp_act1 env_act_serv **N/A**  2
11:10:28 TCPCS3   EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000002 CONNID: 00000000  RC:    0
  Environment Master
                           Create 00000001
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Call
                           GSK_ENVIRONMENT_OPEN - 7F1DB058
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_KEYRING_FILE -  FTPDsafkeyring   3
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_CLIENT_AUTH_TYPE -  FULL
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_SESSION_TYPE -  SERVER
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_PROTOCOL_SSLV2 -  ON
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_PROTOCOL_SSLV3 -  ON
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_PROTOCOL_TLSV1 -  ON
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_IO_CALLBACK -
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_SSL_HW_DETECT_MESSAGE -  1
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Call
                           GSK_ENVIRONMENT_INIT - 7F1DB058
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_SSL_HW_DETECT_MESSAGE -  NULL
11:10:28 TCPCS3   EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000002 CONNID: 00000000  RC:    0
  Environment Master
                           Init 7F1DB058
11:10:28 TCPCS3   EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000001 CONNID: 00000000  RC:    0
  Environment
                           Link 7F1DB058 00000002
11:10:28 TCPCS3   EZD1282I TTLS Start GRPID: 00000001 ENVID: 00000001 CONNID: 00000025 Initial Handshake
                           ACTIONS: grp_act1 env_act_serv **N/A** HS-Server    4
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000001 CONNID: 00000025  RC:    0 Call
                           GSK_SECURE_SOCKET_OPEN - 7F0CA118
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000001 CONNID: 00000025  RC:    0 Set
                           GSK_FD - 00000025
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000001 CONNID: 00000025  RC:    0 Set
                           GSK_USER_DATA - 7F1DB330
```

Sample AT-TLS Trace

1 TCP connection has mapped to an AT-TLS rule. The parameters used to search the AT-TLS rules are listed.

2 AT-TLS is creating an environment instance for the application.

3 AT-TLS is establishing the parameters for this environment, obtained from the TTLSEnvironmentAction statement. This trace message is defining the key ring to be used by this environment.

4 AT-TLS has successfully set up the secure environment and is now initializing the secure connection.

```
11:10:25 TCPCS3   EZD1281I TTLS Map   CONNID: 00000025 LOCAL: 9.42.104.156..21 REMOTE: 9.27.154.171..1271
                           JOBNAME: FTPD2 USERID: FTPD TYPE: InBound STATUS: Enabled RULE: ftp_serv_21
                           ACTIONS: grp_act1 env_act_serv **N/A**  1
11:10:28 TCPCS3   EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000000 CONNID: 00000025  RC:    0
  Connection Init
11:10:28 TCPCS3   EZD1282I TTLS Start GRPID: 00000001 ENVID: 00000001 CONNID: 00000000 Environment Create
                           ACTIONS: grp_act1 env_act_serv **N/A**  2
11:10:28 TCPCS3   EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000002 CONNID: 00000000  RC:    0
  Environment Master
                           Create 00000001
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Call
                           GSK_ENVIRONMENT_OPEN - 7F1DB058
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_KEYRING_FILE -  FTPDsafkeyring   3
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_CLIENT_AUTH_TYPE -  FULL
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_SESSION_TYPE -  SERVER
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_PROTOCOL_SSLV2 -  ON
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_PROTOCOL_SSLV3 -  ON
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_PROTOCOL_TLSV1 -  ON
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_IO_CALLBACK -
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_SSL_HW_DETECT_MESSAGE -  1
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Call
                           GSK_ENVIRONMENT_INIT - 7F1DB058
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000002 CONNID: 00000025  RC:    0 Set
                           GSK_SSL_HW_DETECT_MESSAGE -  NULL
11:10:28 TCPCS3   EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000002 CONNID: 00000000  RC:    0
  Environment Master
                           Init 7F1DB058
11:10:28 TCPCS3   EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000001 CONNID: 00000000  RC:    0
  Environment
                           Link 7F1DB058 00000002
11:10:28 TCPCS3   EZD1282I TTLS Start GRPID: 00000001 ENVID: 00000001 CONNID: 00000025 Initial Handshake
                           ACTIONS: grp_act1 env_act_serv **N/A** HS-Server    4
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000001 CONNID: 00000025  RC:    0 Call
                           GSK_SECURE_SOCKET_OPEN - 7F0CA118
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000001 CONNID: 00000025  RC:    0 Set
                           GSK_FD - 00000025
11:10:28 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000001 CONNID: 00000025  RC:    0 Set
                           GSK_USER_DATA - 7F1DB330
```

Sample AT-TLS Trace

5 Secure data has been received for this connection. During secure handshake, all the data is traced.

6 Secure data is being sent for this connection.

7 The secure handshake has been completed. The protocol negotiated and the cipher suite negotiated are displayed.

8 AT-TLS is sending a secure alert message, because the application closed the socket.

9 The secure connection is being closed.

```
11:10:28 TCPCS3   EZD1285I TTLS Data  CONNID: 00000025 RECV CIPHER 807A010301  5
11:10:28 TCPCS3   EZD1285I TTLS Data  CONNID: 00000025 RECV CIPHER
   005100000020000040100800000500002F000033000032
   00000A0700C000001600001300000906004000001500001200000302008000000800001400001100000100000
   020000180000340000 1B00001A00001700001941E69D75F7DCB55234895D884B271253A522E4BE211250F546
                      4FE5C5AB980FBD
11:10:28 TCPCS3   EZD1285I TTLS Data  CONNID: 00000025 SEND CIPHER
   160301029002000046030141E69D7534693728 57A71168D9
   9D7B93AD6CC30F6F6BDF7F774929CD4D2E8E2A2000000026091B9AAB04F7000000000000000000000000000000
   41E69D7500000001000500 0B00023E00023B0002383082023430820 19DA003020102020100300D06092A8648
   86F70D0101050500302E310B3009060355040613027573311030 0E060355040B130774657374696E67310D30
   0B060355040313044654504 4301E170D303430383039303430305A170D3035303831303   6
11:10:28 TCPCS3   EZD1285I TTLS Data  CONNID: 00000025 RECV CIPHER 1603010086
11:10:28 TCPCS3   EZD1285I TTLS Data  CONNID: 00000025 RECV CIPHER
   10000082008037A6573A4C160A8C0810C542A1CEB73A9FF5
   899D767711EF3BF86D4C2D2743837AA4D5E247DE35F79C8A71A9E6A18DF8CC845D5E0F8F386DF84D746A4004
   B641C14DD7A002FAC5538ED52E3194C2ADE6010381BFC70D1CA6D9F34EDC0F345F0A015575A6C9D85602B1BF
   2877760BA91FC6296625A16A274426112C65DB7A2685
11:10:29 TCPCS3   EZD1285I TTLS Data  CONNID: 00000025 RECV CIPHER 1403010001
11:10:29 TCPCS3   EZD1285I TTLS Data  CONNID: 00000025 RECV CIPHER 01
11:10:29 TCPCS3   EZD1285I TTLS Data  CONNID: 00000025 RECV CIPHER 1603010024
11:10:29 TCPCS3   EZD1285I TTLS Data  CONNID: 00000025 RECV CIPHER
   789DBBACAE9D6F19F62B1AF2B529B1850F7057A6EDDE64CD 2301D91CA43C4EBBB5A3DFE5
11:10:29 TCPCS3   EZD1285I TTLS Data  CONNID: 00000025 SEND CIPHER 140301000101
11:10:29 TCPCS3   EZD1285I TTLS Data  CONNID: 00000025 SEND CIPHER
   603010024FE6548CCBA0D820D73FF439A6B475B4116BCE4
   6FF225DAE1A0F7EC2AEA4690595E63F036
11:10:29 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000001 CONNID: 00000025  RC:    0 Call
                           GSK_SECURE_SOCKET_INIT - 7F0CA118
11:10:29 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000001 CONNID: 00000025  RC:    0 Get
                           GSK_CONNECT_SEC_TYPE -  TLSV1
11:10:29 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000001 CONNID: 00000025  RC:    0 Get
                           GSK_CONNECT_CIPHER_SPEC -  05
11:10:29 TCPCS3   EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000001 CONNID: 00000025  RC:    0
   Initial Handshake
                           7F0CA118 7F1DB058 TLSV1 05  7
11:11:05 TCPCS3   EZD1285I TTLS Data  CONNID: 00000025 SEND CIPHER
   1503010016D47A7AEC70D317976ACEEF3418CDCC8B2DF7
   D3491D      8
11:11:13 TCPCS3   EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000001 CONNID: 00000025  RC: 0 Receive
   Reset
11:11:13 TCPCS3   EZD1282I TTLS Start GRPID: 00000001 ENVID: 00000001 CONNID: 00000025 Connection Close
                           ACTIONS: grp_act1 env_act_serv **N/A**    9
11:11:13 TCPCS3   EZD1284I TTLS Flow  GRPID: 00000001 ENVID: 00000001 CONNID: 00000025  RC: 0 Call
                           GSK_SECURE_SOCKET_CLOSE - 7F0CA118
11:11:13 TCPCS3   EZD1283I TTLS Event GRPID: 00000001 ENVID: 00000001 CONNID: 00000025  RC: 0 Connection
                           Close 7F0CA118 7F1DB058
```