collaborative PP-Module for Server Applications

Acknowledgements

This collaborative Protection Profile Module (PP-Module) was developed by the iTC for Application Software international Technical Community (iTC) also known as AppSW-iTC 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.

Table of Contents

Preface

This module is part of a Collaborative Protection Profile (cPP) whose Target of Evaluation (TOE) is software applications. Under this cPP software applications can be categorized under the following broad categories:

1. Enterprise Server Applications

This cPP-Module is to be used against which all of the above categories of software applications may be evaluated. In addition there are PP-Modules that may be applicable based on the category of application.

In addition to the above categories there are large number of applications (Desktop and Mobile) that fall under “Consumer-grade” category. While such applications could be evaluated under the Application Software cPP, it is not the intention of this iTC to specifically address this category. The iTC doesn’t believe the consumer grade app ecosystem would support the historical cost and timelines associated with a Common Criteria evaluation.

Objectives of Document

This document presents the Common Criteria (CC) collaborative Protection Profile Module (PP-Module) to express the security functional requirements (SFRs) and security assurance requirements (SARs) for Enterprise Server Applications. The Evaluation activities that specify the actions the evaluator performs to determine if a product satisfies the SFRs captured within this PP-Module, are described in [SD].

Scope of Document

The scope of the PP-Module within the development and evaluation process is described in the Common Criteria for Information Technology Security Evaluation. In particular, a PP-Module defines the IT security requirements of a generic type of TOE and specifies the functional security measures to be offered by that TOE to meet stated requirements [[CC1], Section B.14].

Intended Readership

The target audiences of this PP-Module are developers, CC consumers, system integrators, evaluators and schemes.

Although the PP-Module and SD may contain minor editorial errors, the PP-Module is recognized as living document and the iTC is dedicated to ongoing updates and revisions. Please report any issues to the AppSW-iTC.

Related Documents

- [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.
- [SD] Supporting Document Mandatory Technical Document: Evaluation Activities for collaborative Protection Profile Module for Server Applications, Version 1.0, February 23, 2022

For more see the Common Criteria Portal.

1. PP-Module Introduction

1.1. PP-Module Reference Identification

- PP-Module Reference: collaborative PP-Module for Server Applications
- PP-Module Version: 1.0e
- PP-Module Date: 2024-02-15

1.2. TOE Overview

1.2.1. Compliant Targets of Evaluation

This is a Collaborative Protection Profile (cPP) Module whose Target of Evaluation (TOE) is Enterprise Server Applications. This PP-Module is compatible with the cPP for Application Software.

For a distributed TOE, the Server Application is the TOE component, or set of TOE components, that provides management, coordination, policy, API-facing, or other server-side functionality for the TOE. In a microservices architecture, a Server Application component may be a service or application payload that coordinates, exposes, or controls TOE functionality. The container orchestration platform, container runtime, operating system, service mesh infrastructure, ingress infrastructure, cluster networking, and platform-provided secret or configuration stores are part of the operational environment unless explicitly included in the TOE boundary.

1.3. TOE Use Cases

All use cases of Enterprise Server applications defined in the Collaborative Protection Profile for Application Software are applicable to this PP-Module.

1.4. Distributed and Microservices TOE Configurations

This PP-Module may be used in a PP-Configuration with the PP-Module for Agent Applications to evaluate distributed application software. Distributed application software includes server-agent deployments, clustered server deployments, and microservices architectures composed of multiple application payload components.

For a distributed TOE, the ST shall identify each TOE component, describe the role of each TOE component, identify which components implement each claimed SFR, and describe all communications between TOE components. The ST shall also distinguish TOE components from operational environment components. Operational environment components may include container orchestration, container runtimes, operating systems, service mesh infrastructure, ingress infrastructure, cluster networking, platform-provided secret or configuration stores, and other infrastructure services not explicitly included in the TOE boundary.

If the TOE relies on operational environment components for execution, scheduling, networking, isolation, credential storage, configuration storage, time services, or protection of inter-component communications, the ST shall identify the dependency and the guidance shall describe the required environmental configuration.

2. CC Conformance Claims

As defined by the references [CC1], [CC2] and [CC3], this PP-Module:

- conforms to the requirements of Common Criteria v3.1, Revision 5,
- is Part 2 extended,
- does not claim conformance to any other security functional requirement packages.

In order to be conformant to this PP-Module, a ST shall demonstrate Exact Conformance. Exact Conformance, as a subset of Strict Conformance as defined by the CC, is defined as the ST containing all of the SFRs in Security Functional Requirements (these are the mandatory SFRs) of this PP-Module, and potentially SFRs from Consistency Rationale (these are selection-based SFRs) and Selection-Based Requirements (these are optional SFRs) of this PP-Module. While iteration is allowed, no additional requirements (from the CC parts 2 or 3, or definitions of extended components not already included in this PP-Module) are allowed to be included in the ST. Further, no SFRs in Security Functional Requirements of this PP-Module are allowed to be omitted.

2.1. Components allowed with this cPP in a PP-Configuration

The list of packages, PP-Modules and cPPs that may be used in conjunction with this Module can be found at: https://appswcpp.github.io/cPP/AppSW_cPP_allowed-with-list.pdf

3. Security Problem Definition

The security problem is described in terms of the threats that the TOE is expected to address, assumptions about its operational environment, and any organizational security policies that the TOE is expected to enforce.

3.1. Threats

3.1.1. T.LOCAL_ATTACK_SERVER

An attacker can act through unprivileged access on the same computing platform on which the application executes. For example, attackers may provide maliciously formatted input to the application in the form of files or other local communications thus providing unauthorized access to plaintext sensitive data.

SFR Rationale:

- FPT_AEX_EXT.2/Server ensures that the application does not subvert security mechanisms provided by the platform thereby allowing an attacker with local access to exploit the application.
- FMT_MEC_EXT.1/Server ensures that unauthorized access to application's configuration data is not possible.
- FMT_SMF.1/Server ensures that rogue or misconfigured TOE parts/agents do not compromise the security of the server application.

3.1.2. T.PLATFORM_UPDATE_SERVER

Updating the platform that the application operates on could break the application's functionality. As such an end user might choose not to update the platform, thereby preventing the patching of known issues on the platform. An attacker could exploit such unpatched vulnerabilities in the platform to then mount an attack on the application.

SFR Rationale:

- FPT_AEX_EXT.2/Server SFR ensures that the TOE leverages the functionality provided and supported by the platform. This ensures that when the platform is updated, the supported functionality does not break and makes it easier to keep the platform updated without having to worry about breaking the applications running on the platform.

3.1.3. T.UNTRUSTED_COMMUNICATION_CHANNELS_SERVER

Attackers may take advantage of poorly designed or non-secure protocols or poor key management to successfully perform man-in-the middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the application itself. Attackers may attempt to target applications that do not use standardized secure tunneling protocols to protect the critical network traffic. This threat is of particular concern when an application uses protocols that have not been subject to extensive peer

review. Additionally attackers may attempt to connect via untrusted entities and use that connectivity to perform various attacks.

SFR Rationale:

- FCO_CPC_EXT.1/Server SFR ensures that only trusted entities connect with each other.
- FPT_ITT.1/Server SFR ensures that the communication between trusted entities is secure using well known protocols.

3.1.4. Assumptions

All Assumptions of the Collaborative Protection Profile for Application Software apply also to this PP-Module.

3.1.5. Organizational Security Policies

There are no OSPs for applications.

4. Security Objectives

4.1. Security Objectives for the TOE

The following subsections describe objectives for the TOE. Since the Collaborative Protection Profile for Application Software does not specify any Objectives for the TOE. This section contains only additional Objectives for the TOE related to the PP-Module but independent from the Collaborative Protection Profile for Application Software.

4.1.1. O.WELL-BEHAVED_SERVER

The TOE shall not circumvent the security controls provided by the underlying platform.

SFR Rationale:

- FPT_AEX_EXT.2/Server ensures that the app is well-behaved within the narrow context of ensuring security mechanisms of the underlying platforms are not subverted.

4.2. Security Objectives for the Operational Environment

All objectives for the Operational Environment of the Collaborative Protection Profile for Application Software apply also to this PP-Module. Additionally the following objective is added to this PP-Module:

4.2.1. OE.SECURE_LOCATION_SERVER

Enterprise servers that run enterprise applications should be housed in a secure location.

5. Security Functional Requirements

5.1. Conventions

The individual security functional requirements are specified in the sections below. The following conventions are used for the completion of operations:

- [*Italicized text within square brackets*] indicates an operation to be completed by the ST author.
- **Bold text** indicates additional text provided as a refinement.
- [**Bold text within square brackets**] indicates the completion of an assignment.
- [text within square brackets] indicates the completion of a selection.
- Number in parentheses after SFR name, e.g. (1) indicates the completion of an iteration.
- '/Server' has been added to every SFR in this module to distinguish SFRs added from Server modules.

Extended SFRs are identified by having a label “EXT” at the end of the SFR name.

5.2. Security Management (FMT)

5.3. FMT_MEC_EXT.1/Server

FMT_MEC_EXT.1.1/Server Read and write access to the TOE’s configuration data shall be limited to Administrator, TOE and [*assignment: list of authorized entities*].

5.4. FMT_SMF.1/Server

FMT_SMF.1.1/Server The TSF shall be capable of performing the following management functions:

- configuration of communication with other trusted IT entities
- [*selection:*
  - *configuration of communication with other TOE components according to FCO_CPC_EXT.1/Server and FPT_ITT.1/Server*
  - *allow/disallow the enrollment of a TOE agent by administrative function or policy,*
  - *query agent version,*
  - *provide update functionality to agent,*
  - *change administrative passwords,*
  - *change agent credentials,*
  - *configure and change recovery credentials,*
  - *configure number of authentication attempts and failed authentication behavior,*
  - *[assignment: Other management functions]*]

Application Note 1: Functions that relate to management of agents or other separately deployed TOE components are intended to be used in conjunction with the Agent module. The same functions may also be used with third-party entities that are in the operational environment and are not within the TOE boundary.

5.5. Protection of the TSF (FPT)

5.5.1. FPT_AEX_EXT.2/Server

FPT_AEX_EXT.2.1/Server The application shall be compatible with security features provided by the platform vendor.

Application Note 2: This requirement is designed to ensure that platform security features do not need to be disabled in order for the application to run. The assignment in FPT_AEX_EXT.1.3 in the Collaborative Protection Profile for Application Software must be "no exceptions".

6. Security Assurance Requirements

This PP-Module does not define any additional assurance requirements above and beyond what is defined in the Collaborative Protection Profile for Application Software that it extends. Application of the SARs to the TOE boundary described by both the claimed base and this PP-Module is sufficient to demonstrate that the claimed SFRs have been implemented correctly by the TOE.

Appendix A: Selection-Based Requirements

These SFRs apply when the TOE includes separately deployed TOE parts that communicate with one another as part of a PP-Configuration that includes the Agent Module. For microservices architectures, these SFRs apply to the communication relationships between Server Application components and Agent Application components as those components are identified in the ST. The ST author should iterate these SFRs as needed for different component pairs or communication mechanisms.

A.1. Communication (FCO)

A.1.1. FCO_CPC_EXT.1/Server

FCO_CPC_EXT.1.1/Server The TSF shall require a Security Administrator to enable communications between any pair of TOE parts before such communication can take place.

FCO_CPC_EXT.1.2/Server The TSF shall implement a registration process in which TOE parts establish and use a communications channel that uses [*selection:*

- *A channel that meets the secure channel requirements in FPT_ITT.1,*
- *No channel*

].

Application Note 3: "No channel" is selected if the component registration is performed via out-of-band manual means.

FCO_CPC_EXT.1.3/Server The TSF shall enable a Security Administrator to disable communications between any pair of TOE parts.

A.2. Identification and Authentication (FIA)

A.2.1. FIA_X509_EXT.1/ITT/Certificate Validation/Server

FIA_X509_EXT.1.1/ITT/Server

FIA_X509_EXT.1.1/ITT The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation **supporting a minimum path length of two certificates**.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- 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 Section 6.3, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5, no revocation method*]
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - *Server certificates presented for DTLS/TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
  - *Client certificates presented for DTLS/TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
  - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

Application Note 4

This SFR should be chosen if the TOE is distributed and the protocol(s) selected in FPT_ITT.1 utilize X.509v3 certificates for peer authentication. In this case, the use of revocation list checking is optional as there are additional requirements surrounding the enabling and disabling of the ITT channel as defined in FCO_CPC_EXT.1. If the revocation checking is not supported, the ST author should select "no revocation method". However, if certificate revocation checking is supported, the ST author must select whether this is performed using OCSP or CRLs.

The TOE must be capable of supporting a minimum path length of two certificates. That is, it must support a certificate hierarchy comprising of at least a self-signed root certificate and a leaf certificate.

The certificate chain validation is expected to terminate with a trust anchor. This means the validation can terminate with any trusted CA certificate administratively designated as a trust anchor

or default to terminate with a Root CA. If the TOE validates certificates presented by remote endpoints (i.e., external IT entities, remote administrators, or remote parts of the TOE), the CA certificates designated as trust anchors must be loaded into the trust store ('certificate store', ' trusted CA Key Store' or similar) managed by the platform. In such cases, the TOE's trust store must support loading of multiple hierarchical CA certificates or certificate chains and must clearly indicate all certificates it considers trust anchors. If the TOE only presents its own certificate (e.g., a web server without mutual authentication), implementing the trust store is optional.

The validation of X.509v3 leaf certificates comprises several steps:

a. *A Certificate Revocation Check refers to the process of determining the current revocation status of an otherwise structurally valid certificate. This is optionally performed when a certificate is used for authentication, however this behaviour must be consistent. If this check is performed, it must be performed for each certificate in the chain up to, but not including, the trust anchor. This means that CA certificates that are not trust anchors, and leaf certificates in the chain, must be checked. It is not required to check the revocation status of any CA certificate designated a trust anchor, however if such check is performed it must be handled consistently with how other certificates are checked.*
b. *An expiration check must be performed. This check must be conducted for each certificate in the chain, up to and including the trust anchor.*
c. *The continuity of the chain must be checked, showing that the signature on each certificate that is presented to the TOE is valid and the chain terminates at the trust anchor.*

If revocation checking is performed, it is expected that it is performed on both leaf and intermediate CA certificates when a leaf certificate is presented to the TOE as part of the certificate chain during authentication. Revocation checking of any CA certificate designated a trust anchor is not required. It is not sufficient to perform a revocation check of an intermediate CA certificate only when it is loaded onto the device.

If the TOE does not support functionality that uses any of the certificate types listed in the extendedKeyUsage rules in FIA_X509_EXT.1.1/ITT then this is stated in the TSS and the relevant part of the SFR is considered trivially satisfied. However, if the TOE does support functionality that uses certificates of any of these types then the corresponding rule must of course be satisfied as in the SFR.

FIA_X509_EXT.1.2/ITT The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

Application Note 5

This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.

FIA_X509_EXT.1.2/ITT/Server The application shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

Application Note 6: This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.

A.3. Protection of the TSF (FPT)

A.3.1. FPT_ITT.1/Server

FPT_ITT.1.1/Server The TSF shall protect TSF data from **disclosure and detect its modification** when it is transmitted between separate parts of the TOE **through the use of [*selection: SSH, TLS, DTLS, HTTPS*]**.

Application Note 7: The data passed in this trusted communication channel are encrypted as defined by the protocol chosen in the selection. The ST author should identify the channels and protocols used by each pair of communicating TOE parts, iterating this SFR as appropriate.

If certificates are used for authentication in any of the protocols selected above, then FIA_X509_EXT.1/ITT/Server is to be selected.

Appendix B: Optional Requirements

There are currently no Optional requirements. Following section may be applicable in later revisions.

Appendix C: Extended Component Definitions

This appendix contains the definitions for the extended requirements that are used in the PP-Module, including those used in Consistency Rationale and Selection-Based Requirements .

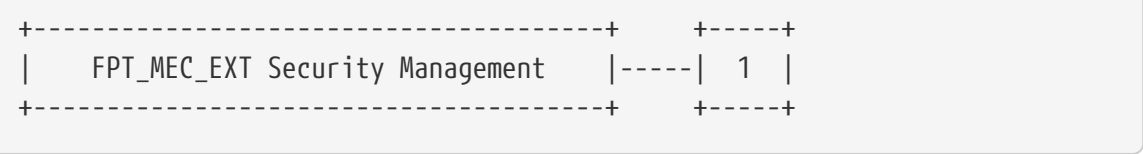
C.1. Security Management (FMT)

C.1.1. Family Behaviour

Components in this family address requirements for secure configuration. This is a new family defined for the FMT class.

C.1.2. Component Levelling

Component leveling



FPT_MEC_EXT.1/Server ensures that the TOE is not vulnerable to malicous configuration changes by unauthorized access or an escalation of privledge attack.

Management: FPT_MEC_EXT.1/Server

The following actions could be considered for the management functions in FPT:

a. There are no management activities foreseen

Audit: FPT_MEC_EXT.1/Server

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a. No audit necessary

FMT_MEC_EXT.1/Server

Hierarchical to: No other components

Dependencies: No other components

FMT_MEC_EXT.1.1/Server Read and write access to the TOE's configuration data shall be limited to Administrator, TOE and [*assignment: list of authorized entities*].

C.2. Anti-Exploitation Capabilities (FPT_AEX_EXT)

C.2.1. Family Behaviour

This is a new component within the FPT class which addresses requirements to ensure the TOE is not susceptible to commonly used exploitation methods. Additionally, it ensures that the application doesn't circumvent security functionality provided by the platform. This is a new family defined for the FPT class.

C.2.2. Component Levelling

Component leveling

```
+--------------------------------------+     +-----+
|    FPT_AEX_EXT Anti-Exploitation     |-----|  2  |
+--------------------------------------+     +-----+
```

FPT_AEX_EXT.2 ensures the TOE is not susceptible to commonly used exploitation methods and that it doesn't circumvent security functionality provided by the platform.

Management: FPT_AEX_EXT.2

The following actions could be considered for the management functions in FPT:

a. There are no management activities foreseen

Audit: FPT_AEX_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a. No audit necessary

FPT_AEX_EXT.2/Server

Hierarchical to: No other components

Dependencies: No other components

FPT_AEX_EXT.2.1/Server The application shall be compatible with security features provided by the platform vendor.

C.3. Communication Partner Control (FCO_CPC_EXT)

C.3.1. Family Behaviour

This is a new component within the FCO class used to define high-level constraints on the ways that partner IT entities communicate. For example, there may be constraints on when communication channels can be used, how they are established, and links to SFRs expressing lower-level security properties of the channels.

C.3.2. Component Levelling

Component leveling

```
+--------------------------------------+     +-----+
|   FCO_CPC_EXT Component Registration  |-----|  1  |
+--------------------------------------+     +-----+
```

FCO_CPC_EXT.1/Server Component Registration Channel Definition, requires the TSF to support a registration channel for joining together server and agent TOE parts or other distributed application TOE components, and to ensure that the availability of this channel is under the control of an Administrator. It also requires statement of the type of channel used (allowing specification of further lower-level security requirements by reference to other SFRs).

Management: FCO_CPC_EXT.1/Server

The following actions could be considered for the management functions in FPT:

a. There are no management activities foreseen

Audit: FCO_CPC_EXT.1/Server

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a. Enabling communications between a pair of TOE parts as in FCO_CPC_EXT.1.1/Server (including identities of the endpoints).

b. Disabling communications between a pair of TOE parts as in FCO_CPC_EXT.1.3/Server (including identity of the endpoint that is disabled).

FCO_CPC_EXT.1/Server

Hierarchical to: No other components

Dependencies: No other components

FCO_CPC_EXT.1.1/Server The TSF shall require a Security Administrator to enable communications between any pair of TOE parts before such communication can take place.

FCO_CPC_EXT.1.2/Server The TSF shall implement a registration process in which TOE parts establish and use a communications channel that uses [*selection:*

- *A channel that meets the secure channel requirements in FPT_ITT.1*,
- *No channel*

].

FCO_CPC_EXT.1.3/Server The TSF shall enable a Security Administrator to disable communications between any pair of TOE parts.

Appendix D: Consistency Rationale

D.1. Consistency of TOE Type

When this PP-Module is used to extend [cPP_APP_SW], the TOE type for the overall TOE is still a generic application. However, one of the functions of the device must be the ability for it to the capability to manage agent applications. The TOE boundary is simply extended to include that functionality.

D.2. SFR Dependencies Analysis

The dependencies between SFRs implemented by the TOE are addressed as shown in the base PP.