Supporting Document Mandatory Technical Document: Evaluation Activities for collaborative Protection Profile Module for Server Applications

Foreword

This is a Supporting Document, intended to complement the Common Criteria (CC) version 3 and the associated Common Evaluation Methodology for Information Technology Security Evaluation.

Supporting Documents may be "Guidance Documents", that highlight specific approaches and application of the standard to areas where no mutual recognition of its application is required, and as such, are not of normative nature, or "Mandatory Technical Documents", whose application is mandatory for evaluations whose scope is covered by that of the Supporting Document. The usage of the latter class is not only mandatory, but certificates issued as a result of their application are recognized under the CCRA.

This Supporting Document has been developed by the iTC for Application Software iTC and is designed to be used to support the evaluations of TOEs against the cPP identified in Section 1.1, “Technology Area and Scope of Supporting Document”.

Acknowledgements

This Supporting Document was developed by the iTC for Application Software international Technical Community with representatives from industry, Government agencies, Common Criteria Test Laboratories, and members of academia.

Revision History

Table 1. Revision history

Version	Date	Description
1.0	2022-04-06	Initial Release
1.0e	2024-02-15	Incorporated feedback received following initial release.

General Purpose

See Section 1.1, “Technology Area and Scope of Supporting Document”.

Field of special use

This Supporting Document applies to the evaluation of TOEs claiming conformance with the collaborative PP-Module for Server Applications.

Table of Contents

1. Introduction

1.1. Technology Area and Scope of Supporting Document

This Supporting Document (SD) is mandatory for evaluations of products that claim conformance to any of the following cPP(s):

- collaborative PP-Module for Server Applications, Version 1.1, 2022-08-16

Although Evaluation Activities (EAs) are defined mainly for the evaluator to follow, the definitions in this SD aim to provide a common understanding for developers, evaluators and users as to what aspects of the TOE are tested in an evaluation against Collaborative Protection Profile for Application Software, and to what depth the testing is carried out. This common understanding in turn contributes to the goal of ensuring that evaluations against Collaborative Protection Profile for Application Software achieve comparable, transparent and repeatable results. In general, the definition of EAs will also help developers to prepare for evaluation by identifying specific requirements for their TOE. The specific requirements in EAs may in some cases clarify the

meaning of SFRs, and may identify particular requirements for the content of Security Targets (STs) (especially the TOE Summary Specification (TSS)), AGD guidance, tests, and possibly required supplementary information (e.g. *any examples, such as for entropy analysis or cryptographic key architecture*).

1.2. Structure of the Document

EAs can be defined for both SFRs and SARs. These are defined in separate sections of this SD.

If any EA cannot be successfully completed in an evaluation then the overall verdict for the evaluation is a 'fail'. In rare cases there may be acceptable reasons why an EA may be modified or deemed not applicable for a particular TOE, but this must be agreed with the Certification Body for the evaluation.

In general, if all EAs (for both SFRs and SARs) are successfully completed in an evaluation then it would be expected that the overall verdict for the evaluation is a 'pass'. To reach a 'fail' verdict when the EAs have been successfully completed would require a specific justification from the evaluator as to why the EAs were not sufficient for that TOE.

Similarly, at the more granular level of Assurance Components, if the Evaluation Activities for an Assurance Component and all of its related SFR Evaluation Activities are successfully completed in an evaluation then it would be expected that the verdict for the Assurance Component is a 'pass'. To reach a 'fail' verdict for the Assurance Component when these Evaluation Activities have been successfully completed would require a specific justification from the evaluator as to why the Evaluation Activities were not sufficient for that TOE.

2. Evaluation Activities for SFRs

2.1. Structure of EAs

All EAs for SFRs defined in this Section include the following items to keep consistency among EAs.

1. Objective of the EA

   Objective defines the goal of the EA. Assessment Strategy describes how the evaluator can achieve this goal in more detail and Pass/Fail criteria defines how the evaluator can determine whether the goal is achieved or not.

2. Dependency

   Where the EA depends on completion of another EA then the dependency and the other EA is also identified here.

3. Tool types required to perform the EA

   If performing the EA requires any tool types in order to complete the EA then these tool types are defined here.

4. Required input from the developer or other entities

   Additional detail is specified here regarding the required format and content of the inputs to the EA.

5. Assessment Strategy

   Assessment Strategy provides guidance and details on how to perform the EA. It includes, as appropriate to the content of the EA;

   a. How to assess the input from the developer or other entities for completeness with respect to the EA
   b. How to make use of any tool types required (potentially including guidance for the calibration or setup of the tools)
   c. Guidance on the steps for performing the EA

6. Pass/Fail criteria

   The evaluator uses these criteria to determine whether the EA has demonstrated that the TOE has met the relevant requirement or that it has failed to meet the relevant requirement.

7. Requirements for reporting

   Specific reporting requirements that support transparency and reproducibility of the Pass/Fail judgement are defined here.

2.2. Justification for EAs for SFRs

EAs in this SD provide specific or more detailed guidance to evaluate the *type of* system, however, it is the CEM work units based on which the evaluator shall perform evaluations.

This Section explains how EAs for SFRs are derived from the particular CEM work units identified in Assessment Strategy to show the consistency and compatibility between the CEM work units and EAs in this SD.

Assessment Strategy for ASE_TSS requires the evaluator to examine that the TSS provides sufficient design descriptions and its verdicts will be associated with the CEM work unit ASE_TSS.1-1. Evaluator verdicts associated with the supplementary information will also be associated with ASE_TSS.1-1, since the requirement to provide such evidence is specified in ASE in the cPP.

Assessment Strategy for AGD_OPE/ADV_FSP requires the evaluator to examine that the AGD guidance provides sufficient information for the administrators/users as it pertains to SFRs, its verdicts will be associated with CEM work units ADV_FSP.1-7, AGD_OPE.1-4, and AGD_OPE.1-5.

Assessment Strategy for ATE_IND requires the evaluator to conduct testing that the iTC has determined that those testing of the TOE in the context of the associated SFR is necessary. While the evaluator is expected to develop tests, there may be instances where it is more practical for the developer to construct tests, or where the developer may have existing tests. Therefore, it is acceptable for the evaluator to witness developer-generated tests in lieu of executing the tests. In this case, the evaluator must ensure the developer’s tests are executing both in the manner

declared by the developer and as mandated by the EA. The CEM work units that derive those EAs are: ATE_IND.1-3, ATE_IND.1-4, ATE_IND.1-5, ATE_IND.1-6, and ATE_IND.1-7.

2.3. Security Management (FMT)

2.3.1. Supported Configuration Mechanism (FMT_MEC_EXT)

2.3.1.1. FMT_MEC_EXT.1.1/Server

2.3.1.1.1. TSS

The evaluator shall review the TSS to identify where the application's configuration data is stored. The evaluator shall also verify that the TSS identifies who has read and write access to the configuration data.

2.3.1.1.2. Operational Guidance

No activities specified.

2.3.1.1.3. Test

The evaluator shall run the following tests:

- Test 1: The evaluator shall verify that the access rules for the configuration files align with the read and write access identified in the TSS.
- Test 2: The evaluator shall run the application while monitoring it with the following platform specific tools and make changes to its configuration. The evaluator shall verify that the tool logs show corresponding changes to the locations identified in the TSS for storage of configuration data. The following platform specific tools and procedures must be used:
  - Windows: SystInternal tool ProcMon
    - The evaluator shall run the application while monitoring it with the SysInternal tool ProcMon and make changes to its configuration. The evaluator shall verify that ProcMon logs show corresponding changes to the locations identified in the TSS for storage of configuration data.
  - Linux or macOS: strace (or equivalent utility)
    - The evaluator shall run the application while monitoring it with the utility strace. The evaluator shall make security-related changes to its configuration. The evaluator shall verify that strace logs corresponding changes to configuration files that reside in /etc (for system-specific configuration) or in the user's home directory (for user-specific configuration).

2.3.2. Specification of Management Functions (FMT_SMF)

2.3.2.1. FMT_SMF.1.1/Server

2.3.2.1.1. TSS

No activities specified.

2.3.2.1.2. Operational Guidance

The evaluator shall verify that every management function specified in the SFR is described in the operational guidance. If multiple management interfaces are supported, the guidance documentation must describe which interfaces may be used to perform the management functions.

2.3.2.1.3. Test

The evaluator shall perform the following test:

- Test 1: The evaluator shall test the application's ability to provide each management function by configuring the application and testing each function specified. The evaluator is expected to test these functions in all the ways in which the ST and guidance documentation state the configuration can be managed. Each function should be tested on each management interface on which the functionality is supported.

2.4. Protection of the TSF (FPT)

2.4.1. Anti-Exploitation Capabilities (FPT_AEX_EXT)

2.4.1.1. FPT_AEX_EXT.2.1/Server

2.4.1.1.1. TSS

No activities specified.

2.4.1.1.2. Operational Guidance

No activities specified.

2.4.1.1.3. Test

The evaluator shall configure the platform in the ascribed manner and carry out one of the prescribed tests:

- Test 1: [conditional] If the application is being tested on Windows, the evaluator shall ensure that the application can run successfully with Windows Defender Exploit Guard Exploit Protection configured with the following minimum mitigations enabled; Control Flow Guard (CFG), Randomize memory allocations (Bottom-Up ASLR), Export address filtering (EAF), Import address filtering (IAF), and Data Execution Prevention (DEP). The following link describes how to enable Exploit Protection, https://docs.microsoft.com/en-us/windows/security/threat-protection/microsoft-defender-atp/enable-exploit-protection.
- Test 2: [conditional] If the application is being tested on Linux, the evaluator shall ensure that the application can successfully run on a system with SELinux (or equivalent platform vendor recommended security features) enabled and enforcing.
- Test 3: [conditional] If the application is being tested on macOS, the evaluator shall ensure that

the application can successfully run on a system without disabling System Integrity Protection (SIP).

3. Evaluation Activities for Selection-Based Requirements

3.1. Communication (FCO)

3.1.1. Component Registration Channel Definition (FCO_CPC_EXT.1/Server)

3.1.1.1. FCO_CPC_EXT.1/Server

3.1.1.1.1. TSS

The evaluator shall examine the TSS to confirm it:

- Describes the method by which a Security Administrator enables and disables communications between pairs of TOE parts.
- Describes the relevant details according to the type of channel in the main selection made in FCO_CPC_EXT.1.2/Server:
  - First type: the TSS identifies the relevant SFR iteration, if present, that specifies the channel used.
  - Second type: the TSS describes details of the channel and the mechanisms that it uses.

3.1.1.1.2. Operational Guidance

The evaluator shall examine the guidance documentation to confirm that it contains instructions for enabling and disabling communications with any individual parts of the TOE. The evaluator shall confirm that the method of disabling is such that all other TOE parts can be prevented from communicating with the part that is being removed from the TOE (preventing the remaining parts from either attempting to initiate communications to the disabled part, or from responding to communications from the disabled part).

The evaluator shall examine the guidance documentation to confirm that it includes recovery instructions should a connection be unintentionally broken during the registration process.

If the TOE uses a registration channel for registering components to the TOE (i.e. where the ST author uses the FPT_ITT.1/Server in the selection for FCO_CPC_EXT.1.2/Server) then the evaluator shall examine the Preparative Procedures to confirm that they:

- Describe the security characteristics of the registration channel (e.g. the protocol, keys and authentication data on which it is based).
- Identify any dependencies between the configuration of the registration channel and the security of the subsequent intra-TOE communications (e.g. where AES-256 intra-TOE communications depend on transmitting 256 bit keys between TOE parts and therefore rely on the registration channel being configured to use an equivalent key length).

- Identify any aspects of the channel can be modified by the operational environment in order to improve the channel security and shall describe how this modification can be achieved (e.g. generating a new key pair, or replacing a default public key certificate).

As background for the examination of the registration channel description, it is noted that the requirements above are intended to ensure that administrators can make an accurate judgement of any risks that arise from the default registration process. Examples would be the use of self-signed certificates (i.e. certificates that are not chained to an external or local Certification Authority), manufacturer-issued certificates (where control over aspects such as revocation, or which devices are issued with recognised certificates, is outside the control of the operational environment), use of generic/non-unique keys (e.g. where the same key is present on more than one instance of a device), or well-known keys (i.e. where the confidentiality of the keys is not intended to be strongly protected – note that this does not imply there is a positive action or intention to publicise the keys).

3.1.1.1.3. Test

The evaluator shall carry out the following tests:

- Test 1.1: The evaluator shall confirm that an Agent application that is not currently a member of the TOE cannot communicate with any part of the TOE until the non-member entity is enabled by a Security Administrator for each of the non-equivalent TOE part with which it is required to communicate.
- Test 1.2: The evaluator shall confirm that after enablement, an Agent application can communicate only with the part that it has been enabled for. This includes testing that the enabled communication is successful for the enabled pair, and that communication remains unsuccessful with any other part for which communication has not been explicitly enabled.

Some TOEs may set up the registration channel before the enablement step is carried out, but in such a case the channel must not allow communications until after the enablement step has been completed.

The evaluator shall repeat Tests 1.1 and 1.2 for each different type of enablement process that can be used in the TOE.

- Test 2: The evaluator shall separately disable each TOE part in turn and ensure that the other TOE parts cannot then communicate with the disabled part, whether by attempting to initiate communications with the disabled part or by responding to communication attempts from the disabled part.
- Test 3: The evaluator shall carry out the following tests according to those that apply to the values of the selection made in the ST for FCO_CPC_EXT.1.2/Server.
    - If the ST uses the first type of communication channel in the selection in FCO_CPC_EXT.1.2/Server then the evaluator tests the channel via the Evaluation Activities for FPT_ITT.1/Server.
    - If the ST uses the ‘no channel’ selection, then no test is required.
- Test 4 [conditional]: If *A channel that meets the secure channel requirements in FPT_ITT.1* is selected in FCO_CPC_EXT.1.2/Server, the evaluator shall perform one of the following tests, according to the TOE characteristics identified in its TSS and operational guidance:

- If the registration channel is not subsequently used for communication between TOE parts, then the evaluator shall confirm that the registration channel can no longer be used after the registration process has completed, by attempting to use the channel to communicate with each of the endpoints after registration has completed.
- If the registration channel is subsequently used for communication between TOE parts then the evaluator shall confirm that any aspects identified in the operational guidance as necessary to meet the requirements for a steady-state inter-part channel (as in FPT_ITT.1) can indeed be carried out (e.g. there might be a requirement to replace the default key pair and/or public key certificate).

3.2. Identification and Authentication (FIA)

3.2.1. Authentication using X.509 certificates (FIA_X509_EXT/Server)

3.2.1.1. FIA_X509_EXT.1.1/ITT/Server

3.2.1.1.1. TSS

The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1/ITT/Server) that are not supported by the TOE or Platform (i.e. where the ST is therefore claiming that they are trivially satisfied). If selected, the TSS shall describe how certificate revocation checking is performed. It is not sufficient to verify the status of a X.509 certificate only when it's loaded onto the TOE or Platform.

3.2.1.1.2. Operational Guidance

No activities specified.

3.2.1.1.3. Test

The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE or Platform. The evaluator shall perform the following tests:

- Test 1a: The evaluator shall load a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the certificate to be used in the function, and shall use this chain to demonstrate that the function succeeds.
- Test 1b: The evaluator shall then delete one of the certificates in the chain (i.e. the root CA certificate or other intermediate certificate, but not the end-entity certificate), and show that the function fails.
- Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.
- Test 3: [conditional] The evaluator shall test that the TOE or Platform can properly handle revoked certificates if CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the TOE certificate and

revocation of the TOE intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. No testing is required if no revocation method is selected.

- Test 4: [conditional] If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set, and verify that validation of the CRL fails.
- Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)
- Test 6: The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)
- Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The hash of the certificate will not validate.)

3.2.1.2. FIA_X509_EXT.1.2/ITT/Server

3.2.1.2.1. TSS

No activities specified.

3.2.1.2.2. Operational Guidance

No activities specified.

3.2.1.2.3. Test

The evaluator shall perform the following tests. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.1.1/ITT/Server. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE or Platform (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted.

The evaluator shall create a chain of at least two certificates: the node certificate to be tested, and the self-signed Root CA.

- Test 1: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate does not contain the basicConstraints extension. The validation of the certificate path fails.
- Test 2: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the CA flag in the basicConstraints extension set to FALSE. The validation of the certificate path fails.

- Test 3: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the CA flag in the basicConstraints extension set to TRUE. The validation of the certificate path succeeds.

3.3. Protection of the TSF (FPT)

3.3.1. Basic internal TSF data transfer protection (FPT_ITT.1/Server)

3.3.1.1. FPT_ITT.1.1/Server

3.3.1.2. TSS

247. The evaluator shall examine the TSS to ensure it describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied).

248. If selected, the TSS shall describe how certificate revocation checking is performed. It is expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication).

3.3.1.3. Guidance Documentation

249. The evaluator shall also ensure that the guidance documentation describes where the check of validity of the certificates takes place, describes any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) and describe how certificate revocation checking is performed.

3.3.1.4. Tests

FIA_X509_EXT.1.1/ITT

250. The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.:

    a. Test 1a: The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function and shall use this chain to demonstrate that the function succeeds. Test 1a shall be designed in a way that the chain can be 'broken' in Test 1b by either being able to remove the trust anchor from the TOEs trust store, or by setting up the trust store in a way that at least one intermediate CA certificate needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store).

       Test 1b: The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE's trust store used to terminate the chain, or by removing one of the

intermediate CA certificates (provided together with the leaf certificate in Test 1a) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.

b. Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.

c. Test 3: The evaluator shall test that the TOE can properly handle revoked certificates--–depending on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the TOE certificate and revocation of the TOE intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. No testing is required if no revocation method is selected. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore, the revoked certificate(s) used for testing shall not be a trust anchor.

d. Test 4: If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.

e. Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)

f. Test 6: The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

g. Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The hash of the certificate will not validate.)

h. Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The following tests are run when a minimum certificate path length of three certificates is implemented:

   Test 8a: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a valid chain of EC certificates (terminating in a trusted CA certificate), where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.

   Test 8b: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from

outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a chain of EC certificates (terminating in a trusted CA certificate), where the intermediate certificate in the certificate chain uses an explicit format version of the Elliptic Curve parameters in the public key information field, and is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

Test 8c: The evaluator shall establish a subordinate CA certificate, where the elliptic curve parameters are specified as a named curve, that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is accepted into the TOE's trust store. The evaluator shall then establish a subordinate CA certificate that uses an explicit format version of the elliptic curve parameters, and that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is rejected, and not added to the TOE's trust store.

FIA_X509_EXT.1.2/ITT

251. The evaluator shall perform the following tests for FIA_X509_EXT.1.2/ITT. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/ITT. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted.
252. The goal of the following tests is to verify that the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using basicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation).
253. For each of the following tests the evaluator shall create a chain of at least two certificates: a self-signed root CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain).
    a. Test 1: The evaluator shall ensure that one CA in the chain does not contain the basicConstraints extension. The evaluator confirms that the TOE rejects such a certificate at each of the following points supported: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).
    b. Test 2: The evaluator shall ensure that at least one of the CA certificates in the chain has a basicConstraints extension in which the CA flag is set to FALSE. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

4. Evaluation Activities for SARs

The PP-Module does not define any SARs beyond those defined within the App PP base to which it must claim conformance. It is important to note that a TOE that is evaluated against the PP-Module is inherently evaluated against this Base-PP as well. The Collaborative Protection Profile for Application Software includes a number of Evaluation Activities associated with both SFRs and SARs. Additionally, the PP-Module includes a number of SFR-based Evaluation Activities that similarly refine the SARs of the Base-PPs. The evaluation laboratory will evaluate the TOE against the Base-PP and supplement that evaluation with the necessary SFRs that are taken from the PP-Module.

5. References

- [CC1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017.
- [CC2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017.
- [CC3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017.
- [CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, CCMB-2017-04-004, Version 3.1 Revision 5, April 2017.
- [addenda] CC and CEM addenda, Exact Conformance, Selection-Based SFRs, Optional SFRs, Version 0.5, May 2017
- [cPP] Collaborative Protection Profile for Application Software, Version 1.1, 2022-08-16