|  |  |  |  |
| --- | --- | --- | --- |
| **Item Title** | Consolidated iTC Feedback – NDcPP v2 | **Reviewer** | iTC Editorial Team |
| **Item Identifier** | ND cPP v2 | **Review Date** | 24 March 2016 |
| **Version; Date:** | v-1-0-3; 14/03/16 |  |  |

**Notes :-**

|  |  |  |
| --- | --- | --- |
| Severity | 1 | **Significant** - Impact the correct or efficient operation of the item. Needs discussion during a review meeting. |
| 2 | **Moderate** - Normally clarifications or proposed improvements to the item which are unlikely to impact other areas. Probably doesn’t need discussing at a review meeting. |
| 3 | **Minor** - Does not affect the correct operation or interpretation of the item. These are usually syntax and format errors which have no effect on the meaning or interpretation of the item. |

**This is a public commenting process: the text of comments and responses may be distributed, or made available in other ways during the process, without restriction. If you wish to submit private comments then please contact the iTC Chair or Technical Editors directly.**

| **No.** | **Location** | **Comment** | **Suggested Change** | **Severity** | **Action** |
| --- | --- | --- | --- | --- | --- |
|  | NDcPP, B.2.1.5 | For distributed TOEs selecting “TLS” in FPT\_ITT.1 means FCS\_TLSC\_EXT.1.2 must be included.  FCS\_TLSC\_EXT.1.2, and specifically RFC 6125, would require the “joiner” TOE component to build a reference identifier independent of its “Gatekeeper”.  However some distributed components will find its “Gatekeeper” based on messaged exchanged in a ‘discovery’ process. Based on attributes returned in those messages it selects a “Gatekeeper” to begin the authentication and join process.  This means the “joiner” doesn’t independently build a reference identifier, nor is it possible to configure one. Therefore it’s not accurate to apply RFC 6125 in FCS\_TLSC\_EXT.1.2 for these distributed TOEs.  FCO\_CPC\_EXT.1 which requires the administrator to enable communications between the pair of TOE components is acceptable/sufficient for distributed TOEs. | Two suggested solutions are below:   1. Create a separate iteration of FCS\_TLSC\_EXT.1 called FCS\_TLSC\_EXT.1/ITT and remove FCS\_TLS\_EXT.1.2 element.   - or-   1. Remove the FCS\_TLS\_EXT.1.2 element from the current FCS\_TLS\_EXT.1 component.   This would permit an ST author to select TLS in FPT\_ITT.1 (and in FTP\_TRP.1/Admin) without the issue as identified in the comment column.  The requirement to verify that the presented identifier matches the reference identifier would remain in the FCS\_TLS\_EXT.2 component, which is required to be selected for TLS connections to external IT entities. | 1 | An additional note has been added to Application note 53 which outlines that the need to create a reference identifier (as outlined in RFC 6125) is relaxed for an inter-TSF channel. |
|  | NDcPP, A.5.2 | FPT\_ITT.1 needs to allow use of DTLS as a protocol. For a component to manage another component as described in use case 3, some distributed TOEs may use a different protocol.  DTLS is similar to TLS but uses UDP instead of TCP. | Due to space limitations see attached document “**Suggested Change-Add DTLS**” for details. | 1 | Requestor has been asked to supply a more comprehensive set of DTLS SFR(s) and EAs for consideration by the iTC. This is work in progress now within the iTC. |
|  | NDcPP, page 3 | The defined use cases for distributed TOEs are described in terms of fulling cPP requirements but not cPP plus extended package PPs developed for use by national schemes (i.e., US PPs developed by NIAP-CCEVS).  There are network device products that become distributed when an extended package PP is used. For example NDcPP plus WLAN EP. | Please adjust the wording in section 3 to allow cPP plus extended package PPs as valid uses cases for distributed TOEs. | 2 | No action taken. EPs are specific to one scheme (i.e. NIAP) and should therefore not be specifically referenced in a cPP. |
|  | NDcPP, B.2.1.5 | The mandatory ciphersuite in FCS\_TLSC\_EXT.1.1 was removed and replaced with a list of optional ciphersuites. | Since the intention was to provide a list of ciphersuites for the ST author to choose, remove the word “Optional” and remove the statement from Application Note 81 which says TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA is required. | 3 | The word ‘Optional’ has been removed. |
|  | NDcPP, Page 49 | "…be audited on botheach component side." | “… be audited on both sides of each component…” | 3 | Typo not found in v1-0-3 and it is suspected that this comment is based on a local formatting issue within Word. |
|  | NDcPP, Page 50 | ITCFIA\_X509\_EXT.1/Rev | FIA\_X509\_EXT.1/Rev | 3 | Typo not found in v1-0-3 and it is suspected that this comment is based on a local formatting issue within Word. |
|  | NDcPP, Page 80 | "FDP\_ITCFTP\_ITC.1 or FTP\_ITTFPT\_ITT.1)" | FTP\_ITC.1 or FPT\_ITT.1 | 3 | Typo not found in v1-0-3 and it is suspected that this comment is based on a local formatting issue within Word. |
|  | NDcPP, Pages 104 and 106 | There is a formatting issue with the bullets. The optional cipher suites are under the application note. | Correct formatting | 3 | Typo not found in v1-0-3 and it is suspected that this comment is based on a local formatting issue within Word. |
|  | NDcPP, Page 111 | TLS 10.2 is listed | must mean TLS v1.2 | 3 | Typo not found in v1-0-3 and it is suspected that this comment is based on a local formatting issue within Word. |
|  | NDcPP, IPsec in Appendix B Selection-Based Requirements and Appendix C Extended Component Definitions | IPsec in Appendix B is defined as:  FCS\_IPSEC\_EXT.1.11 The TSF shall ensure that all IKE protocols implement DH ~~Groups 14 (2048-bit MODP), and~~ [selection: 14 (2048-bit MODP, 19 (256-bit Random ECP), 5 (1536-bit MODP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP), no other DH groups].  IPsec in Appendix C is defined as:  FCS\_IPSEC\_EXT.1.11 The TSF shall ensure that all IKE protocols implement DH Groups ~~14 (2048-bit MODP), and~~ [selection: 14 (2048-bit MODP), 19 (256-bit Random ECP), 5 (1536-bit MODP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP), [assignment: other DH groups that are implemented by the TOE], no other DH groups]. | Either add “…[assignment: other DH groups that are implemented by the TOE” to the selection operation in Appendix B or remove the assignment operation in Appendix C. | 1 | Assignment has been added to the selection within FCS\_IPSEC\_EXT.1.11 in Appendix B. |
|  | NDcPP, Footnote Page 17 | Broken link to the definition | Fix the broken link to Exact Conformance definition | 3 | Broken link has been fixed. |
|  | NDcPP, Section 3, page 19 | The term component is used throughout the description of distributed TOEs, with the exception at the end of the first paragraph where ‘elements’ is used. | Use the same terminology throughout for clearer understanding and consistency.  Change ‘element’ to ‘component’. | 3 | Language modified to retain consistency. |
|  | NDcPP, Section 3, page 20 | Reads as: “…fulfil the cPP a Management Component may also be offered for use with the product. However…” | Change to read: “…fulfil the cPP a Management Component may also be offered for use with the TOE. However…” | 3 | Language updated in line with suggested change. |
|  | NDcPP, Page 28, Table 1 | FAU\_STG.3/LocSpace description reads as “Display wanring for local storage space” | Change to read as “Display warning for local storage space” | 3 | Typo corrected. |
|  | NDcPP, Page 29, Table 1 | FCS\_RBG\_EXT.1 is required to be met by all components. If hardware entropy source is defined, such as a chip, how does a management component that is software package (JRE file) that can be loaded on any management workstation meet this requirement? | Change to “Feature Dependent” | 1 | No action taken. A software only TOE component e.g. management component is not support by version 2 of the cPP. Irrespective, FCS\_RBG\_EXT.1 may be iterated if the product has multiple entropy sources. |
|  | NDcPP, Page 29, Table 1 | FIA\_UIA\_EXT.1 description reads as “User Identificiation and Authentication” | Change to read as “User Identification and Authentication” | 3 | Typo corrected. |
|  | NDcPP, Page 44,  Section 6, added second paragraph | Reads as “…by at least one TOE component or whether they are dependent upon the feature being implemented by the TOE component.” | Change to read as “…by at least one TOE component or whether they are dependent upon the feature being implemented by one component or the TOE as a whole.” | 2 | No action taken as suggested change does not represent the intention of the original language. |
|  | NDcPP, Page 44, Section 6.1, Iteration | The definition for ‘Iterations’ is simply defined as “…a string starting with “/”. “  The format is not well defined and no example is provided, so is the format as follows:  FCS\_COP.1 abcde  /FCS\_COP.1 efgh  /FCS\_COP.1 jikl  Or as follows:  FCS\_COP.1/abcde  FCS\_COP.1/efgh  FCS\_COP.1/ijkl | Recommend adding an example of the format. | 1 | Example added to section 6.1. |
|  | NDcPP, Page 53, Section 6.3.1.2 FAU\_GEN.2 | Application Note 5 reads as:  Where an auditable event is instigated by another component, the component that records the event must associate the event with the identity of the instigator (applies to distributed TOEs only).  Not sure ‘instigated’ and ‘instigator’ are the correct term to use. | Recommend the application note be changed to read as:  Where an auditable event is triggered by another component, the component that records the event must associate the event with the identity of the user that caused the event (applies to distributed TOEs only). | 3 | Suggested change not implemented since the intention of the original language was not to imply that the user identity associated with a given event should be captured, rather it was that the initiating component identity should be captured. Language updated to reflect and clarify the original intention. |
|  | NDcPP, Page 54, Section 6.3.2.1, FAU\_STG\_EXT.1 | Fix the spacing in the application note 6, from  “….external(FTP\_ITC.1) or intercomponent(FPT\_ITT.1)….” | To:  “….external (FTP\_ITC.1) or intercomponent (FPT\_ITT.1 or FTP\_ITC.1)….” | 3 | Spacing fixed. |
|  | NDcPP,  AppNote 6  page 54 | “In case the transmission would not be done in real-time the TSS has to provide details about **the possible** as well as acceptable frequency for the transfer of audit data.” | Change “the possible” to “that possibility” | 3 | Application note 6 and 101 updated with new language to clarify intent of the requirement for non real-time audit transfer. |
|  | NDcPP,  AppNote 8  page 56 | “FCS\_CKM.1 mandates compliance to FIPS 186-4. Implementations according to **FIPS 186-2** shall be accepted for products entering evaluation no later than **December 31, 2015**.”  Should we even allow FIPS 186-2 anymore? By the time this version is published, Dec 2015 will be a long time ago. This looks like a non-compliance being allowed. | Remove this statement. | 1 | Statement removed. |
|  | NDcPP, Section 6.5.5.1  Page 64 | “The TSF shall validate the revocation status of the certificate using [selection: the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5280, Certificate Revocation List (CRL) as specified in **RFC 5759**, [assignment: list of sections which TSF enforces] ].”  RDC 5759, section 5, requires that CRL must be signed using ECDSA based on Suite B NSA requirement. ECDSA signature/verification is not mandated by NDcPP. See FCS\_COP.1.1/SigGen. Therefore, this RFC should be selectable (i.e., optional) for consistency purpose. | Make RFC 5759 selectable or optional. | 1 | No change. RFC 5759 is already selectable in the SFR. |
| 1. **a** | NDcPP, Section 6.5.5.2  Page 65 | “**FIA\_X509\_EXT.2.2** When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [selection: **allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate**].” | We should make it explicitly clear this is only applicable for HTTPS only. For TLS and IPsec, if the peer certificate is invalid for any reason, the connection cannot be established period (See FCS\_IPSEC\_EXT.1.14 and FCS\_TLSC\_EXT.1/2.3). | 3 | Application Note 22 updated to reflect that the selection within FIA\_X509\_EXT.2.2 should be consistent with FCS\_IPSEC\_EXT.1.14 and FCS\_TLSC\_EXT.1/2.3 |
|  | NDcPP, Section 6.6.3.1  Page 67 | * Ability to configure the interaction between TOE components;   Since distributed TOE is optional, we should make that clear. | * Ability to configure the interaction between TOE components, if applicable; | 3 | Suggested change implemented. |
|  | NDcPP, Section 7.3.2  Page 80 | “highlight any situation in which a secret value may be transmitted over a channel that uses a **shorter key** than the transmitted key”  We should use “comparable strengths” from NIST 800-57 since asymmetric and symmetric algorithms have different key length. | “highlight any situation in which a secret value/key may be transmitted over a channel that uses less comparable strength than the transmitted key” | 2 | Added reference to comparable strength with high level definition. Reference NIST 800-57 for further guidance. |
|  | NDcPP, Section C.2.1.1  Page 97 | “**FCS\_HTTPS\_EXT.1.3** The TSF shall [selection: not establish the connection, request authorization to establish the connection, no other action] if the peer certificate is deemed invalid.” | We should make it explicitly clear that the first two options are only applicable if we are claiming a HTTPS client or a HTTPS server that supports mutual authentication. The third option (‘no other action’) is only applicable for HTTPS server that don’t support mutual authentication as there is no peer certificate to verify. | 2 | Selection updated to remove the assignment ‘no other action’ option. |
|  | NDcPP, AppNote 75  Page 104 | “The list of trusted certification authorities can only be selected if x509v3-ecdsa-sha2-nistp256 or x509v3-ecdsa-sha2-nistp384 are selected in FCS\_SSHC\_EXT.1.5.” | Need to add x509v3-ecdsa-sha2-nistp521 as well to be consistent with FCS\_SSHC\_EXT.1.5. | 2 | Suggested change implemented. |
|  | NDcPP, Section C.2.1.5/.6  Page 106/108 | TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA is removed from mandatory ciphersuite but not added to the optional ciphersuite. | Add ciphersuite to the mandatory list. This looks like an oversight. | 2 | TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA added in to selection for FCS\_TLSC\_EXT.1.1 and FCS\_TLSC\_EXT.2.1 |
|  | NDcPP, Section 6.4.1.3, FCS\_CKM.4, page 57  NDcPP SD | The cryptographic key destruction methods specified in FCS\_CKM.4.1 does not cover all storage types (including SSDs) and are in conflict with some FIPS 140-2 requirements. If a FIPS 140-2 module goes into an error state with keys stored in memory, the module is not permitted to perform any cryptographic operations while an error state (AS09.05) and this would include requesting output from the RBG.  Additionally at higher FIPS 140-2 levels which include tamper response circuitry to ensure that an attacker gaining physical access to the box is unable to gain access to keys, the read-verify may not be possible as after destroying keys in active memory the module powers off to ensure that no residual volatile memory information is available.  The FCS\_CKM.4.1 activity in the supporting document currently only requires TSS verification and does not require any functional verification of the key destruction methods. In the current FCS\_CKM.4.1 activity “system power off” and “disconnection of trusted channels” are provided as examples, however the clearing of volatile memory during power off or the clearing of volatile memory by an OS or application during termination of a trusted channel does not require a method specified in the current FCS\_CKM.4.1 requirement to destroy the cryptographic keys in volatile memory and make them inaccessible to cryptographic functions in the TOE. | **Recommended update to FCS\_CKM.4:**  FCS\_CKM.4.1 The TSF shall destroy plaintext cryptographic keys in accordance with a specified cryptographic key destruction method [selection:  • For volatile memory the destruction shall be executed by  [selection: a direct overwrite consisting of zeroes, a direct overwrite consisting of a pseudo-random pattern using the TSF’s RBG, [assignment: other actions]]  • For non-volatile memory the destruction shall be executed by  [selection: a direct overwrite consisting of zeroes, a direct overwrite consisting of a pseudo-random pattern using the TSF’s RBG, a block erase, [assignment: other actions]]  ]  **Recommended update to FCS\_CKM.4 supporting document activity:**  Tests  The evaluator shall note which keys are present in the module and initiate the cryptographic key destruction methods specified. Following the completion of the cryptographic key destruction methods, the evaluator shall attempt to perform cryptographic operations using each of the plaintext keys that were stored in the TOE. The evaluator shall verify that each plaintext key cannot be accessed.  The evaluator shall monitor the physical interfaces of the TOE during cryptographic key destruction to ensure that no plaintext key information is released while the cryptographic destruction method is performed.  **Rationale:**  As the intent of the existing FCS\_CKM.4 requirement is to ensure that all plaintext keys are no longer accessible in their current memory location by performing a read verify for applicable memory types. The recommended update to the supporting documentation activity would require the evaluator to confirm that the cryptographic key destruction method was successful and the key is no longer available for use by cryptographic functions provided by the TOE.  The assurance provided by the updated activity would ensure a remote attacker would not be able to access a key once the cryptographic destruction activity has completed without gaining physical access to the TOE memory and performing advanced recovery techniques, which would not be applicable if the TOE is deployed in a physically secure environment. | 1 | FCS\_CKM.4 (and its Evaluation Activities) has been rewritten for this version, and is subject to ongoing discussions for future versions of the cPP. The commenter is requested to review the updated FCS\_CKM.4 language during the public review and provide feedback accordingly. |
|  | Page 96 | FCS\_SSHS\_EXT.1.8 is vague as to whether incoming packets, outgoing packets, or both are to be counted. By using the word “transmitted,” it would seem the intent is “transmitted by the TOE,” but the associated test case counts only the incoming packets transmitted by the SSH client.  (SSH uses separate keys for incoming and outgoing packets.) | In FCS\_SSH\_EXT.1.8, if the intent is to only count packets in one direction, then after the word “transmitted” add one of:   1. by the SSH client 2. by the TOE   If the intent is to count and rekey whichever direction reaches 2^28 packets first, then consider splitting this SFR element into two separate SFR elements: one for incoming packets and one for outgoing packets. | 1 | Test description has been updated as described in NIT Decision RfI#24. |
|  | Page 94 | FCS\_SSHC\_EXT.1.8 has the same issue as FCS\_SSHS\_EXT.1.8 mentioned above in item #1 above. | Same as above. | 1 | As noted for comment ‎31 above, the test description has been updated as described in NIT Decision RfI#24. |
|  | Page 60 | Application Note 23 refers to FCS\_CKM.1(1), but there is no FCS\_CKM.1(1). | Replace FCS\_CKM.1(1) with FCS\_CKM.1. | 3 | Change Implemented |
|  | Page 60 | Application Note 23 shows that FIA\_X509\_EXT.3 has a clear dependency on FCS\_CKM.1, yet the ECD for FIA\_X509\_EXT.3 (C.3.4.3 on page 131) states that FIA\_X509\_EXT.3 has no dependencies. In addition, FIA\_X509\_EXT.3.1 requires the TOE to perform key generation. | In the ECD for FIA\_X509\_EXT.3 on page 131, add FCS\_CKM.1 as a dependency. Also update the SFR Dependency Analysis in Appendix E Table 6. | 2 | Table 6 and ECD for FIA\_X509\_EXT.3 updated to include dependency for FCS\_CKM.1. |
|  | Page 51 | The FCS\_CKM.1 Application Note 8 wording of:  “If the TOE acts as a receiver in the RSA key establishment scheme, the TOE does not need to implement RSA key generation.”  causes a lot of confusion with vendors (i.e., is somewhat misleading). Vendors with TOEs that only support RSA believe that, due to this Application Note, their TOE does **not** have to perform any key generation in order to comply to FCS\_CKM.1. But, this is not true.  Reason 1: FIA\_X509\_EXT.3 requires key generation.  Reason 2: FCS\_CKM.1 does not have a “none” selection option, so at least one of the 3 key gen schemes must be selected. If FCS\_CKM.1 had a “none” selection option, then FCS\_CKM.1 would effectively be an optional SFR instead of mandatory SFR. | Consider enhancing the wording of Application Note 8 to say:  “If the TOE acts as a receiver in the RSA key establishment scheme, the TOE does not need to implement RSA key generation, but the ST must specify at least one of the FCS\_CKM.1 key generation standards in order to satisfy FIA\_X509\_EXT.3 and the TOE must implement the specified standard(s).” | 3 | Added clarifying reference within Application Note 8 to state that within a distributed TOE, if the TOE components acts as a receiver then it does not need to implement RSA key generation. |
|  | TOE Objectives | It is difficult to determine the intent (rationale for the wording) of many of the SFRs because no description of intent is provided by this PP; thus, the interpretation of the SFRs outside of the iTC has a tendency to be different from that of the iTC. The Application Notes to date have been a poor substitute for intent. | Consider adding TOE Objectives to this PP. | 2 | No change. Having found that TOE Objectives in PPs often fail to capture a useful layer of description of the TOE, we have deliberately tried to work with a new approach, based on: the ‘direct rationale’ mapping of threats to SFRs in section 4.1, the SFR Architecture in 6.2, the narrative text introducing the SFRs, and the Application Notes. We appreciate that this is not perfect, and would be happy to consider specific comments based on improving these sections. |
|  | *3.1 Case 3, Figure 7* | Could the definition of “similar” network device be clarified? Would network devices which have completely different data plane functionality, but which share a common code base/functionality for their management plane/management functionality be considered similar? Or, does this only refer to different hardware models in a family of devices? | Add an application note or some other clarification of the definition of the term. | 2 | The description for distributed TOEs has been updated and the reference to ‘similar’ network devices removed. |
|  | 3.1 | Does the management component have to be a hardware appliance, or can it be a software application or virtual appliance? | Add an application note or some other definition of what can be considered to be a | 2 | No change. The current version 2 NDcPP does not support software only or virtualised components. Therefore the management component must be a hardware appliance. |
|  | FCS\_COP.1/Hash | This currently reads “and cryptographic key sizes [*assignment:* *cryptographic key sizes*]”, which is language which is stricken through the current published revision of the NDcPP. It appears that it was included here by mistake since hash algorithms do not use keys | Remove the language, as there are no key sizes in hash algorithms. | 3 | Agree that reference to key size in the context of a hash is unnecessary. SFR has been modified to strikethrough the reference to key size and added a reference to the digest size. |
|  | NDcPP/NDSD | FIA\_X509: it is possible to make selections that would lead to FIA\_X509 certificates not actually being needed in order to meet the SFRs, yet the requirements are still left as required | Make the X.509-related requirements objective rather than mandatory | 3 | All X.509 SFRs have been moved to Appendix B, and are now selection-based requirements. |
|  | *general* | Several SFRs require compliance to one or more RFCs. The idea behind such a requirement is usually not to demonstrate exact compliance to the full RFC but to dedicated parts of the RFC which is related to security as addressed by the SRF. | A general application note should be added to the cPP that clarifies compliance to the relevant parts of the referenced RFC(s) is demonstrated by the SFR and fulfilling the corresponding evaluation activities defined in the Supporting Document for the particular SFR.  For future versions of the cPPs it should be considered to add more specific references to RFCs in the SFRs. | 1 | General comment relating to RFCs has been added to section 6.1 |
|  | FCS\_CKM.4 | (RfI#3, summarized): FCS\_CKM.4 does make requirements on key destruction that might cause some trouble for products using certain types of technologies. | NIT sees the need to keep the wording for FCS\_CKM.4 technology neutral. NIT proposes to change the wording for FCS\_CKM.4 in V2 of ND cPP and FW cPP back to the FCS\_CKM.4 definition in NDPP V1.1 (incl. the evaluation activity). Since this definition covers also CSPs, there would be no need to introduce FDP\_RIP.1 in V2 of the cPPs. It should be attempted to improve the wording for FCS\_CKM.4 in future versions of the cPPs.  Remark: NIT is aware that NIT awaits another proposal for FCS\_CKM.4, but since it is not clear if that proposal will be provided in time and what the content will be in particular, NIT is upholding the original decision. | 1 | FCS\_CKM.4 (and its Evaluation Activities) has been rewritten for this version, and is subject to ongoing discussions for future versions of the cPP. The commenter is requested to review the updated FCS\_CKM.4 language and provide feedback accordingly. |
|  | FIA\_X509\_EXT.x | (RfI#10) NDcPP V1.0 requires several SFRs related to digital certificates including, FIA\_X509\_EXT.1, FIA\_X509\_EXT.2, FIA\_X509\_EXT.3.  However, it is possible for a product to only select cryptographic services that do not rely upon digital certificates. Specifically, if all remote communication (both with remote servers and remote administrators) use SSH with "ssh-rsa" and the TOE does not use digital certificates for trusted updates then the FIA\_X509\_EXT.1, FIA\_X509\_EXT.2, FIA\_X509\_EXT.3 requirements are not applicable to any required services provided by the TOE.  In this scenario, it does not make sense for the FIA\_X509\_EXT.1, FIA\_X509\_EXT.2, FIA\_X509\_EXT.3 requirements to be mandatory. | NIT recommends moving the FIA\_X509\_EXT SFRs to Annex B and make them selection-based requirements. This RfI requires input by the overall Network iTC to gather enough information about the different use cases and selection scenarios. | 2 | All X.509 SFRs have been moved to Appendix B, and are now selection-based requirements. |
|  | FCS\_SSHC\_EXT.1.2 | (RfI#12) FCS\_SSHC\_EXT.1.2 has been changed to keep public key authentication as mandatory but password authentication has been made optional. This is in agreement with the corresponding NIT Technical recommendation. Unfortunately, the SD has not been updated accordingly, though comments are placed in the corresponding sections. | Update also SD accordingly | 1 | Accepted and the SD has been updated to reflect the fact that password-based authentication is now optional. |
|  | FCS\_IPSEC\_EXT.1.3 | (RfI#14) For FCS\_IPSEC\_EXT.1.3 tunnel mode and transport mode have been made selectable which is in agreement with the corresponding NIT Technical recommendation. For administration of the TOE transport mode should be mandated. | Add application note to mandate transport mode for TOE administration. | 2 | No change.  Currently there does not seem to be a clear reason to demand transport mode for administration. This can be re-examined as part of the public review. |
|  | Protection Profile v1.0.3, Section 6.6.2,  FMT\_MTD.1 Management of TSF Data | “CoreData” is not defined anywhere and appears to be a superficial label. However, any change of this nature would make this SFR refined. | Avoid unnecessary modifications to Part 2 SFRs. | 1 | Added application notes to FMT\_MTD.1/CoreData and FMT\_MTD.1/CryptoKeys to clarify label usage |
|  | Protection Profile v1.0.3, Section 6.7.3 FPT\_TST\_EXT.1 TSF testing | Application Note 30 states:  “For distributed TOEs all TOE components have to perform self-tests.”  This language (“have to”) mandates performing of self-tests for “all TOE components”. This is does not align with actual SFR wording or how it is applied to non-distributed TOEs. | Clarify Application Note language to better align it with SFR. If mandatory component testing is intended new requirement, consider changing SFR wording to make this clearer. | 1 | Added additional explanatory paragraph to Application Note 30 which provides additional clarity over the requirement. |
|  | Protection Profile v1.0.3, Section 6.7.5, FPT\_STM.1 Time Stamps | Application Note 38 states:  “For distributed TOEs it is expected that the administrator ensures synchronization between the time settings of different TOE components.”  This application note would likely be misinterpreted to argue that MANUAL time settings is acceptable. | Revise application note to clarify that automatic synchronization between components is necessary. | 2 | Application note 38 reworded to clarify the requirement. |
|  | Protection Profile v1.0.3,  Section 6.5.1 FIA\_AFL.1 | Authentication Failure Management is much needed addition to NDcPP. Unfortunately, FIA\_AFL.1.2 as written will lead to usability issues.  Administrative Usernames are not presently treated as sensitive and often follow organization-mandated conventions (e.g. FirstLastname). Consequently, for many organizations it is possible to compile comprehensive username list as part of reconnaissance phase. As a result, it is very common to see the logs of dictionary attacks against administrative usernames for any outside-facing management interface. Without option to do IP blocking FIA\_AFL will lead to frequent administrative account lockouts, possibly making system difficult to administer. | I believe IP blocking must be allowed in addition to account blocking or end-users will simply turn off such feature to prevent frequent administrative lockouts. | 1 | No change. The addition of IP based blocking with FIA\_AFL requires further discussion by the wider iTC and the commenter is advised to raise this for discussion as part of a future version. |
|  | *FPT\_STM.1* | *“The TSF does not provide reliable information about the current time at the TOE’s location by itself…”*  This application note is problematic because it enables arguments that hardware real-time clock is unnecessary. | Add explicit language stating that real-time clock functionality is expected, and that distributed TOE components are expected to implement local real-time clocks. | 2 | Application note 38 reworded to clarify the requirement. |
|  | *FCS\_CKM.4* | The current wording for FCS\_CKM.4 defines different requirements for different storage technologies. The wording should be revised to be more technology neutral. | Modify FCS\_CKM.4 to be more technology neutral. | 1 | FCS\_CKM.4 (and its Evaluation Activities) has been rewritten for this version, and is subject to ongoing discussions for future versions of the cPP. The commenter is requested to review the updated FCS\_CKM.4 language and provide feedback accordingly. |
|  | NDcPP: FCS\_IPSEC\_EXT.1  Application Note 60 | Application Note 60 currently make forward looking statements regarding IKEv2 | Concurred with the proposed removal of “IKEv2 will be required for those TOEs entering evaluation after Quarter 3, 2016.” | 2 | Forward looking statement removed from App note 60 |
|  | NDcPP: FCS\_IPSEC\_EXT.1  Application Note 68 | Application Note 68 currently make forward looking statements regarding ECDSA | Concurred with the proposed removal of “Peer authentication using ECDSA X.509v3 certificates will be required for TOEs entering evaluation after Quarter 3, 2015.” | 2 | Forward looking statement removed from App note 68 |
|  | NDcPP: FCS\_SSHS\_EXT.1.5  Application Note 73 & 79 | There is a forward looking statement in Application Notes 73 and 79 (Relating to FCS\_SSHC\_EXT.1.5 and FCS\_SSHS\_EXT.1.5). These applications notes both contained the following statement "Future versions of this profile may remove ssh-rsa as a selection". | Concurred with the proposed removal of “Future versions of this profile may remove ssh-rsa as a selection” | 2 | Forward looking statement removed from App notes 73 and 79 |
|  | NDcPP: FCS\_SSHS\_EXT.1.5 & FCS\_SSHC\_EXT.1.5 | FCS\_SSHS\_EXT.1.5 and FCS\_SSHC\_EXT.1.5 both seemed to be missing selections for nistp521 curves | Concurred with the proposed additional of “-nistp521” curves | 2 | Nistp521 curves added |
|  | NDcPP: FCS\_CKM.1.1 Application Note 8 | Application Note 8 states that "Implementations according to FIPS 186-2 shall be accepted for products entering evaluation no later than December 31, 2015. Implementations according to FIPS 186-3 shall also be accepted but may be discontinued in a later version of this document." Clearly this is statement is out of date and we would welcome proposals on how this should be updated | FCS\_CKM.1 states that implementations must meet the latest standards in FIPS PUB 186-4. IAD recommend requiring compliance with these latest standards and removing the last paragraph of Application Note 8 which states, " FCS\_CKM.1 mandates compliance to FIPS 186-4. Implementations according to FIPS 186-2 shall be accepted for products entering evaluation no later than December 31, 2015. Implementations according to FIPS 186-3 shall also be accepted but may be discontinued in a later version of this document." | 2 | Outdated reference to old standards have been removed. |
|  | NDcPP: FIA\_X509\_EXT.1.1/Rev | In version 1 of the cPP FIA\_X509\_EXT.1.1 referenced RFC 5759 in relation to CRLs. RFC 5759 appears to be a specific profile for SuiteB and so the editorial team has chosen to add RFC 5280 in to the selection. | Concurred with the addition of the RFC 5280 to the selection | 2 | RFC 5280 added to selection for FIA\_X509\_EXT.1.1/Rev and /ITT |
|  | NDcPP: 3.1 Support Distributed TOE Use Cases | Is there a specific requirement for the Management Component to be either a hardware, software, or virtual-based device? | Specify any specific types of Management Component or provide a definition of a Management Component | 2 | No change. The current version 2 NDcPP does not support software only or virtualised components. Therefore the management component must be a hardware appliance. |
|  | NDcPP: 3.1 Case3, 3rd paragraph | It is unclear the word “similar” Network Devices means as appose to identical Network Devices. | Need to elaborate or define the word “similar” | 2 | Repeat of #37. The term ‘similar’ has been removed from the description. |
|  | NDcPP: Table 1 - FAU\_STG.3/LocSpace, Description column | “warning” is misspelled in “Display wanring for local storage space” | Replace with “warning” | 3 | Typo corrected |
|  | NDcPP: Table 1 - FAU\_STG.3/LocSpace, Description column | “Identification” is misspelled. “User Identificiation and Authentication” | Replace with “Identification” | 3 | Typo corrected |
|  | NDcPP: A.1 Audit Events for Optional SFRs – Table 4, Auditable Events | “Management of of cryptographic keys. | Remove the extra “of” | 3 | Extra word removed |
|  | NDcPP: A.2.1.3 FAU\_STG.3/LocSpace | “Action in case of possible audit data lossDisplay warning for local storage space” | Insert space between “loss” and “Display” | 3 | Space added |
|  | NDcPP:  FIA\_X509 | Selections can be made such that X509 certificates are not used by the network device.  For example, if all remote communication (both with remote servers and remote administrators) use SSH with "ssh-rsa" and the TOE does not use digital certificates for trusted updates then the FIA\_X509\_EXT.1, FIA\_X509\_EXT.2, FIA\_X509\_EXT.3 requirements are not applicable to any required services provided by the TOE. | Either move the X509 requirements to the selection based section or mandate certificate usage for all NDs. | 1 | All X.509 SFRs have been moved to Appendix B, and are now selection-based requirements. |
|  | NDcPP Section 3.4 | Inconsistency in terms (spelling error) -- Feature Dependant (‘Feature Dependent’) | Modify so consistent. | 3 | Consistency fixed to read ‘Dependent’ |
|  | NDcPP FCS\_CKM.1 | This statement in the Application Note indicates that this SFR is optional (depending on implementation details): “If the TOE acts as a receiver in the RSA key establishment scheme, the TOE does not need to implement RSA key generation.” | If this is the intent, then move SFR. Otherwise, remove statement and make this SFR mandatory for all NDs, no App Note exceptions. | 1 | Added clarifying reference within Application Note 8 to state that within a distributed TOE, if the TOE components acts as a receiver then it does not need to implement RSA key generation. |
|  | NDcPP and ND-SD: **FCS\_SSHS\_EXT.1.8** | There are other ways to meet the intent of this requirement besides sending over 2^28 packets to a device. For example, showing that a rekey occurs after a specified time period and justification that no more than 2^28 packets will be transmitted in that time. | Consider updating the SFR and/or Test in SD to accommodate other testing methods that still ensure the threat is mitigated. | 2 | As noted for comment ‎31 above, the test description has been updated as described in NIT Decision RfI#24. |
|  | **FAU\_GEN.1.1** | Application note 1, 3rd paragraph.  The statement is that each component must maintain a full audit record. The example given is that in a distributed TOE, failure to build up a secure communications channel should result in an audit event being generated by both components. This may be possible if both components know about each other, but it may be the case that one component attempts to contact the other and is prevented from doing so by an intervening firewall. In that case, only one component would know about the failure – the other component would be unaware that a failure had occurred. | Delete the example. Labs will latch onto examples like this and treat them as absolute truth. | 2 | Example in Application Note 1 has been updated to avoid misinterpretation. Example now reads “rejection of a connection by one component while attempting to establish a secure communication channel between two components should result in an audit event being generated by both components” |
|  | 6.4.1.3, FCS\_CKM.4 | The TSF shall destroy cryptographic keys… when? It does not specify when this needs to happen. | Add “when no longer needed”. | 3 | No change. Due to the changes of FCS\_CKM.4 more widely, it is suggested that the commenter raises this within the iTC for wider discussion. |
|  | 6.4.2.1 | AES-CBC is mandated here, but some popular vulnerability scanners (e.g. Nessus) commonly used by government entities will report vulnerabilities when CBC mode is used. | Not a suggested PP change – but suggest outreach (on behalf of the iTC) to the companies who build these scanners to encourage them not to report use of CBC as a vulnerability. | 0 | Noted. |
|  | Section *6.4.1.3 Pg. 53* | Requirement states: “For non-volatile flash memory, the destruction shall be executed by [selection: a single, direct overwrite consisting of zeroes, a block erase] followed by a read-verify.” | This is getting far to prescriptive and doesn’t belong in a PP. Perhaps it might in a supporting document that can be updated more frequently as new technologies come up.  A “read-verify” requirement eliminates basically all modern solid-state flash devices with wear-levelling as wear-levelling in the controller should preclude access to the same block by a subsequent read. For wear-levelled devices, either full device encryption or trusting that the wear-levelling prevents subsequent access to the keys are really the only viable options. Full disk encryption should be positively verifiable and just as with other vulnerabilities, labs should be always watching for attacks against wear-levelling. It should also be possible to test a particular flash device outside of the TOE to determine if wear-levelling properly prevents access to previously written and unallocated blocks (e.g. select two given known block values, write the first value to every allocatable block in the device’s file-system as one huge file; delete the file; start a loop where you allocate a block, attempt to read it and check for the known values, write a random value (to defeat “smart” write prevention techniques) then write the second value, end loop. If you ever read one of your known values, you can be sure that the device’s wear levelling is not preventing reads of the contents of unallocated blocks.) Labs should probably also ask for API documentation for the flash based devices to ensure no “debug” reading capability exists, although that might exceed the assurance level intended for PP’s. | Significant | FCS\_CKM.4 will has undergone a significant re-write. This re-write removes the read-verify requirement. |
|  | Section 6.7.4.1 Note 37; Pg. 67 | Note states: “All discrete software components (e.g. applications, drivers, kernel, firmware) of the TSF need to be protected, i.e. they should either be digitally signed by the corresponding manufacturer and subsequently verified by the mechanism performing the update or a hash should be published for them which needs to be verified before the update” | This note seems to imply that each component needs to be hashed/signed separately. Many, if not most, developers would choose to protect a whole update archive or other collection of components delivered together in a single package as one unit. That should explicitly be allowed. | Significant | No change.  (However, note that “components” in the quoted text has been replaced by elements, to avoid confusion with components of a distributed TOE.)  The requirement does not specify the granularity of the signatures, and this is not currently seen as necessary to add. Note however that all components of a distributed TOE need to support trusted update, and where digital signatures are used these must be checked by each component. |
|  | Section 6.7.4.1 Note 37; Pg. 67 | Note states: “it is essential that the update process verify that both the update and NV images were produced by the same manufacturer (e.g. by comparing public keys) or signed by legitimate signing keys (e.g. successful verification of certificates when using X.509 certificates).” | This note seems to both require signing rather than hashing as well as to preclude the possibility of recovering a device once it has had an improper flash image or corruption of a given image installed. I’m not even sure how the running image would access the public key/certificate associated with the private key used to sign the image or image archive previously installed, unless that public key/certificate was installed in the image. This seems like an additional requirement that installed images must include the public key or certificate associated with the private key used to sign the image. Previous requirements seem to allow images to not contain the public key/certificate or just be hashed (without being signed). I think this note should only ensure that the new image was properly signed or hashed. | Significant | Application Note 37 updated and the highlighted sentence has been removed. |
|  | Section 6.7.5.1 Note 38; Pg. 67-68 | Note states: “The TSF does not provide reliable information about the current time at the TOE’s location by itself, but depends on external time and date information, either provided manually by the administrator or through the use of an NTP server. The term ‘reliable time stamps’ refers to the strict use of the time and date information, that is provided externally, and the logging of all changes to the time settings including information about the old and new time. With this information the real time for all audit data can be calculated.  For distributed TOEs it is expected that the administrator ensures synchronization between the time settings of different TOE components. All TOE components shall either be in sync or the timezone offset should be known to the administrator for every pair of TOE components.” | The note seems to require that the TOE depend on an external time source. It is certainly possible that a TOE might have an internal reliable time source that might need to be initialized by an administrator, but from there on is reliable.  “With this information the real time for all audit data can be calculated” … and logged?  “All TOE components shall either be in sync or the timezone offset should be known to the administrator for every pair of TOE components” – This seems needlessly prescriptive. The requirement is that audit (and probably all other logging) must include a reliable real time stamp that can be used to determine the exact timing of audit events by an external observer. How that’s to be done should be left up to the developer and customer.  Also, we should probably note that where NTP \*is\* used, it is desired that the clients authenticate the servers and that FPT\_ITC, FTP, and FCS requirements do not apply to NTP, since to my knowledge, there is no supported NTP over TLS time-source available, although HTTPS connections can be used to constrain the un-encrypted NTP time sources. | Significant | In response to #48 and #50, Application Note 38 has been re-written to provide greater clarity on the subject of time management. The commenter is requested to review updated Application Note and provide further feedback if this doesn’t address the core concern raised. Points relating to the use of NTP should be raised as part of the version 3 discussions |
|  | Section 6.9.1.1 Pg. 69 | FTP\_ITC.1.1 The TSF shall be capable of using [selection: IPsec, SSH, TLS, HTTPS] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [selection: authentication server, assignment: [other capabilities]] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data. | We seem to be edging again into overly prescriptive verbiage here. What if a desired protocol built-in encryption using supported ciphers, key exchange, and authentication methods? I think we should at least allow the possibility to evaluate it. Do we need a “FCS\_GENERAL…” set of SFR’s? Might actually make the PP shorter. How about DTLS? | Moderate | No action. The iTC currently has no plan to consider vendor proprietary protocols in support of FTP\_ITC.1.1 or FPT\_ITT.1. Where there is a need to include additional, standard protocols such as DTLS, this will be considered but not part of the editorial process. The requestor should add DTLS in to the iTC roadmap document. |
|  | Section B2.1.3, FCS\_SSHC\_EXT.1.3; Pg. 93 Mirrored in Section B.2.1.4, FCS\_SSHS\_EXT.1.3. | FCS\_SSHC\_EXT.1.3 states: “FCS\_SSHC\_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: number of bytes] bytes in an SSH transport connection are dropped.”  FCS\_SSHS\_EXT.1.3 states: “FCS\_SSHS\_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: number of bytes] bytes in an SSH transport connection are dropped.” | RFC4253 has no requirement that packets longer than any given size be dropped. The word “dropped” is not used anywhere in RFC4253.  It does specify near and under “Maximum Packet Length”: “The minimum size of a packet is 16 (or the cipher block size, whichever is larger) bytes (plus 'mac'). Implementations SHOULD decrypt the length after receiving the first 8 (or cipher block size, whichever is larger) bytes of a packet.” and “All implementations MUST be able to process packets with an uncompressed payload length of 32768 bytes or less and a total packet size of 35000 bytes or less (including 'packet\_length', 'padding\_length', 'payload', 'random padding', and 'mac'). The maximum of 35000 bytes is an arbitrarily chosen value that is larger than the uncompressed length noted above. Implementations SHOULD support longer packets, where they might be needed. For example, if an implementation wants to send a very large number of certificates, the larger packets MAY be sent if the identification string indicates that the other party is able to process them.”  So a minimum packet size is defined, a required minimal implemented maximum compressed packet size of 32768 bytes is defined (with a correlated required minimal implemented maximum total packet size of 35000 bytes). There is only a recommendation that implementations: “SHOULD check that the packet length is reasonable in order for the implementation to avoid denial of service and/or buffer overflow attacks”  Where exactly are the FCS\_SSHC\_EXT.1.3 and FCS\_SSHS\_EXT.1.3 requirements coming from? Should it just be stated that implementations must document their maximum supported packet length they will negotiate and their behaviour when that length is exceeded or can we document some other standard where this requirement is stated? | Moderate | No change.  Whilst agreeing about the RFC statements, a decision has been made to be more specific in the cPP, requiring there to be a developer-specified maximum packet size, and for the action to be dropping packets above this size. In the absence of practical and implementation-based issues (and noting that this same requirement has also been used in NDPP), there seems no reason to change this requirement in the cPP. |
|  | Section B.2.1.5; FCS\_TLSC\_EXT.1.1; Pg 97 | FCS\_TLSC\_EXT.1.1, FCS\_TLSC\_EXT.2, FCS\_TLSS\_EXT.1.1, and FCS\_TLSS\_EXT.2.1 seem to be missing a few cipher-suites that would normally be included in the current set of allowed cipher-suites. Namely:  ECDHE-RSA-AES256-SHA384 & ECDHE-RSA-AES128-SHA256 from RFC5289  and  AES256-GCM-SHA384 and AES128-GCM-SHA256 are included in RFC5288 | Also include these ciphers? The current list makes cipher-suite selection strings needlessly large. For example, with openssl, you need to explicitly list every desired cipher-suite string, which make the full cipher—string larger than many product can support, so supported ciphers would have to be left off.  Alternatively, if these ciphers were added, then the full openssl cipher-string that would include only the list of desired ciphers would be: “EECDH+AES:RSA+AES:@STRENGTH”, which is simple, short, and give exactly the desired cipher suites in descending strength order. |  | Proposed additional cipher-suites have been added to the selections for the highlighted SFRs. |
|  | *Section 3.3* | There is a requirement to check the EKU of TLS certificates for ServerAuth or ClientAuth depending on if the entity is a server or a client (FIA\_X509\_EXT.1.1). This is fine for communicating with external entities such as an audit server or authentication server where the entity is always a TLS server OR TOE that supports mutual authentication of external web users where the entities will always be TLS client. However, for TOE components of some products, they can be TLS server or client based on the configuration or deployment OR be a TLS server on one connection and a TLS client on another connection simultaneously. For example, HA or clustering. While the solution may be simple (use two certificates or have one certificate with both ServerAuth and ClientAuth in the EKU), the question is this really necessary? For example, in section 3.3 it states “Choose FPT\_ITT.1 if certificate revocation is not performed.” Do we need to check the server/client relationship for an internal trusted communication where the device certificates are created/maintained by the vendor? For which mutual authentication is supported so all devices certificate are validated regardless if they are TLS client or server.  The problem is really for product upgrade. If the product did not check for these EKU values and now it must, all previous version of the products must have the TOE components de-register, re-generate all the device certificates with the correct EKU values, and re-register back to the Manager or Gatekeeper. For customers with large deployment, this will not make them happy. While this is doable, it’s not sure what additional security benefits do checking which TOE components are server/client will add. At the same time, it is negatively affecting the customers experience. This requirement makes sense for external communication where the relationship is fixed. However, for Distributed TOE, this checks are not needed and may require device certificates to be re-generated during upgrade. | Make checking EKU optional like checking revocation status for Distributed TOE only. | 1 | The editorial group has reviewed and discussed this feedback with the ITT sub group and the conclusion is that no change will be made to the cPP. The group believe that the cPP should be encouraging conscious design of certificate infrastructure and that this should include the identification of keys to meet specific requirements. |