JPN\_IWE14057: TEST SPECIFICATION

Step-by-step guide to test JPN\_IWE14057 to Wind River VxWorks 6.2

16-Oct 15

Contents

[1. Development and Test Environment for JPN\_IWE14057 3](#_Toc432778897)

[2. Bringing up Wind River Board 3](#_Toc432778898)

[3. Test Environment Setup & Execution Procedure 4](#_Toc432778899)

[A. Test Preparation 4](#_Toc432778900)

[B. Test Procedure 5](#_Toc432778901)

[CASE:A01 DES-CBC3-MD5 5](#_Toc432778902)

[CASE:A02 DES-CBC3-MD5 6](#_Toc432778903)

[CASE:A03 RC4-MD5 6](#_Toc432778904)

[CASE:A04 RC4-MD5 7](#_Toc432778905)

[CASE:A05 DHE-RSA-AES256-SHA 7](#_Toc432778906)

[CASE:A06 DHE-RSA-AES256-SHA 7](#_Toc432778907)

[CASE:A07 EDH-RSA-DES-CBC-SHA 8](#_Toc432778908)

[CASE:A08 EDH-RSA-DES-CBC-SHA 8](#_Toc432778909)

[CASE: A09 ECDHE-RSA-AES256-SHA 8](#_Toc432778910)

[CASE: A10 ECDHE-RSA-AES256-SHA 9](#_Toc432778911)

[CASE: A11 ECDHE-ECDSA-AES256-SHA 9](#_Toc432778912)

[CASE: A12 ECDHE-ECDSA-AES256-SHA 10](#_Toc432778913)

[CASE: A13 ECDHE-RSA-DES-CBC3-SHA 10](#_Toc432778914)

[CASE: A14 ECDHE-RSA-DES-CBC3-SHA 10](#_Toc432778915)

[CASE :A15 DES-CBC3-SHA 11](#_Toc432778916)

[CASE: A16 DES-CBC3-SHA 11](#_Toc432778917)

[CASE: A17 DHE-RSA-AES128-SHA 12](#_Toc432778918)

[CASE: A18 DHE-RSA-AES128-SHA 12](#_Toc432778919)

[CASE: A19 ECDHE-RSA-RC4-SHA 12](#_Toc432778920)

[CASE: A20 ECDHE-RSA-RC4-SHA 13](#_Toc432778921)

[CASE: A21 ECDHE-ECDSA-RC4-SHA 13](#_Toc432778922)

[CASE: A22 ECDHE-ECDSA-RC4-SHA 14](#_Toc432778923)

[CASE: A23 RC4-SHA 14](#_Toc432778924)

[CASE: A24 RC4-SHA 14](#_Toc432778925)

[CASE: A25 DES-CBC-SHA 15](#_Toc432778926)

[CASE: A26 DES-CBC-SHA 15](#_Toc432778927)

[CASE: A27 DES-CBC3-SHA 16](#_Toc432778928)

[CASE: A28 DES-CBC3-SHA 16](#_Toc432778929)

[CASE: A29 AES256-SHA 16](#_Toc432778930)

[CASE: A30 AES256-SHA 17](#_Toc432778931)

[CASE: A31 ECDHE-RSA-RC4-SHA 17](#_Toc432778932)

[CASE: A32 ECDHE-RSA-RC4-SHA 18](#_Toc432778933)

[CASE: A33 ECDHE-RSA-DES-CBC3-SHA 18](#_Toc432778934)

[CASE: A34 ECDHE-RSA-DES-CBC3-SHA 18](#_Toc432778935)

[CASE: A35 ECDHE-RSA-AES256-SHA 19](#_Toc432778936)

[CASE: A36 ECDHE-RSA-AES256-SHA 19](#_Toc432778937)

[CASE: A37 ECDHE-ECDSA-RC4-SHA 20](#_Toc432778938)

[CASE: A38 ECDHE-ECDSA-RC4-SHA 20](#_Toc432778939)

[CASE: A39 ECDHE-ECDSA-DES-CBC3-SHA 21](#_Toc432778940)

[CASE: A40 ECDHE-ECDSA-DES-CBC3-SHA 21](#_Toc432778941)

[CASE: A41 ECDHE-ECDSA-AES256-SHA 21](#_Toc432778942)

[CASE: A42 ECDHE-ECDSA-AES256-SHA 22](#_Toc432778943)

[CASE: A43 DHE-RSA-AES256-SHA 22](#_Toc432778944)

[CASE: A44 DHE-RSA-AES256-SHA 23](#_Toc432778945)

[CASE: A45 AES256-SHA256 23](#_Toc432778946)

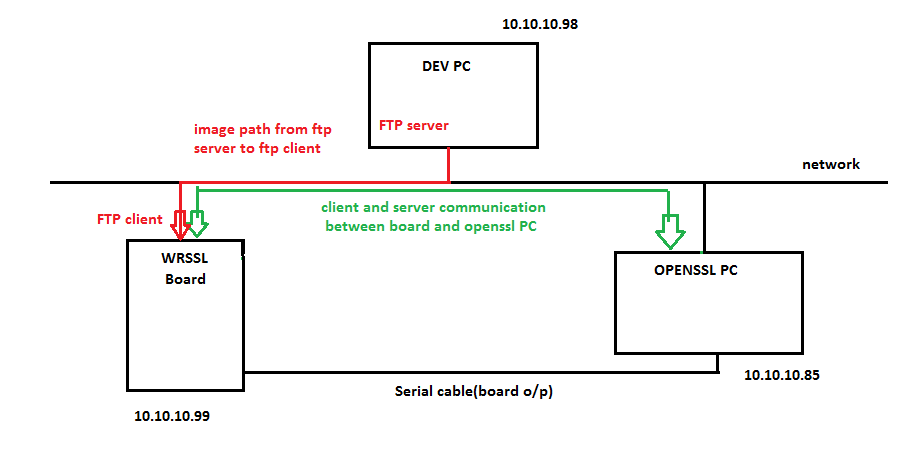
[CASE: A46 AES256-SHA256 23](#_Toc432778947)

[CASE: A47 ECDHE-ECDSA-AES256-SHA 24](#_Toc432778948)

[CASE: A48 ECDHE-ECDSA-AES256-SHA 24](#_Toc432778949)

# Development and Test Environment for JPN\_IWE14057

Development and test environment for the JPN-IWE14057 project is as shown below.



SSL Client-Server communication between VxWorks target board and OpenSSL PC

Image download via FTP

Target board running

VxWorks 6.2 + WR-SSL

PC running Ubuntu Linux + OpenSSL

PC running Win XP + Workbench

Ensure that required hardware and software are available to realize below setup before following instructions in this document.

Document also assumes that above setup is realized with required physical connections made as shown in above diagram.

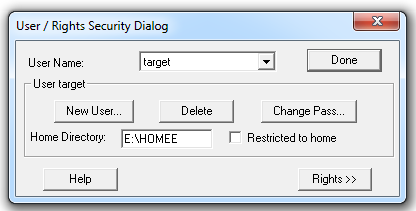
# Bringing up Wind River Board

1. Power on the WindRivertargetboard.
2. Target board should be connected to OPENSSL PC using serial port.[usepicocom for serial input/output or to see the console log of board. ].

**i.e.picocom -b 9600 /dev/ttyS0**

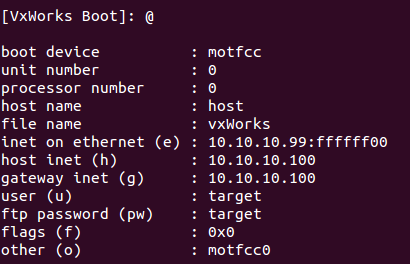
1. Make a LAN connection between Target board and system in which ftp server is running, i.e., DEV PC.[FTP server ispart ofwindriverpackageand available under **start->all programs->windriver->vxworks-6.2->ftp server]**.
2. Board and DEV PC should be on same network.
3. Configure ftp server as below:

* Launch ftp server from **start->all programs->windriver->vxworks-6.2->ftp server.**
* Select **security->user/rights.**Pop up window as shown below fig will open.
* Select username( if no user available then create new by pressing 'new user') and set password and home directory for that user.(set the directory where vxworks images will bepresent as the home directory of ftpserver).



1. Switch off and switch on the board and wait for ‘[VxWorks Boot]’command prompt.
2. Configure board as below: ( press 'c' to get configure window)

* Select boot device( type 'help' to get the list of supported boot device).
* set appropriate IP address for board.
* hostip and gateway should be same as FTP server system's IP address.
* FTP user and password field should match with FTP server configuration.
* following fig. shows sample example



1. Once board configuration is done, ‘[VxWorks Boot]’ prompt will be displayed again. You may use command ‘p’ to ensure that configuration is correct.
2. Copy Vxworks images created to FTP home directory.

# Test Environment Setup & Execution Procedure

## Test Preparation

**On OPENSSL-LINUX PC**

* + - 1. download and install OPENSSL-1.0.1o from <https://www.openssl.org/source/openssl-1.0.1o.tar.gz>
      2. Following certificates are to be generated and should be kept under respective folders.

md5WithRSAEncryption

$opensslgenrsa -out md5rsakey.pem 2048

$opensslreq -out md5rsa.csr -key md5rsakey.pem -new -md5

$opensslreq -in md5rsa.csr -out md5rsa.pem -text

sha1WithRSAEncryption

$opensslgenrsa -out sha1rsakey.pem 2048

$opensslreq -out sha1rsa.csr -key sha1rsakey.pem -new -sha1

$opensslreq -in sha1rsa.csr -out sha1rsa.pem -text

sha256WithRSAEncryption

$opensslgenrsa -out sha256rsakey.pem 2048

$opensslreq -out sha256rsa.csr -key sha256rsakey.pem -new -sha256

$opensslreq -in sha256rsa.csr -out sha256rsa.pem -text

SHA1withECDSA

$opensslecparam -name secp256k1 -genkey -out sha1ecdsakey.pem

$opensslreq -new -x509 -key sha1ecdsakey.pem -out sha1ecdsa.pem -days 730

Note: You mayverifythe key and the certificate contents using

$opensslecparam -in sha1ecdsakey.pem -text –noout

$openssl x509 -in sha1ecdsa.pem -text -noout

sha256withECDSA

$opensslecparam -name secp256k1 -genkey -out sha256ecdsakey.pem

$opensslreq -new -x509 -key sha256ecdsakey.pem -out sha256ecdsa.pem -days 730

Note: You may verify the key and the certificate contents using

$opensslecparam -in sha256ecdsakey.pem -text -noout

$openssl x509 -in sha256ecdsa.pem -text -noout

* + - 1. Copy above generated certificate and key files to openssl-1.0.1o folder on OPENSSL-LINUX PC
      2. Transfer above generated certificate and key files (via FTP or any other method) to home directory of FTP server on DEV-PC.

## Test Procedure

### CASE:A01 DES-CBC3-MD5

**On WRSSL-Board and picocomon OPENSSL-PC**

* + - 1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server"5","-cert", "md5rsa.pem", "-key", "md5rsakey.pem","-ssl2"

On OPENSSL-PC

* + - 1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -ssl2 -cipher DES-CBC3-MD5 -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE:A02 DES-CBC3-MD5

**On OPENSSL-LINUX -PC:**

* + - 1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd/home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert md5rsa.pem –key md5rsakey.pem -ssl2 -msg

**On WRSSL-Board and picocom on OPENSSL-PC**

* + - 1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client"5","-ssl2","-cipher","DES-CBC3-MD5","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE:A03 RC4-MD5

**On WRSSL-Board and picocom on OPENSSL-PC:**

* + - 1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server"5","-cert", "md5rsa.pem", "-key", "md5rsakey.pem","-ssl2"

**On OPENSSL-LINUX PC**

* + - 1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -ssl2 -cipher RC4-MD5 -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE:A04 RC4-MD5

**On OPENSSL-LINUX -PC:**

* + - 1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd/home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert md5rsa.pem -key md5rsakey.pem -ssl2 -msg

* + - 1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client"5","-ssl2","-cipher","RC4-MD5","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE:A05 DHE-RSA-AES256-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

* + - 1. Run WRSSL as server using following command in WRSSL-PC picocom terminal

->nm\_server"5","-cert", "sha1rsa.pem", "-key", "sha1rsakey.pem","-ssl3"

**On OPENSSL-LINUX -PC:**

* + - 1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -ssl3 -cipherDHE-RSA-AES256-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE:A06 DHE-RSA-AES256-SHA

**On OPENSSL-LINUX -PC:**

* + - 1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd/home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha1rsa.pem -key sha1rsakey.pem -ssl3 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

* + - 1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client "5","-ssl3","-cipher","DHE-RSA-AES256-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE:A07 EDH-RSA-DES-CBC-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

* + - 1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server "5","-cert", "sha1rsa.pem", "-key", "sha1rsakey.pem","-ssl3"

**On OPENSSL-LINUX -PC:**

* + - 1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -ssl3 -cipher EDH-RSA-DES-CBC-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE:A08 EDH-RSA-DES-CBC-SHA

**On OPENSSL-LINUX -PC:**

* + - 1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha1rsa.pem -key sha1rsakey.pem -ssl3 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

* + - 1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client"5","-ssl3","-cipher","ECDHE-ECDSA-AES256-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A09 ECDHE-RSA-AES256-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

* + - 1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server"5","-cert", "sha1rsa.pem", "-key", "sha1rsakey.pem","-ssl3")

**On OPENSSL-LINUX -PC:**

* + - 1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd/home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -ssl3 -cipher ECDHE-RSA-AES256-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A10 ECDHE-RSA-AES256-SHA

**On OPENSSL-LINUX -PC:**

* + - 1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha1rsa.pem -key sha1rsakey.pem -ssl3 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

* + - 1. Run WRSSL as client using following command in WRSSL PC picocom terminal.

->nm\_client"5","-ssl3","-cipher","ECDHE-RSA-AES256-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A11 ECDHE-ECDSA-AES256-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server "5","-cert", "sha1ecdsa.pem", "-key", "sha1ecdsakey.pem","-ssl3"

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd/home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -ssl3 -cipher ECDHE-ECDSA-AES256-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A12 ECDHE-ECDSA-AES256-SHA

**On OPENSSL-LINUX -PC:**

1.Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha1ecdsa.pem -key sha1ecdsakey.pem -ssl3 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client"5","-ssl3","-cipher","ECDHE-ECDSA-AES256-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A13 ECDHE-RSA-DES-CBC3-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server"5","-cert", "sha1rsa.pem", "-key", "sha1rsakey.pem","-ssl3"

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -ssl3 -cipher ECDHE-RSA-DES-CBC3-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A14 ECDHE-RSA-DES-CBC3-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha1rsa.pem -key sha1rsakey.pem -ssl3 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

* nm\_client "5","-ssl3","-cipher","ECDHE-RSA-DES-CBC3-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE :A15 DES-CBC3-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server "5","-cert", "sha1rsa.pem", "-key", "sha1rsakey.pem","-ssl3")

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -ssl3 -cipher DES-CBC3-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A16 DES-CBC3-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha1rsa.pem -key sha1rsakey.pem -ssl3 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client"5","-ssl3","-cipher","DES-CBC3-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A17 DHE-RSA-AES128-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server "5","-cert", "sha1rsa.pem", "-key", "sha1rsakey.pem","-ssl3")

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client –ssl3 -cipher DHE-RSA-AES128-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A18 DHE-RSA-AES128-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha1rsa.pem -key sha1rsakey.pem –ssl3 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client"5","-ssl3","-cipher","DHE-RSA-AES128-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A19 ECDHE-RSA-RC4-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server "5","-cert", "sha1rsa.pem", "-key", " sha1rsakey.pem","-ssl3")

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client –ssl3 -cipher ECDHE-RSA-RC4-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A20 ECDHE-RSA-RC4-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha1rsa.pem -key sha1rsakey.pem –ssl3 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client "5","-ssl3","-cipher","ECDHE-RSA-RC4-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A21 ECDHE-ECDSA-RC4-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server "5","-cert", "sha1ecdsa.pem", "-key", " sha1ecdsakey.pem","-ssl3"

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL client as shown below

$cd/home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client –ssl3 -cipherECDHE-ECDSA-RC4-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A22 ECDHE-ECDSA-RC4-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha1ecdsa.pem -key sha1ecdsakey.pem –ssl3 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client "5","-ssl3","-cipher","ECDHE-ECDSA-RC4-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A23 RC4-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

e.g.

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server"5","-cert", "sha256rsa.pem", "-key", " sha256rsakey.pem","-tls1")

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -tls1 -cipher RC4-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A24 RC4-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha256rsa.pem -key sha256rsakey.pem -tls1 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client"5","-tls1","-cipher","RC4-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A25 DES-CBC-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server"5","-cert", "sha256rsa.pem", "-key", " sha256rsakey.pem","-tls1"

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -tls1 -cipher DES-CBC-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A26 DES-CBC-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha256rsa.pem -key sha256rsakey.pem -tls1 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client"5","-tls1","-cipher","DES-CBC-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A27 DES-CBC3-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server"5","-cert", "sha256rsa.pem", "-key", " sha256rsakey.pem","-tls1"

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -tls1 -cipher DES-CBC3-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A28 DES-CBC3-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha256rsa.pem -key sha256rsakey.pem -tls1 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client"5","-tls1","-cipher","DES-CBC3-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A29 AES256-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server"5","-cert", "sha256rsa.pem", "-key", " sha256rsakey.pem","-tls1"

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -tls1 -cipher AES256-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A30 AES256-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha256rsa.pem -key sha256rsakey.pem -tls1 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client"5","-tls1","-cipher","AES256-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A31 ECDHE-RSA-RC4-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server"5","-cert", "sha256ecdsa.pem", "-key", "sha256ecdsakey.pem","-tls1"

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -tls1 -cipher ECDHE-RSA-RC4-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A32 ECDHE-RSA-RC4-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha256ecdsa.pem -key sha256ecdsakey.pem -tls1 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client "5","-tls1","-cipher","ECDHE-RSA-RC4-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A33 ECDHE-RSA-DES-CBC3-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server "5","-cert", "sha256rsa.pem", "-key", " sha256rsakey.pem","-tls1")

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -tls1 -cipher ECDHE-RSA-DES-CBC3-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A34 ECDHE-RSA-DES-CBC3-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha256rsa.pem -key sha256rsakey.pem -tls1 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

nm\_client "5","-tls1","-cipher","ECDHE-RSA-DES-CBC3-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A35 ECDHE-RSA-AES256-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server "5","-cert", "sha256rsa.pem", "-key", " sha256rsakey.pem","-tls1"

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -tls1 -cipher ECDHE-RSA-AES256-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A36 ECDHE-RSA-AES256-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha256ecdsa.pem -key sha256ecdsakey.pem -tls1 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client"5","-tls1","-cipher","ECDHE-RSA-AES256-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A37 ECDHE-ECDSA-RC4-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server"5","-cert", "sha256ecdsa.pem", "-key", " sha256ecdsakey.pem","-tls1"

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -tls1 -cipher ECDHE-ECDSA-RC4-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A38 ECDHE-ECDSA-RC4-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha256ecdsa.pem -key sha256ecdsakey.pem -tls1–msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client"5","-tls1","-cipher","ECDHE-ECDSA-RC4-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A39 ECDHE-ECDSA-DES-CBC3-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server "5","-cert", "sha256ecdsa.pem", "-key", " sha256ecdsakey.pem","-tls1"

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -tls1 -cipher ECDHE-ECDSA-DES-CBC3-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A40 ECDHE-ECDSA-DES-CBC3-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha256ecdsa.pem -key sha256ecdsakey.pem -tls1 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

nm\_client"5","-tls1","-cipher","ECDHE-ECDSA-DES-CBC3-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A41 ECDHE-ECDSA-AES256-SHA

**On OPENSSL-LINUX -PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server "5","-cert", "sha256ecdsa.pem", "-key", " sha256ecdsakey.pem","-tls1\_1")

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -tls1\_1 -cipher ECDHE-ECDSA-AES256-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A42 ECDHE-ECDSA-AES256-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha256ecdsa.pem -key sha256ecdsakey.pem -tls1\_1 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client"5","-tls1\_1","-cipher","ECDHE-ECDSA-AES256-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A43 DHE-RSA-AES256-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server"5","-cert", "sha256rsa.pem", "-key", " sha256rsakey.pem","-tls1\_1"

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -tls1\_1 -cipher DHE-RSA-AES256-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A44 DHE-RSA-AES256-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run OpenSSL server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha256rsa.pem -key sha256rsakey.pem -tls1\_1 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

nm\_client"5","-tls1\_1","-cipher","DHE-RSA-AES256-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A45 AES256-SHA256

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server"5","-cert", "sha256rsa.pem", "-key", " sha256rsakey.pem","-tls1\_2"

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -tls1 -cipher AES256-SHA256 -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A46 AES256-SHA256

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha256rsa.pem -key sha256rsakey.pem -tls1\_2–msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client"5","-tls1\_2","-cipher","AES256-SHA256","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A47 ECDHE-ECDSA-AES256-SHA

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as server using following command in WRSSL-PC picocom terminal.

->nm\_server"5","-cert", "sha256ecdsa.pem", "-key", " sha256ecdsakey.pem","-tls1\_2"

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl client as shown below

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_client -tls1\_2 -cipherECDHE-ECDSA-AES256-SHA -connect <host>:4433

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

### CASE: A48 ECDHE-ECDSA-AES256-SHA

**On OPENSSL-LINUX -PC:**

1. Open terminal, change directory to openssl-1.0.1o and run openssl server as shown below

e.g.

$cd /home/user1/openssls/openssl-1.0.1o

$./apps/openssls\_server -accept 4433 -cert sha256ecdsa.pem -key sha256ecdsakey.pem -tls1\_2 -msg

**On WRSSL-Board and picocom on OPENSSL-PC:**

1. Run WRSSL as client using following command in WRSSL-PC picocom terminal.

->nm\_client "5","-tls1","-cipher","ECDHE-ECDSA-AES256-SHA","-connect","<OPENSSL\_PC\_IPAddr>:4433"

###### **Expected result:**

SSL connection should establish successfully. Verify by typing in a test message on the client terminal and seeing the same on server terminal and vice versa. Try to close connection from client. Connection should get closed without problems.

/EOD