Integrating the Healthcare Enterprise

IHE Secure Node Tests

Electronic Radiology Laboratory
Mallinckrodt Institute of Radiology
510 South Kingshighway Blvd.
St. Louis, MO 63110
314.362.6965 (Voice)
314.362.6971 (Fax)

Revision 9.8.0
5-Dec-2005

1 Secure Node Tests

IHE Secure Nodes combine one or more IHE actors with secure communications. Secure communications implies the following:

- The Secure Node uses and requires authentication of network operations using TLS
- The Secure Node sends audit records to the Audit Record Repository

For the purposes of this document, we will classify nodes as clients or servers. A client is one that initiates a network connection; a server listens for and accepts a network connection. Many systems may operate as both a client and as a server. This term is not used in the IHE Technical Framework.

This document will describe tests for both client and server applications.

1.1 Introduction

Each test is run using the same procedure. We assume you are using an interactive terminal or terminal emulator and are logged on to the MESA test system. Change direct1ory to *$MESA_TARGET/mesa_tests/rad/actors/secure_node*. Make sure the $MESA_TARGET and $MESA_STORAGE environment variables are set properly.

1.1.1 Integration Profiles and Test Procedures

This document lists a number of tests for Secure Node Systems. You may not be responsible for all of these tests.

Please refer to the Connectathon web tool to list the required tests for your system. The web address of this tool depends on the year and project manager. Please contact the appropriate project manager to obtain this information.

1.2 Message Attributes

This section is applicable for other actors and other tests. Expect that all fields of X.509 certificates and IHE Audit (syslog) messages are subject to evaluation.

1.3 Message Values

This section is applicable for other actors and other tests.

1.4 Configuration

The Secure Node scripts described below use an ASCII configuration file to identify parameters such as host names and port numbers. The configuration file is named *secure_test.cfg* and is included in the directory *$MESA_TARGET/mesa_tests/rad/actors/secure_node*. Edit the file and

change entries (host name, port number) that pertain to your system. Your system is identified by entries that begin with TEST.

The table below gives parameters for MESA servers that will receive messages from your system.

Application	Port Number
MESA Syslog server	4000
MESA TLS Server – configured to exchange proper certificates	4100
MESA TLS Server – configured to respond with an unregistered certificate	4101
MESA TLS Server – configured to respond with an expired certificate	4102

Read the Runtime Notes section of the *Installation Guide* to determine the proper settings for the MESA runtime environment.

1.5 Starting the MESA Servers

These instructions assume you are using a terminal emulator on Unix systems or an MS DOS command window under Windows NT. Each test uses a command line interface; there is no graphical user interface. Before you start the test procedure, you need to start several MESA servers. Make sure the appropriate database is running (PostgreSQL, SQL Server). To start the MESA servers:

1. Enter the Secure Node exam directory: *mesa_tests/rad/actors/secure_node*
2. Execute the appropriate script to start the servers:

   `scripts/start_mesa_servers.csh [loglevel]` (Unix)

   `scripts\start_mesa_servers.bat` (Windows)

In the unix environment, loglevel is an optional argument (0 is default). Log levels are:

0 no logging
1 errors
2 warnings
3 verbose
4 conversational (really verbose)

To set the log level for Windows NT, set the environment variable LOGLEVEL to the desired value before invoking the batch file.

When you are finished running one or more tests, you can stop the servers:

`scripts/stop_mesa_servers.csh` (Unix)

`scripts\stop_mesa_servers.bat` (Windows)

Log files are stored in $MESA_TARGET/logs.

1.6 External Audit Record Repositories

The MESA tools are shipped with an Audit Record Repository that supports the BSD Syslog protocol (UDP). Reliable Syslog is handled using products from different systems. This release of the software relies on a Knoppix CD made available by HIPAAT, Inc. To send audit messages using the Reliable Syslog protocol, you will need to download the ISO image of the Knoppix CD or request a CD from the Project Manager.

There is a separate document from HIPAAT that describes how to start/run the CD. The CD is shipped assuming you will use DHCP to obtain an IP address. You can modify the network setup to give the PC a fixed IP address.

1.7 Digital Certificates

All digital certificates for testing are located in the directory $MESA_TARGET/runtime/certificates. Included in the directory are pairs of files for the private key and public certificate. This is not a secure way to distribute these, but the goal is to work on the technology of certificates.

Your system should use the private key and certificate found in the files starting with test_sys_1. Import these into your system using whatever configuration is necessary.

The MESA client and server applications will use certificates found in the file mesa_list.cert. Use this as the list of all certificates that MESA may use when communicating with your system.

Do not use your own certificates for these tests or try to configure the MESA tools with different certificates. If there are issues with the certificates, then please log a bug report.

2 Basic Secure Node (Client or Server)

Each section below describes one test that is appropriate for a Secure Node in the Basic Security Integration Profile that is configured as either a client node or a server node. Later sections will list tests that are specific to client operations or server operations.

2.1 Test Instructions: 120x Tests

Each test is independent of the others. You must collect the results of one test before starting a new test. You do not have to run the tests in the order listed. Each of the tests in the 120x section are designed for the IHE Radiology Basic Security Profile. This uses the provisional schema and UDP messaging to the MESA Audit Record Repository.

1. Enter the Secure Node directory: *mesa_tests/rad/actors/secure_node.*
2. Remember the MESA servers were started according to the directions in section 1.5.
3. Enter a specific test directory (1201, 1202, …). Follow the test instructions for each test found in the next sections of this document.

2.2 Test Case 1200: SEC List Audit Messages

This is a documentation procedure where you list all of the audit messages that your system is required to produce. The purpose of the test is for you to provide that list to the Project Manager so that the manager can determine if your system is producing the proper set of messages. This test result is due 3 weeks in advance of the normal test deadlines. This will give you time to recover in the event that you are missing audit events required of your system.

1. Create a text file named: grade_1200.txt. Content of the file should be as listed below.
    a. Line 1: Company Name
    b. Line 2: System Name
    c. Line 3 and following: List of all Integration Profile/Actor combinations for this system.
    d. Following lines: List all audit events for which your system produces an audit message.
2. Submit the text file to the Project Manager for evaluation.

2.3 Basic Secure Node Test 1201: Actor Start

This sequence tests your ability to send an audit record to the MESA Audit Record Repository. This test covers the basic functionality of transmitting the message and the proper XML format of the message. The Actor Start message is chosen as that is required of all actors and is

independent of other IHE transactions. This can be run using the IETF or INTERIM audit record format.

1. If not already done, start the MESA servers according to the directions in section 1.5.
2. By whatever means you use, "start" your actor such that it generates the actor-start-stop audit record message. This should be sent to the MESA Audit Record Repository.
3. Run the evaluation script to examine the last audit record sent by your system to the MESA Audit Record Repository. It should find the last audit record was for Actor-start-stop and it should verify the content of that record against the IHE XML schema:

   ```
   perl 1201/eval_client_1201.pl <log> <INTERIM or IETF>
   ```

   <log> is a log level (1 – 4). The results file (*1201/grade_client_1201.txt*) should show 0 failures.
4. Run the evaluation at log level 4 and submit the test results to the Project Manager.

2.4 Basic Secure Node Test 1202: System Configuration

This sequence tests your ability to send an audit record to the MESA Audit Record Repository. This test covers the basic functionality of transmitting the message and the proper XML format of the message. The Actor-config message is chosen.

1. If not already done, start the MESA servers according to the directions in section 1.5.
2. By whatever means you use, configure or reconfigure your actor such that it generates the IHE Actor-config audit record message. This should be sent to the MESA Audit Record Repository. For example, you might change the host name.
3. Run the evaluation script to examine the last audit record sent by your system to the MESA Audit Record Repository. It should find the last audit record was for Actor-config and it should verify the content of that record against the IHE XML schema:

   ```
   perl 1202/eval_client_1202.pl
   ```

   The results file (*1202/grade_client_1202.txt*) should show 0 failures.
4. Submit the test results to the Project Manager.

2.5 Basic Secure Node Test 1203: User Authenticated

This sequence tests your ability to send an audit record to the MESA Audit Record Repository. This test covers the basic functionality of transmitting the message and the proper XML format of the message. The User-Authenticated message is chosen.

1. If not already done, start the MESA servers according to the directions in section 1.5.

2. By whatever means you use, authenticate (login) a user with your system. This should generate an Audit message (User-Authenticated) that should be sent to the MESA Audit Record Repository.
3. Run the evaluation script to examine the last audit record sent by your system to the MESA Audit Record Repository. It should find the last audit record was for User-Authentication and it should verify the content of that record against the IHE XML schema:

   ```
   perl 1203/eval_client_1203.pl
   ```

   The results file (*1203/grade_client_1203.txt*) should show 0 failures.
4. Submit the test results to the Project Manager.

2.6 Basic Secure Node Test 1205: Unspecified Records

Tests 1201 through 1204 cover specific events that defined by the MESA documentation. For test 1205, the system under test is asked to send three or more Audit Record messages to the MESA Audit Record Repository. These messages are evaluated by the MESA software, and the user collects the messages and sends them to the Project Manager for distribution to other systems.

You are welcome to use the events that are specified for tests 1201 through 1204. You might want to use other events so that your software is more fully testetd.

1. If not already done, start the MESA servers according to the directions in section 1.5.
2. Clear the MESA Audit Record Repository of existing messages:

   ```
   perl scripts/clear_db.pl
   ```
3. Generate three (3) or more Audit Record messages and send these to the MESA Audit Record Repository.
4. Run the evaluation script to examine all audit records sent during this test:

   ```
   perl 1205/eval_client_1205.pl
   ```
5. Collect all of the files (tar/zip) in *$MESA_TARGET/logs/syslog* and submit these to the Project Manager.

2.7 Basic Secure Node Test 1211: Time Synchronization

Time synchronization requires an external system that serves as an NTP server. The MESA tools do not include such a server, but they are readily available.

If time permits, the Project Manager will load the time server software and allow access for participants. As of this version of the document, that is not available.

1. Read about NTP at the site http://www.ntp.org
2. Select/locate a public NTP server. Follow any rules of etiquette posted for that server.
3. Configure your system to synchronize time with that public NTP server.
4. At exactly 13:00 local time, 7 days before the MESA test results are due, send an email to the Project Manager (just kidding).

We do not require proof that you have performed this test.

3 ATNA Secure Node (Client or Server)

Each section below describes one test that is appropriate for a Secure Node in the ATNA Integration Profile that is configured as either a client node or a server node.

3.1 11100: ATNA: List Audit Messages

This is a documentation procedure where you list all of the audit messages that your system is required to produce. The purpose of the test is for you to provide that list to the Project Manager so that the manager can determine if your system is producing the proper set of messages. This test result is due 3 weeks in advance of the normal test deadlines. This will give you time to recover in the event that you are missing audit events required of your system.

3. Create a text file named: grade_11120.txt. Content of the file should be as listed below.
    a. Line 1: Company Name
    b. Line 2: System Name
    c. Line 3 and following: List of all Integration Profile/Actor combinations for this system.
    d. Following lines: List all audit events for which your system produces an audit message.
4. Submit the text file to the Project Manager for evaluation.

3.2 11101: ATNA Audit: Actor Start BSD

This sequence tests your ability to send an audit record to a “BSD” Audit Record Repository. This test covers the basic functionality of transmitting the message and the proper XML format of the message. The Actor Start message is chosen as that is required of all actors and is independent of other IHE transactions.

1. If not already done, start the MESA servers according to the directions in section 1.5.
2. By whatever means you use, “start” your actor such that it generates the Start/Stop audit record message. Send this message to the MESA “BSD” Audit Record Repository.
3. Run the evaluation script to examine the last audit record sent by your system to the MESA Audit Record Repository. It should find the last audit record was for Actor-start-stop and it should verify the content of that record against the IHE XML schema:

```
perl 11101/eval_11101.pl <log level> INTERIM (or)
perl 11101/eval_11101.pl <log level> IETF
```

   where INTERIM or IETF indicate which schema is to be used.

   The results file (*11101/grade_11101.txt*) should show 0 failures.

4. Submit the test results to the Project Manager run at log level 4.

3.3 11102: ATNA Audit: Actor Start Reliable Syslog

This sequence tests your ability to send an audit record to a Reliable Syslog Audit Record Repository. This test covers the basic functionality of transmitting the message and the proper XML format of the message. The Actor Start message is chosen as that is required of all actors and is independent of other IHE transactions.

1. Start the Reliable Syslog Audit Record Repository (separate instructions). There are 3 possible servers to use:
    a. HIPAAT Knoppix CD run in your laboratory
    b. SDSC implementation run in your laboratory
    c. Server maintained on the Internet by the Project Manager.
2. By whatever means you use, "start" your actor such that it generates the Start/Stop audit record message. Send this message to the MESA "BSD" Audit Record Repository.
3. Extract the log message from the server.
    a. For the HIPAAT Knoppix CD, use a web browser to connect to the system. Use the address of the server with no further designation (for example):

        ```
        http://192.168.1.10
        ```

        Use the web browser to search through the log messages; copy your log message and paste into a file for evaluation below.
    b. SDSC: not documented in this release.
4. Evaluate the log message using the command below:

    ```
    perl 11102/eval_11102.pl <log level> INTERIM FILE(or)
    perl 11102/eval_11102.pl <log level> IETF FILE
    ```

    where INTERIM or IETF indicate which schema is to be used and FILE is the name of the extracted file.

    The results file (*11102/grade_11102.txt*) should show 0 failures.
5. Submit the test results to the Project Manager run at log level 4.

3.4 11103: ATNA Audit: Actor Specific Audit Message

In this test, the actor generates a log message that is specific to the actor. The Actor Start/Stop or User Authentication messages are general in nature.

1. Start the Reliable Syslog Audit Record Repository (separate instructions) or the MESA BSD Audit Record Repository as appropriate.
2. By whatever means you use, generate an audit record message that is specific to one or more of the actors in your system. Send this message to the appropriate repository.
3. Extract the log message from the server. Instructions depend on the server. If you send to the MESA BSD Audit Record Repository, the file is found in $MESA_STORAGE/logs/syslog/last_log.xml.
4. Evaluate the log message using the command below:

   ```
   perl 11103/eval_11103.pl <log level> INTERIM FILE(or)
   perl 11103/eval_11103.pl <log level> IETF FILE
   ```

   where INTERIM or IETF indicate which schema is to be used and FILE is the name of the extracted file.

   The results file (*11103/grade_11103.txt*) should show 0 failures.
5. Submit the test results to the Project Manager run at log level 4.

3.5 11104: ATNA Audit: User Authentication

In this test, the system generates a User Authentication log message and evaluates it using the MESA tools.

1. Start the Reliable Syslog Audit Record Repository (separate instructions) or the MESA BSD Audit Record Repository as appropriate.
2. Generate an audit record message for a User Authentication event. Send this message to the appropriate repository.
3. Extract the log message from the server. Instructions depend on the server. If you send to the MESA BSD Audit Record Repository, the file is found in $MESA_STORAGE/logs/syslog/last_log.xml.
4. Evaluate the log message using the command below:

   ```
   perl 11104/eval_11104.pl <log level> INTERIM FILE(or)
   perl 11104/eval_11104.pl <log level> IETF FILE
   ```

   where INTERIM or IETF indicate which schema is to be used and FILE is the name of the extracted file.

   The results file (*11104/grade_11104.txt*) should show 0 failures.

5. Submit the test results to the Project Manager run at log level 4.

3.6 11121: ATNA Audit: Patient Records

For test 11121, the system under test is asked to generate three or more audit messages. The user collects the messages and sends them to the Project Manager for distribution to other systems.

1. Generate three (3) or more Audit Record messages containing at least one record for:
    a. User Authentication
    b. Patient Record Access

    If the Patient Record Access is not pertinet, substitute a different event (PHI export). The third record is of your choosing.
2. Place each message in a separate XML file and tar/zip the collection together. Name the tar/zip file using the system name found in the Kudu web tool.
3. Submit the tar/zip file to the Project Manager. The Project Manager will distribute to other vendors for testing.
4. Please submt the records 2 weeks in advance of the normal deadline to allow distribution to other systems.

4 Secure Client Node Tests

Each section below lists one test for a Secure Client Node. The test authors define a Secure Node Client as a client system in the traditional client/server model where the client initiates a network connection. These tests in this section assume the Secure Node is initiating such a connection.

4.1 Secure Client Node Test 1221: Certificate Exchange with Valid Certificate

In this test, your client application requests a network connection with a MESA server using the standard X.509 (unexpired) certificates. The MESA server will complete the TLS handshake, read data from the socket and then disconnect after 2 seconds. Therefore, this test demonstrates your ability to implement the basic TLS handshake with the cyphersuite:

TLS_RSA_WITH_NULL_SHA

1. Configure your system using the X.509 certificate assigned to you ($MESA_TARGET/runtime/certificates/test_sys_1). The MESA servers are configured to recognize this certificate.
2. If not already done, start the MESA servers according to the directions in section 1.5.
3. (Optional) Clear the log files of prior messages:

   ```
   perl scripts/clear_logs.pl
   ```

4. Open a network connection with the MESA TLS server listening at port 4100. We assume this means you have to initiate an IHE transaction; this transaction will fail because the MESA TLS server does not respond beyond the TLS handshake.
5. After the TLS handshake and aborted message sequence, examine the MESA log file in *$MESA_TARGET/logs/tls_server.txt*. This should indicate a successful TLS handshake.
6. Cut/paste the entry from the MESA log file. Create a file named SYSTEM_1221.log, enter the log information and submit that file to the Project Manager.

4.2 Secure Client Node Test 1222: Certificate Exchange with Unregistered Certificate

In this test, your client application requests a network connection with a MESA server using the standard X.509 certificates. The MESA server will attempt the TLS handshake by offering an unregistered certificate. You are expected to abort the network connection and log an audit record message with the MESA Audit Record Repository (at port 4000 on the MESA system).

1. Configure your system using the X.509 certificate assigned to you ($MESA_TARGET/runtime/certificates/test_sys_1). The MESA servers are configured to recognize this certificate.
2. If not already done, start the MESA servers according to the directions in section 1.5.
3. (Optional) Clear the log files of prior messages:

   ```
   perl scripts/clear_logs.pl
   ```

4. Open a network connection with the MESA TLS server listening at port 4101. We assume this means you have to initiate an IHE transaction; the MESA server will attempt the TLS handshake with an unregistered certificate.
5. After the TLS handshake and aborted message sequence, examine the MESA log file in *$MESA_TARGET/logs/tls_server.txt*. This should indicate a connection attempt from your system and an aborted connection.
6. Run the evaluation script to examine the last audit record sent by your system to the MESA Audit Record Repository. It should find the last audit record was for Node-Authentication failure and it should verify the content of that record against the IHE XML schema:

   ```
   perl 1222/eval_client_1222.pl IETF (or)
   perl 1222/eval_client_1222.pl INTERIM
   ```

   The results file (*1222/grade_client_1222.txt*) should show 0 failures.

   This test assumes the BSD syslog mechanism. If your system does not send the audit message using that protocol, extract the log message and evaluate as follows:

   ```
   perl 1222/eval_file.pl <log> <IETF or INTERIM> FILE
   ```

7. Submit the grade file to the Project Manager.

4.3 Secure Client Client Test 1223: Certificate Exchange with Expired Certificate

In this test, your client application requests a network connection with a MESA server using the standard X.509 certificates. The MESA server will attempt the TLS handshake by offering an expired certificate. You are expected to abort the network connection and log an audit record message with the MESA Audit Record Repository (at port 4000 on the MESA system).

1. Configure your system using the X.509 certificate assigned to you ($MESA_TARGET/runtime/certificates/test_sys_1). The MESA servers are configured to recognize this certificate.
2. If not already done, start the MESA servers according to the directions in section 1.5.
3. (Optional) Clear the log files of prior messages:

   ```
   perl scripts/clear_logs.pl
   ```

4. Open a network connection with the MESA TLS server listening at port 4102. We assume this means you have to initiate an IHE transaction; the MESA server will attempt the TLS handshake with an expired certificate.
5. After the TLS handshake and aborted message sequence, examine the MESA log file in *$MESA_TARGET/logs/tls_server.txt*. This should indicate a connection attempt from your system and an aborted connection.
6. Run the evaluation script to examine the last audit record sent by your system to the MESA Audit Record Repository. It should find the last audit record was for Node-Authentication failure and it should verify the content of that record against the IHE XML schema:

   ```
   perl 1223/eval_client_1223.pl INTERIN (or)
   perl 1223/eval_client_1223.pl IETF
   ```

   The results file (*1223/grade_client_1223.txt*) should show 0 failures.

   This test assumes the BSD syslog mechanism. If your system does not send the audit message using that protocol, extract the log message and evaluate as follows:

   ```
   perl 1223/eval_file.pl <log> <IETF or INTERIM> FILE
   ```

8. Submit the grade file to the Project Manager.

4.4 Secure Client Node Test 1224: TLS Handshake for 3DES

This test is not available with the current MESA software.

In this test, your client application requests a network connection with a MESA server using the standard X.509 (unexpired) certificates. The MESA server will only support 3DES encryption. The MESA server will complete the TLS handshake, read data from the socket and then disconnect after 2 seconds. Therefore, this test demonstrates your ability to implement the basic TLS handshake with the cyphersuite:

TLS_RSA_WITH_3DES_SHA

1. Configure your system using the X.509 certificate assigned to you. Make certain the MESA servers are configured to recognize this certificate.
2. If not already done, start the MESA servers according to the directions in section 1.5.
3. Open a network connection with the MESA TLS server listening at port 4103. We assume this means you have to initiate an IHE transaction; this transaction will fail because the MESA TLS server does not respond beyond the TLS handshake.
4. After the TLS handshake and aborted message sequence, examine the MESA log file in *$MESA_TARGET/logs/tls_server.txt*. This should indicate a successful TLS handshake.

4.5 Secure Client Node Test 1226: DICOM Verification with TLS

This test is for DICOM client applications that are lacking a fully integrated MESA test sending data with TLS. In this test, your actor establishes a TLS connection with a MESA server and sends a DICOM C-Echo command (Verification class). If your actor has a fully integrated MESA test that exercises TLS and DICOM, you can skip this test.

1. Configure your system using the X.509 certificate assigned to you ($MESA_TARGET/runtime/certificates/test_sys_1). The MESA servers are configured to recognize this certificate.
2. If not already done, start the MESA servers according to the directions in section 1.5.
3. (Optional) Clear the log files of prior messages:

   ```
   perl scripts/clear_logs.pl
   ```

4. Establish a DICOM/TLS connection with the MESA server running on port 2350. DICOM AE titles are ignored. Send a C-Echo request to that server.

5. This should run to completion with no errors. If you encounter an error, you will need to correct the communication problem and rerun the test.
6. When you have successfully completed the C-Echo request, there will be log information stored in the MESA Image Manager log: $MESA_TARGET/logs/imgmgr.log. Submit that log file to the Project Manager as the output of this test.

Troubleshooting:

1. The most typical problem is using the wrong certificate. Start the MESA servers with the highest log level. If you cannot get a C-Echo command to work, examine the MESA Image Manager log file.
2. Make sure you are using the Certificate designed for test systems and you have a copy of the MESA certificate for your own configuration.

4.6 Secure Client Node Test 1227: DICOM Verfication: TLS: Unregistered Certificate

In this test, your actor establishes a TLS connection with a MESA server that has a certificate that is not registered with your system. That is, your system should attempt to establish a DICOM connection, determine that the MESA system is using a certificate that is not known to you, and abort/terminate the network connection.

1. Configure your system using the X.509 certificate assigned to you ($MESA_TARGET/runtime/certificates/test_sys_1). The MESA servers are configured to recognize this certificate. (That step would have been completed for test 1226).
2. If not already done, start the MESA servers according to the directions in section 1.5.
3. (Optional) Clear the log files of prior messages:

```
perl scripts/clear_logs.pl
```

4. Attempt to establish a DICOM/TLS connection with the MESA server running on port **2351**. DICOM AE titles are ignored.
5. Your system should not complete the DICOM association negotiation.
6. When you have successfully completed the test, there will be log information stored in the MESA Image Manager log: $MESA_TARGET/logs/imgmgr.log. Submit that file to the Project Manager.

Notes:

1. This is a rather difficult test as it is designed to make something fail on purpose. Whether your system closes the connection gracefully or merely exits depends on your design and software.

4.7 Secure Client Node Test 1227: DICOM Verfication: TLS: Expired Certificate

In this test, your actor establishes a TLS connection with a MESA server that has a certificate that has expired. That is, your system should attempt to establish a DICOM connection, determine that the MESA system is using a certificate that is expired, and abort/terminate the network connection.

1. Configure your system using the X.509 certificate assigned to you ($MESA_TARGET/runtime/certificates/test_sys_1). The MESA servers are configured to recognize this certificate. (That step would have been completed for test 1226).
2. If not already done, start the MESA servers according to the directions in section 1.5.
3. (Optional) Clear the log files of prior messages:

```
perl scripts/clear_logs.pl
```

4. Attempt to establish a DICOM/TLS connection with the MESA server running on port **2352**. DICOM AE titles are ignored.
5. Your system should not complete the DICOM association negotiation.
6. When you have successfully completed the test, there will be log information stored in the MESA Image Manager log: $MESA_TARGET/logs/imgmgr.log. Submit that file to the Project Manager.

Notes:

1. The certificate used by the Image Manager for this test is located in *$MESA_TARGET/runtime/certificates/expired.cert.* You should try to configure your system to know that this is the peer certificate. This is to make sure you are testing for expired certificates rather than unregistered certificates.

5 Secure Server Node Tests

Each section below lists one test for a Secure Server Node.

5.1 Secure Server Node Test 1221: Certificate Exchange with Valid Certificate

In this test, a MESA client application requests a network connection with your server using the standard X.509 (unexpired) certificates. The MESA client will complete the TLS handshake and then disconnect after 2 seconds. Therefore, this test demonstrates your ability to implement the basic TLS handshake with the cyphersuite:

TLS_RSA_WITH_NULL_SHA

1. Configure your system using the X.509 certificate assigned to you ($MESA_TARGET/runtime/certificates/test_sys_1). The MESA servers are configured to recognize this certificate.
2. Make sure the configuration file *secure_test.cfg* accurately describes your system.
3. If not already done, start the MESA servers according to the directions in section 1.5.
4. (Optional) Clear the log files of prior messages:

   ```
   perl scripts/clear_logs.pl
   ```

5. Instruct the MESA client to open a connection with your server:

   ```
   perl 1221/1221_server_test.pl
   ```

6. After the TLS handshake and aborted message sequence, examine the MESA log file in *$MESA_TARGET/logs/tls_client.txt*. This should indicate a successful TLS handshake.

5.2 Secure Server Node Test 1222: Certificate Exchange with Unregistered Certificate

In this test, a MESA client application requests a network connection with your server using the standard X.509 certificates. The MESA server will attempt the TLS handshake by offering an unregistered certificate. You are expected to abort the network connection and log an audit record message with the MESA Audit Record Repository (at port 4000 on the MESA system).

1. Configure your system using the X.509 certificate assigned to you ($MESA_TARGET/runtime/certificates/test_sys_1). The MESA servers are configured to recognize this certificate.
2. Make sure the configuration file *secure_test.cfg* accurately describes your system.
3. If not already done, start the MESA servers according to the directions in section 1.5.
4. (Optional) Clear the log files of prior messages:

   ```
   perl scripts/clear_logs.pl
   ```

5. Instruct the MESA client to open a connection with your server:

   ```
   perl 1222/1222_server_test.pl
   ```

6. After the TLS handshake and aborted message sequence, examine the MESA log file in *$MESA_TARGET/logs/tls_client.txt*. This should indicate the aborted network connection.
7. Run the evaluation script to examine the last audit record sent by your system to the MESA Audit Record Repository. It should find the last audit record was for Node-Authentication failure and it should verify the content of that record against the IHE XML schema:

   ```
   perl 1222/eval_server_1222.pl
   ```

   The results file (*1222/grade_server_1222.txt*) should show 0 failures.

5.3 Secure Server Node Test 1223: Certificate Exchange with Expired Certificate

In this test, a MESA client application requests a network connection with your server using the standard X.509 certificates. The MESA server will attempt the TLS handshake by offering an expired certificate. You are expected to abort the network connection and log an audit record message with the MESA Audit Record Repository (at port 4000 on the MESA system).

1. Configure your system using the X.509 certificate assigned to you ($MESA_TARGET/runtime/certificates/test_sys_1). The MESA servers are configured to recognize this certificate.
2. Make sure the configuration file *secure_test.cfg* accurately describes your system.
3. If not already done, start the MESA servers according to the directions in section 1.5
4. (Optional) Clear the log files of prior messages:

   ```
   perl scripts/clear_logs.pl
   ```

5. Instruct the MESA client to open a connection with your server:

   ```
   perl 1223/1223_server_test.pl
   ```

6. After the TLS handshake and aborted message sequence, examine the MESA log file in *$MESA_TARGET/logs/tls_client.txt*. This should indicate the aborted network connection.
7. Run the evaluation script to examine the last audit record sent by your system to the MESA Audit Record Repository. It should find the last audit record was for Node-Authentication failure and it should verify the content of that record against the IHE XML schema:

   ```
   perl 1223/1223_eval_server.pl
   ```

   The results file (*1223/grade_server_1223.txt*) should show 0 failures.

5.4 Secure Server Node Test 1224: TLS Handshake for 3DES

This test is not available with the current MESA software. It will be added at a later date.

In this test, a MESA client application requests a network connection with your server using the standard X.509 (unexpired) certificates. The MESA client will offer 3DES encryption. The MESA client will complete the TLS handshake and then disconnect after 2 seconds. Therefore, this test demonstrates your ability to implement the basic TLS handshake with the cyphersuite: TLS_RSA_WITH_3DES_SHA

1. Configure your system using the X.509 certificate assigned to you ($MESA_TARGET/runtime/certificates/test_sys_1). The MESA servers are configured to recognize this certificate.Make sure the configuration file *secure_test.cfg* accurately describes your system.
2. If not already done, start the MESA servers according to the directions in section 1.5.
3. (Optional) Clear the log files of prior messages:

   ```
   perl scripts/clear_logs.pl
   ```

4. Instruct the MESA client to open a connection with your server:

   ```
   perl 1224/1224_server_test.pl
   ```

5. After the TLS handshake, examine the MESA log file in *$MESA_TARGET/logs/tls_client.txt*. This should indicate a successful TLS handshake.

5.5 Secure Server Node Test 1226: DICOM Verification with TLS

This test establishes a TLS connection with your server and sends a DICOM C-Echo command (Verification class) to your system.

1. Configure your system using the X.509 certificate assigned to you ($MESA_TARGET/runtime/certificates/test_sys_1). The MESA servers are configured to recognize this certificate.
2. Make sure the configuration file *secure_test.cfg* accurately describes your system.
3. If not already done, start the MESA servers according to the directions in section 1.5.
4. (Optional) Clear the log files of prior messages:

   ```
   perl scripts/clear_logs.pl
   ```

5. Instruct the MESA client to open a connection with your server:

   ```
   perl 1226/1226_server_test.pl
   ```

6. This should run to completion with no errors. If you encounter an error, you will need to correct the communication problem and rerun the test.

7. When the test is working successfully, run the test and redirect the output to a file. Submit that file to the Project Manager for evaluation.

5.6 Secure Server Node Test 1227: DICOM Verification: TLS: Unregistered Certificate

This test attempts to establish a TLS connection with your server using an unregistered certificate. Should you accept the connection, the MESA application sends a DICOM C-Echo command (Verification class) to your system. The proper behavior is that your system refuses the TLS connection.

1. Configure your system using the X.509 certificate assigned to you ($MESA_TARGET/runtime/certificates/test_sys_1). The MESA servers are configured to recognize this certificate.
2. Make sure the configuration file *secure_test.cfg* accurately describes your system.
3. If not already done, start the MESA servers according to the directions in section 1.5.
4. (Optional) Clear the log files of prior messages:

```
perl scripts/clear_logs.pl
```

5. Instruct the MESA client to open a connection with your server:

```
perl 1227/1227_server_test.pl
```

6. This should run to completion and indicate that a connection was not completed. If the script completes a DICOM verification this indicates an error that should be corrected.
7. When the test is working properly, run the test and redirect the output to a file. Submit that file to the Project Manager for evaluation.

5.7 Secure Server Node Test 1228: DICOM Verification: TLS: Expired Certificate

This test attempts to establish a TLS connection with your server using an expired certificate. Should you accept the connection, the MESA application sends a DICOM C-Echo command (Verification class) to your system. The proper behavior is that your system refuses the TLS connection.

1. Configure your system using the X.509 certificate assigned to you ($MESA_TARGET/runtime/certificates/test_sys_1). The MESA servers are configured to recognize this certificate.Make sure the configuration file *secure_test.cfg* accurately describes your system.
2. If not already done, start the MESA servers according to the directions in section 1.5.
3. (Optional) Clear the log files of prior messages:

```
perl scripts/clear_logs.pl
```

4. Instruct the MESA client to open a connection with your server:

```
perl 1227/1227_server_test.pl
```

5. This should run to completion and indicate that a connection was not completed. If the script completes a DICOM verification this indicates an error that should be corrected.
6. When the test is working properly, run the test and redirect the output to a file. Submit that file to the Project Manager for evaluation.

Notes:

1. The certificate used by the Image Manager for this test is located in *$MESA_TARGET/runtime/certificates/expired.cert.* You should try to configure your system to know that this is the peer certificate. This is to make sure you are testing for expired certificates rather than unregistered certificates.

6 ATNA Tests for Client Applications

The tests in this section are for ATNA applications that initiate TLS connections. In that sense, these are considered client applications.

6.1 Test 11141: ATNA Certificate Exchange with Valid Certificate

1. Run test 1221 described in this document.
2. Rename the grade file grade_11141.txt.
3. Submit the grade file to the Project Manager.

6.2 Test 11142: ATNA Certificate Exchange with Unregistered Certificate

1. Run test 1222 described in this document.
2. Rename the grade file grade_11142.txt.
3. Submit the grade file to the Project Manager.

6.3 Test 11143: ATNA Certificate Exchange with Expired Certificate

1. Run test 1223 described in this document.
2. Rename the grade file grade_11143.txt.
3. Submit the grade file to the Project Manager.

7 ATNA Tests for Server Applications

The tests in this section are for ATNA applications that accept TLS connections. In that sense, these are considered server applications.

7.1 Test 11141: ATNA Certificate Exchange with Valid Certificate

1. Run test 1221 described in this document.
2. Rename the grade file grade_11141.txt.
3. Submit the grade file to the Project Manager.

7.2 Test 11142: ATNA Certificate Exchange with Unregistered Certificate

1. Run test 1222 described in this document.
2. Rename the grade file grade_11142.txt.
3. Submit the grade file to the Project Manager.

7.3 Test 11143: ATNA Certificate Exchange with Expired Certificate

1. Run test 1223 described in this document.
2. Rename the grade file grade_11143.txt.
3. Submit the grade file to the Project Manager.