Network Device Interpretation # 35

Failure testing for TLS session establishment, relation to TD0040, T3

Status: ☒ *Active* ☐ *Inactive*

Date: *21-Nov-2016*

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

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

Affected Document(s): *ND cPP V1.0, FW cPP V1.0*

Affected Section(s): *FCS_TLSS_EXT.1.1*

Superseded Interpretation(s): *None*

Issue:

Further clarification on TD0040 needed with respect to NDcPP(138)

I was working on the NDcPP for a pre-evaluation of a product and ran into difficulties understanding FCS_TLSS_EXT.1.1, Test 3. It turns out that the same issue in the NDcPP was discovered for the MDM 2.0 and codified as TD0040. I assume I can reuse this decision for the NDcPP.

However, after looking at TD0040, I believe there still needs to be a clarification. TD0040 states:

> *Test 3: Change wording to:*
>
> *Test 3: The evaluator shall use a client to send a key exchange message in the TLS connection that the does not match the server-selected ciphersuite (for example, send an ECDHE key exchange while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite or send a RSA key exchange while using one of the ECDSA ciphersuites.) The evaluator shall verify that the TOE sends a fatal alert after receiving the client's change cipher spec message.*

The problem here is that no compliant-TLS server should send a fatal alert message because:

> *(a) no secret was successfully exchanged allowing for encrypted alerts and*
>
> *(b) handshaking failure alerts are not to be sent in this case as per RFC5246, section 7.4.7.1:* *"*In any case, a TLS server MUST NOT generate an alert if processing an RSA-encrypted premaster secret message fails*."*

Instead, I think the wording should be changed to something like this:

"The evaluator shall verify that the TOE disconnects after receiving the client Finished message." (as per figure 1 of RFC5246, the client sends three messages to finalize their end of the handshake without any acknowledgement from a server until the Finished message is sent).

Answer provided by NIAP TRRT:

Unfortunately, we are unable to provide interpretations/decisions regarding cPPs. However, an Interpretations Team has been established by the iTC to discuss inquiries such as these. We will be forwarding your message shortly, however, we recommend your participation in the iTC to bring this to closure.

Resolution:

The requestor is correct in the RSA case. If a ciphersuite requiring RSA key exchange has been selected, then the server must terminate the connection without an alert after receiving the client's ChangeCipherSpec message or Finished message. If a ciphersuite requiring Diffie-Hellman key agreement has been selected, then the server may send an alert or simply terminate the connection after receiving the client's ChangeCipherSpec message or Finished message. Since the server cannot distinguish the different cases the Test Case 3 shall be changed to:

Test 3: The evaluator shall use a client to send a key exchange message in the TLS connection that does not match the server-selected ciphersuite (for example, send an ECDHE key exchange while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite or send a RSA key exchange while using one of the ECDSA ciphersuites.) The evaluator shall verify that the TOE either sends an alert after receiving the client's ChangeCipherSpec message or Finished message; or terminates the connection after receiving the client's ChangeCipherSpec message or Finished message.

Rationale:

The client's key exchange message doesn't have any encoding to allow the server to distinguish between an RSA, DHE, or ECDHE key exchange message. Unfortunately, in the case of RSA key exchange, Section 7.4.7.1 of the TLS v1.2 RFC 5246, states in the event of a bad key exchange:

> In any case, a TLS server MUST NOT generate an alert if processing an RSA-encrypted premaster secret message fails, or the version number is not as expected. Instead, it MUST continue the handshake with a randomly generated premaster secret.

*Therefore, in the RSA case, the test **must not** require that any alert be sent.*

*In the Diffie-Hellman case, the server **may** send an alert or simply terminate the connection.*

The rationale for allowing connection termination after the server receives either the client's ChangeCipherSpec message or the client's Finished message is that some implementations involving hardware acceleration batch the client's handshake messages through the Finished message, so in those cases, any response is made after receiving the client's Finished message.

Further Action:

None

Action by Network iTC:

None.