Network Device Interpretation # 201705

Handling of TLS connections with and without mutual authentication

Status: ☒ *Active* ☐ *Inactive*

Date: *26-Jul-2017*

Type of Document: ☒ *Technical Decision* ☐ *Technical Recommendation*

Approved by: ☒ *Network iTC Interpretations Team* ☒ *Network iTC*

Affected Document(s): *ND SD V1.0, ND SD V2.0*

Affected Section(s): *FCS_DTLSC_EXT.2.5 (ND SD V2.0), FCS_TLSC_EXT.2 (ND SD V1.0, ND SD V2.0)*

Superseded Interpretation(s): *None*

Issue:

FCS_TLSC_EXT.2.5, Test 1: The evaluator shall perform the following modification to the traffic:

a) Configure the server to require mutual authentication and then modify a byte in a CA field in the Server's Certificate Request handshake message. The modified CA field must not be the CA used to sign the client's certificate. The evaluator shall verify the connection fails.

Issues:

The test case cannot be carried out as written. Additionally, the intention behind this test case is not relevant to the SFR.

1. There is no "CA field" in the Server's Certificate Request (type 13). Closest is certificate authorities field that contains a list of DN of trusted CAs (aka trusted roots).

2. Any test modifying the Server's Certificate Request (type 13) would be exercising SERVER functionality, as enforcing mutual authentication by verifying client's certificate is purely SERVER's functionality. Therefore, this test case is only appropriate for FCS_TLSS_EXT.2.x SFRs. The expected outcome (connection fails) is enforced by the server, and not the TOE acting as a client.

Specifically, RFC 5246 on page 53 states:

- That the certificate Request message structure is formed by: client certificate type, supported signature algorithms and the trusted CA distinguished names

- The certificate request message is not hashed or singed.

- The client verify message is a digitally- signed message of all handshake messages sent or received, starting at client hello and up to, but not including, the certificate verify message. This message is used to provide explicitly verification of the client certificate.

Flow of handshake messages between the TLS client and the server:

Client ----------Client Hello---------- → Server

Client <----------Server Hello / certificate / server key exchange / certificate request / server hello done---------- Server

Client ----------Certificate / client key exchange/ Certificate verify / ChangecipherSpec / Finished ---------- → Server

Client <------ ChangeCipherSpec / Finished message ---------- Server

Conclusion/Recommendation: FCS_TLSC_EXT.2.5 SFR specifies mutual authentication. To appropriately test CLIENT response is to verify that TOE responds to Server's Certificate Request with Certificate (Type 14) and CertifiacteVerify (Type 15) messages. However, verification of these messages is entirely up to the server.

Resolution:

The NIT acknowledges the issue described in the 'Issue' section above and in particular that the functionality in the focus of the test is sever functionality but not client functionality. The NIT is of the opinion that this test on the client side should therefore be replaced by a test that is intended to verify that a TOE which is a TLS client supporting mutual authentication is handling correctly both, connection requests with and without mutual authentication. For ND SD V1.0 and ND SD V2.0 FCS_TLSC_EXT.2.5 Test 1 shall therefore be modified as follows:

"The purpose of these tests is to confirm that the TOE appropriately handles connection to peer servers that support and do not support mutual authentication.

Test 1: The evaluator shall establish a connection to a peer server that is not configured for mutual authentication (i.e. does not send Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE did not send Client's Certificate message (type 11) during handshake.

Test 2: The evaluator shall establish a connection to a peer server with a shared trusted root that is configured for mutual authentication (i.e. it sends Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) messages."

For ND SD V2.0 this consideration applies not only to FCS_TLSC_EXT.2.5 but also FCS_DTLSC_EXT.2.5. FCS_DTLSC_EXT.2.5 Test 1 shall therefore be modified as follows:

"The purpose of these tests is to confirm that the TOE appropriately handles connection to peer servers that support and do not support mutual authentication.

Test 1: The evaluator shall establish a connection to a peer server that is not configured for mutual authentication (i.e. does not send Server's Certificate Request (type 13) message). The evaluator

observes negotiation of a DTLS channel and confirms that the TOE did not send Client's Certificate message (type 11) during handshake.

Test 2: The evaluator shall establish a connection to a peer server with a shared trusted root that is configured for mutual authentication (i.e. it sends Server's Certificate Request (type 13) message). The evaluator observes negotiation of a DTLS channel and confirms that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) messages."

Rationale:

see Resolution section

Further Action:

None

Action by Network iTC:

None