Comment: Comment-1-
Comment: Comment-2-
Comment: Comment-3-
Comment: Comment-4-
Comment: Comment-5-
Comment: Comment-6-
Comment: Comment-7-
Comment: Comment-8-

PP-Module for Software Defined Networking Controllers

Version: 1.0
2024-10-31
National Information Assurance Partnership

Revision History

Version	Date	Comment
1.0	2024-10-31	First draft of version 1.0 for comment

Contents

1 Introduction

1.1 Overview

The scope of this Protection Profile Module (PP-Module) is to describe the security functionality of a software defined network (SDN) controller in terms of [CC] and to define functional and assurance requirements for such products. This PP-Module is intended for use with the following Base-PP:

- collaborative Protection Profile for Network Devices (NDcPP), Version 3.0E

This Base-PP is valid because an SDN controller is a specific implementation of a network device. Specifically, an SDN controller is one of many components of an SDN networking architecture. Specifically, an SDN controller manages and distributes network policies, collects routing and payload information from the data plane, and interfaces with user applications in the management plane. Each of the planes in an SDN system is composed of multiple logical or physical components. SDN controllers logically centralize the network intelligence and state in the control plane.

1.2 Terms

The following sections list Common Criteria and technology terms used in this document.

1.2.1 Common Criteria Terms

Assurance	Grounds for confidence that a TOE meets the SFRs [CC].
Base Protection Profile (Base-PP)	Protection Profile used as a basis to build a PP-Configuration.
Collaborative Protection Profile (cPP)	A Protection Profile developed by international technical communities and approved by multiple schemes.
Common Criteria (CC)	Common Criteria for Information Technology Security Evaluation (International Standard ISO/IEC 15408).
Common Criteria Testing Laboratory	Within the context of the Common Criteria Evaluation and Validation Scheme (CCEVS), an IT security evaluation facility accredited by the National Voluntary Laboratory Accreditation Program (NVLAP) and approved by the NIAP Validation Body to conduct Common Criteria-based evaluations.
Common Evaluation Methodology (CEM)	Common Evaluation Methodology for Information Technology Security Evaluation.
Direct Rationale	A type of Protection Profile, PP-Module, or Security Target in which the security problem definition (SPD) elements are mapped directly to the SFRs and possibly to the security objectives for the operational environment. There are no security objectives for the TOE.
Distributed TOE	A TOE composed of multiple components operating as a logical whole.
Extended Package (EP)	A deprecated document form for collecting SFRs that implement a particular protocol, technology, or functionality. See Functional Packages.
Functional Package (FP)	A document that collects SFRs for a particular protocol, technology, or functionality.
Operational Environment (OE)	Hardware and software that are outside the TOE boundary that support the TOE functionality and security policy.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of products.
Protection Profile Configuration (PP-Configuration)	A comprehensive set of security requirements for a product type that consists of at least one Base-PP and at least one PP-Module.
Protection	An implementation-independent statement of security needs for a TOE type complementary

Profile Module (PP-Module)	to one or more Base-PPs.
Security Assurance Requirement (SAR)	A requirement to assure the security of the TOE.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Target (ST)	A set of implementation-dependent security requirements for a specific product.
Target of Evaluation (TOE)	The product under evaluation.
TOE Security Functionality (TSF)	The security functionality of the product under evaluation.
TOE Summary Specification (TSS)	A description of how a TOE satisfies the SFRs in an ST.

1.2.2 Technical Terms

Address Space Layout Randomization (ASLR)	An anti-exploitation feature which loads memory mappings into unpredictable locations. ASLR makes it more difficult for an attacker to redirect control to code that they have introduced into the address space of a process.
Administrator	An administrator is responsible for management activities, including setting policies that are applied by the enterprise on the operating system. This administrator could be acting remotely through a management server, from which the system receives configuration policies. An administrator can enforce settings on the system which cannot be overridden by non-administrator users.
Application (app)	Software that runs on a platform and performs tasks on behalf of the user or owner of the platform, as well as its supporting documentation.
Application Programming Interface (API)	A specification of routines, data structures, object classes, and variables that allows an application to make use of services provided by another software component, such as a library. APIs are often provided for a set of libraries included with the platform.
Control Plane	A logical entity that receives instructions or requirements from the SDN application layer through its northbound interface and relays them to the data plane through its southbound interface. The controller extracts information about the network from the data plane and communicates back to the SDN application layer with an abstract view of the network, including statistics and events about what is happening.
Credential	Data that establishes the identity of a user (e.g., a cryptographic key or password).
Critical Security Parameters (CSP)	Information that is either user or system defined and is used to operate a cryptographic module in processing encryption functions including cryptographic keys and authentication data, such as passwords, the disclosure or modification of which can compromise the security of a cryptographic module or the security of the information protected by the module.
DAR Protection	Countermeasures that prevent attackers, even those with physical access, from extracting data from non-volatile storage. Common techniques include data encryption and wiping.
Data Execution Prevention (DEP)	An anti-exploitation feature of modern operating systems executing on modern computer hardware, which enforces a non-execute permission on pages of memory. DEP prevents pages of memory from containing both data and instructions, which makes it more difficult for an attacker to introduce and execute code.
Data Plane	Controls the forwarding and data processing capabilities for the network. This includes forwarding and processing of the data path.
Developer	An entity that writes OS software. For the purposes of this document, vendors and developers are the same.

Host-based Firewall	A software-based firewall implementation running on the OS for filtering inbound and outbound network traffic to and from processes running on the OS.
Hybrid Authentication	A hybrid authentication factor is one where a user has to submit a combination of a cryptographic token and a PIN or password and both must pass. If either factor fails, the entire attempt fails.
Management Plane	Composed of programs that communicate behaviors and needed resources with the SDN controller via application programming interfaces (APIs). In addition, the applications can build an abstracted view of the network by collecting information from the controller for decision-making purposes. These applications could include networking management, analytics, or business applications used to run large data centers. For example, an analytics application might be built to recognize suspicious network activity for security purposes. This is sometimes also referred to as the Orchestration Layer.
Northbound	Communications between an SDN and applications in the management plane.
Operating System (OS)	Software that manages physical and logical resources and provides services for applications. The terms TOE and OS are interchangeable in this document.
Personal Identification Number (PIN)	An authentication factor that is comprised of a set of numeric or alphabetic characters that may be used in addition to a cryptographic token to provide a hybrid authentication factor. At this time it is not considered as a stand-alone authentication mechanism. A PIN is distinct from a password in that the allowed character set and required length of a PIN is typically smaller than that of a password as it is designed to be input quickly.
Personally Identifiable Information (PII)	Any information about an individual maintained by an agency, including, but not limited to, education, financial transactions, medical history, and criminal or employment history, and information which can be used to distinguish or trace an individual's identity, such as their name, social security number, date and place of birth, mother's maiden name, biometric records, etc., including any other personal information which is linked or linkable to an individual.
Sensitive Data	Sensitive data may include all user or enterprise data or may be specific application data such as PII, emails, messaging, documents, calendar items, and contacts. Sensitive data must minimally include credentials and keys. Sensitive data shall be identified in the OS's TSS by the ST author.
Southbound	Communications between an SDN and network devices in the data plane.
User	A user is subject to configuration policies applied to the operating system by administrators. On some systems, under certain configurations, a normal user can temporarily elevate privileges to that of an administrator. At that time, such a user should be considered an administrator.

1.3 Compliant Targets of Evaluation

Compliant TOEs will implement necessary functionality in a "do no harm" manner with respect to network security. Specifically:

- They can be remotely managed in a secure manner.
- Any software/firmware updates are from a trusted source.
- Any interfaces to applications, VMs, and other devices are trusted (zone of trust).
- All security-relevant events are reported.
- The control plane and data plane are logically separated.
- The control plane and management plane are logically separated.
- Data that is collected by the control plane and distributed to the data plane, such as flow tables and configurations, is protected.
- Data that is collected by the control plane and distributed to the management plane, such as auditable events and traffic or packet statistics, is protected.
- Data that is collected, distributed, and shared within the control plane, such as when multiple SDN controllers exist in the architecture, is protected.

A conformant TOE decouples its data and control planes such that data traffic and control traffic are restricted to their respective planes. It also enforces centralization of control so that all components of the SDN environment are under its control. This can expand across multiple distributed controllers for large, complex, or geographically dispersed networks.

1.3.1 TOE Boundary

The TOE boundary for an SDN controller is one or more physical or virtual devices. An SDN controller may be a distributed TOE, as defined by the Base-PP.

- In a physical standalone SDN controller device, the device's hardware, firmware, and software define the evaluation boundary.
- In a virtual SDN controller, the software of the Virtual Machine (VM) defines the evaluation boundary.
- All security functionality is contained and executed within the evaluation boundary of the SDN controller.

The following figure shows the SDN controller sitting between the management and data planes within the SDN infrastructure. This is a simplified diagram of the TOE's position in an SDN deployment. Other dependencies that are necessary to meet security requirements, such as an audit server, remote management interface, or source of certificate revocation information are not shown.

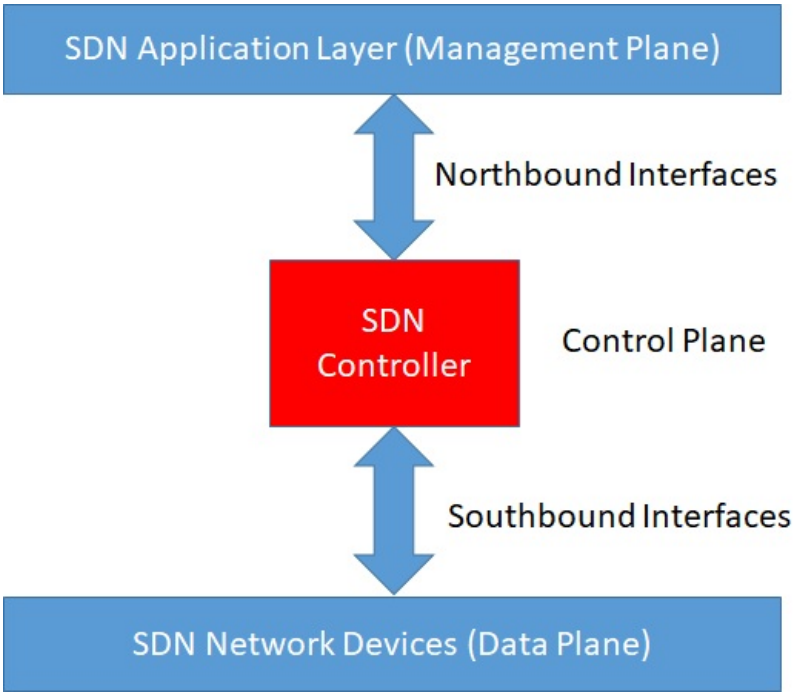

Figure 1: High-Level SDN Representation

The following elements of an SDN controller are outside the scope of this PP-Module and are therefore considered to be non-interfering with respect to security, even if they are included as part of a compliant product:

- Examination of data plane content, such as virus or email scanning.
- Intrusion detection or prevention capabilities.
- Network Address Translation (NAT) as a security function.
- If the TOE boundary is a single virtual machine, the hardware or firmware of the underlying platform.
- If the TOE boundary is a single virtual machine, the host operating system or runtime environment.
- Specific security functionality that is not global to all SDN controllers (e.g. firewall, load balancing).
- Ohter objects belonging in the data plane.

1.4 Use Cases

Requirements in this PP-Module are designed to address the security problems in at least the following use cases. These use cases are intentionally very broad, as many specific use cases exist for an SDN controller These use cases may also overlap with one another. An SDN controller's functionality may even be effectively extended by privileged applications installed on it. However, these are out of scope of this PP.

[USE CASE 1] Standalone Physical Device
The TOE is a single physical appliance that provides SDN controller functionality.

[USE CASE 2] Virtual Device
The TOE is a virtual machine that provides SDN controller functionality. The TOE boundary is either limited to the virtual machine or includes the hypervisor and underlying physical hardware, depending on whether or not the hypervisor may include virtual machines that are not part of the SDN controller.

[USE CASE 3] Cluster (stand-alone or virtual)
Regardless of whether the TOE is physical, virtual, or both, it includes multiple distinct instances that are combined into a distributed cluster.

[USE CASE 4] Hyper-Converge
The TOE amalgamates various computing resources into a single unit for storage-centric, server-centric, or hybrid (storage-server) workloads.

2 Conformance Claims

Conformance Statement

An ST must claim exact conformance to this PP-Module.

The evaluation methods used for evaluating the TOE are a combination of the workunits defined in [CEM] as well as the Evaluation Activities for ensuring that individual SFRs and SARs have a sufficient level of supporting evidence in the Security Target and guidance documentation and have been sufficiently tested by the laboratory as part of completing ATE_IND.1. Any functional packages this PP claims similarly contain their own Evaluation Activities that are used in this same manner.

CC Conformance Claims

This PP-Module is conformant to Part 2 (extended) and Part 3 (conformant) of Common Criteria CC:2022, Revision 1.

PP Claim

This PP-Module does not claim conformance to any Protection Profile.

There are no PPs or PP-Modules that are allowed in a PP-Configuration with this PP-Module.

Package Claim

This PP-Module is not conformant to any Functional or Assurance Packages.

3 Security Problem Definition

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

3.1 Threats

T.INSECURE_API
Insecure application programming interfaces (APIs) and user interfaces can provide attackers with opportunities to inject malicious code to the TOE or to retrieve sensitive information from it. Executing improper or unauthorized API functions or providing malicious input can compromise network services and sensitive data leakage can take place through insecure interfaces.

Recommendations above to adjust threat name for more specificity and clarify the description.

T.ATTACKER_ACCESS
An attacker may attempt to exploit services and functionality to intercept communications and mount attacks against machines in the network.

Threat kept as-is but this seems unnecessary. Unless there are SDN-specific threat cases (such as the API threat listed above), this just seems like an extension of threats already defined in the NDcPP, and it should be fine for module SFRs to map back to threats from the base PP.

3.2 Assumptions

These assumptions are made on the Operational Environment (OE) in order to be able to ensure that the security functionality specified in the PP-Module can be provided by the TOE. If the TOE is placed in an OE that does not meet these assumptions, the TOE may no longer be able to provide all of its security functionality.

A.SECURED_INFRASTRUCTURE
Connections between SDN Controller and network devices under the SDN Controller's control are secure from unauthorized access.

A.SUPPORTED_API
The SDN Controller will have an API in the management interface.

4 Security Objectives

4.1 Security Objectives for the TOE

This document does not define any additional SOs.

4.2 Security Objectives for the Operational Environment

The following security objectives for the operational environment help the OS in correctly providing its security functionality. These track with the assumptions about the environment.

OE.QQQQ
QQQQ

4.3 Security Objectives Rationale

This section describes how the assumptions and organizational security policies map to operational environment security objectives.

Table 1: Security Objectives Rationale

Assumption or OSP	Security Objectives	Rationale
A.SECURED_INFRASTRUCTURE	OE.QQQQ	
A.SUPPORTED_API	OE.QQQQ	

5 Security Requirements

This chapter describes the security requirements which have to be fulfilled by the product under evaluation. Those requirements comprise functional components from Part 2 and assurance components from Part 3 of [CC]. The following conventions are used for the completion of operations:

- **Refinement** operation (denoted by **bold text** or ~~strikethrough text~~): Is used to add details to a requirement or to remove part of the requirement that is made irrelevant through the completion of another operation, and thus further restricts a requirement.
- **Selection** (denoted by *italicized text*): Is used to select one or more options provided by the [CC] in stating a requirement.
- **Assignment** operation (denoted by *italicized text*): Is used to assign a specific value to an unspecified parameter, such as the length of a password. Showing the value in square brackets indicates assignment.
- **Iteration** operation: Is indicated by appending the SFR name with a slash and unique identifier suggesting the purpose of the operation, e.g. "/EXAMPLE1."

5.1 Collaborative Protection Profile for Network Devices Security Functional Requirements Direction

In a PP-Configuration that includes the NDcPP, the TOE is expected to rely on some of the security functions implemented by the Network Device as a whole and evaluated against the NDcPP. The following sections describe any modifications that the ST author must make to the SFRs defined in the NDcPP in addition to what is mandated by Section 5.2 TOE Security Functional Requirements.

5.1.1 Modified SFRs

The SFRs listed in this section are defined in the NDcPP and relevant to the secure operation of the TOE.

5.1.1.1 Trusted Path/Channels (FTP)

FTP_ITC.1: Inter-TSF Trusted Channel

This SFR has been modified from its definition in the NDcPP to define external interfaces to environmental entities that are particular to this specific technology type.

The text of the requirement is replaced with:

FTP_ITC.1.1 The TSF shall be capable of using [**selection:** *IPsec, SSH, TLS, DTLS, HTTPS*] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, **northbound components, southbound components, external east/west components**[**selection:** *authentication server, [**assignment:** other capabilities], no 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.

FTP_ITC.1.2 The TSF shall permit [**selection:** *the TSF, the authorized IT entities*] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [**assignment:** *list of services for which the TSF is able to initiate communications*].

Application Note: This PP-Module modifies this SFR to allow for the specification of any northbound, southbound, or east/west environmental components with which the TSF may implement protected communications. A conformant TOE may implement a distributed east/west configuration rather than the east/west entities being in the OE; in this case, the ST would define the TOE boundary as a distributed TOE in accordance with the NDcPP and use FPT_ITT.1 to define the interface between east/west distributed TOE components.

5.2 TOE Security Functional Requirements

The following section describes the SFRs that must be satisfied by any TOE that claims conformance to this PP-Module. These SFRs must be claimed regardless of which PP-Configuration is used to define the TOE.

5.2.1 Security Audit (FAU)

FAU_GEN.1/SDN Audit Data Generation (SDN)

FAU_GEN.1.1/SDN

The SDN controller must have maximum auditing and logging of all unauthorized API usage, especially API usage attempts from users that are not in the API roles.

This will be populated as needed by the audit events for other SFRs

The TSF shall implement functionality to generate audit data of the following **SDN** auditable events:

a. Startup and shutdown of the audit functions;
b. All auditable events for [*not specified*] level of audit;
c. All unauthorized usage of all API endpoints, including create, read, update, and delete.
d. All authorized usage of all API endpoints, including create, read, update, and delete.
e. Full HTTP REST request parameters and values of any API requests sent to the SDNC controller API.
f. All HTTP Response Codes returned from any HTTP API requests sent to the SDNC controller API.
g. All error codes and error messages from usage of the API.
h. [*all auditable events for mandatory SFRs specified in Table t-audit-mandatory, selected SFRs in Table t-audit-sel-based*].

FAU_GEN.1.2/SDN

The TSF shall record within the **SDN** audit data at least the following information:

a. Date and time of the event, type of event, subject identity, (if relevant) the outcome (success or failure) of the event; and
b. For each audit event type, based on the auditable event definitions of the functional components included in the PP-**Module**/ST, [*Additional Audit Record Contents as specified in Table t-audit-mandatory, Table t-audit-sel-based*].

5.2.2 User Data Protection (FDP)

FDP_ACC.1 Subset Access Control

FDP_ACC.1.1

The TSF shall enforce the [*API access control policy*] on [*all APIs used to access the TSF*].

FDP_ACF.1 Security Attribute-Based Access Control

FDP_ACF.1.1

1. API Validation: The SDN controller must perform validation of API calls against the API templates.
2. API Validation: The SDN controller must block API calls that fail validation.
3. API Validation: The SDN controller must ensure that API calls conform to the API templates, and block any API calls that do not conform.

The TSF shall enforce the [*API access control policy*] to objects based on the following: [*supported operations that can be performed on or by API objects, the validity of the API call being issued, whether the API call is authorized based on the subject's role*].

FDP_ACF.1.2

The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [*an API call that is valid with respect to an allowlisted API template can manipulate an object if allowing this manipulation has been configured*].

FDP_ACF.1.3

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [*an allowlist explicitly authorizes the API request*].

Application Note: This does not explicitly assert that the request is valid with respect to the template, nor does it explicitly assert that the object being accessed can have the desired action performed on it.

FDP_ACF.1.4

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [*no additional rules*].

5.2.3 Security Management (FMT)

FMT_API_EXT.1 Management of API Behavior

FMT_API_EXT.1.1

•API Call Templates: The SDN controller must have API templates that define how API calls must be performed to access API objects.

The TSF shall provide the ability to define the following API templates [**assignment**: *list of API templates*] against the following API objects [**assignment**: *list of API objects*].

FMT_API_EXT.1.2

The TSF shall permit API templates to be defined if they meet the following specified rules: [**assignment**: *rules determining allowable templates*].

FMT_MOF.1/SDN Management of Functions Behavior (SDN)

FMT_MOF.1.1/SDN

Manipulation Control: The SDN controller must let the administrator specify how the allowed API objects can be manipulated.

The TSF shall restrict the ability to [*modify the behaviour of*] the functions [*API access function as defined by FDP_ACC.1, API validity function as defined by FMT_API_EXT.1*] to [*API Administrator*].

Application Note: The restriction of modifying API access and validity functions to the API Administrator is intended to imply that the API User does not have the ability to modify API functions. The API User role having read-only access to these functions is not precluded by this requirement, as this requirement only relates to the ability to modify the behavior of the API.

FMT_SMF.1/SDN Specification of Management Functions (SDN)

FMT_SMF.1.1/SDN

Allowlisting: The SDN controller must let the administrator be able to allowlist API templates so that configuration attempts from all other APIs not in the allowlist is blocked.

The TSF shall be capable of performing the following management functions: [*API access control as defined by FDP_ACF.1, API validity as defined by FMT_API_EXT.1*].

Application Note: API access control refers to configuring the allowlist for the permitted API templates. API validity refers to the ability to defining the templates for how API calls must be performed to access API objects. The intent of this requirement is for the TSF to have the ability to define both when and how a given API can be invoked.

FMT_SMR.2/SDN Restrictions on Security Roles (SDN)

FMT_SMR.2.1/SDN

This could also just be done as a Modified SFR to FMT_SMR.2 in the NDcPP but it may be cleaner just to do it as its own iteration here.

1. RBAC: The SDN controller must provide an API user role and API administrator role for role-based access control
2. RBAC: The SDN controller must require that only administrators in the API administrator role can perform API calls.

The TSF shall maintain the roles: [*API User, API Administrator*].

FMT_SMR.2.2/SDN

The TSF shall be able to associate users with roles.

FMT_SMR.2.3/SDN

The TSF shall ensure that the conditions [*API User and API Administrator roles cannot be held simultaneously*] are satisfied.

5.3 TOE Security Functional Requirements Rationale

The following rationale provides justification for each SFR for the TOE, showing that the SFRs are suitable to address the specified threats:

Table 2: SFR Rationale

Threat	Addressed by	Rationale

T.INSECURE_API

6 Consistency Rationale

6.1 Collaborative Protection Profile for Network Devices

6.1.1 Consistency of TOE Type

When this PP-Module is used to extend the NDcPP, the TOE type for the overall TOE is still a network device. The TOE boundary is simply extended to include SDN controller functionality that is provided by the network device.

6.1.2 Consistency of Security Problem Definition

The threats defined by this PP-Module (see section 3.1) supplement those defined in the NDcPP as follows:

Table 3: Consistency of Security Problem Definition (NDcPP base)

PP-Module Threat, Assumption, OSP	Consistency Rationale
T.INSECURE_API	
T.ATTACKER_ACCESS	
A.SECURED_INFRASTRUCTURE	
A.SUPPORTED_API	

6.1.3 Consistency of OE Objectives

TBD

Table 4: Consistency of OE Objectives (NDcPP base)

PP-Module OE Objective	Consistency Rationale
OE.QQQQ	

6.1.4 Consistency of Requirements

This PP-Module identifies several SFRs from the NDcPP that are needed to support Software Defined Networking Controller functionality. This is considered to be consistent because the functionality provided by the NDcPP is being used for its intended purpose. The rationale for why this does not conflict with the claims defined by the NDcPP are as follows:

Table 5: Consistency of Requirements (NDcPP base)

PP-Module Requirement	Consistency Rationale
Modified SFRs	
FTP_ITC.1	This PP-Module expands the Base-PP SFR to define additional entities for trusted channels.
Additional SFRs	
This PP-Module does not add any requirements when the NDcPP is the base.	
Mandatory SFRs	
FAU_GEN.1/SDN	
FDP_ACC.1	
FDP_ACF.1	
FMT_API_EXT.1	
FMT_MOF.1/SDN	
FMT_SMF.1/SDN	
FMT_SMR.2/SDN	
Optional SFRs	
This PP-Module does not define any Optional requirements.	

Objective SFRs

This PP-Module does not define any Objective requirements.

Implementation-dependent SFRs

This PP-Module does not define any Implementation-dependent requirements.

Selection-based SFRs

This PP-Module does not define any Selection-based requirements.

Appendix A - Optional SFRs

A.1 Strictly Optional Requirements

This PP-Module does not define any Strictly Optional SFRs or SARs.

A.2 Objective Requirements

This PP-Module does not define any Objective SFRs.

A.3 Implementation-dependent Requirements

This PP-Module does not define any Implementation-dependent SFRs.

Appendix B - Selection-based Requirements

This PP-Module does not define any Selection-based SFRs.

Appendix C - Entropy Documentation and Assessment

This appendix describes the required supplementary information for the entropy source used by the OS. The documentation of the entropy source should be detailed enough that, after reading, the evaluator will thoroughly understand the entropy source and why it can be relied upon to provide sufficient entropy. This documentation should include multiple detailed sections: design description, entropy justification, operating conditions, and health testing. This documentation is not required to be part of the TSS.

C.1 Design Description

Documentation will include the design of the entropy source as a whole, including the interaction of all entropy source components. Any information that can be shared regarding the design should also be included for any third-party entropy sources that are included in the product.
The documentation will describe the operation of the entropy source to include, how entropy is produced, and how unprocessed (raw) data can be obtained from within the entropy source for testing purposes. The documentation should walk through the entropy source design indicating where the entropy comes from, where the entropy output is passed next, any post-processing of the raw outputs (hash, XOR, etc.), if and where it is stored, and finally, how it is output from the entropy source. Any conditions placed on the process (e.g., blocking) should also be described in the entropy source design. Diagrams and examples are encouraged.
This design must also include a description of the content of the security boundary of the entropy source and a description of how the security boundary ensures that an adversary outside the boundary cannot affect the entropy rate.
If implemented, the design description shall include a description of how third-party applications can add entropy to the RBG. A description of any RBG state saving between power-off and power-on shall be included.

C.2 Entropy Justification

There should be a technical argument for where the unpredictability in the source comes from and why there is confidence in the entropy source delivering sufficient entropy for the uses made of the RBG output (by this particular OS). This argument will include a description of the expected min-entropy rate (i.e. the minimum entropy (in bits) per bit or byte of source data) and explain that sufficient entropy is going into the OS randomizer seeding process. This discussion will be part of a justification for why the entropy source can be relied upon to produce bits with entropy.
The amount of information necessary to justify the expected min-entropy rate depends on the type of entropy source included in the product.
For developer-provided entropy sources, in order to justify the min-entropy rate, it is expected that a large number of raw source bits will be collected, statistical tests will be performed, and the min-entropy rate determined from the statistical tests. While no particular statistical tests are required at this time, it is expected that some testing is necessary in order to determine the amount of min-entropy in each output.
For third-party-provided entropy sources, in which the OS vendor has limited access to the design and raw entropy data of the source, the documentation will indicate an estimate of the amount of min-entropy obtained from this third-party source. It is acceptable for the vendor to 'assume' an amount of min-entropy, however, this assumption must be clearly stated in the documentation provided. In particular, the min-entropy estimate must be specified and the assumption included in the ST.
Regardless of type of entropy source, the justification will also include how the DRBG is initialized with the entropy stated in the ST, for example by verifying that the min-entropy rate is multiplied by the amount of source data used to seed the DRBG or that the rate of entropy expected based on the amount of source data is explicitly stated and compared to the statistical rate. If the amount of source data used to seed the DRBG is not clear or the calculated rate is not explicitly related to the seed, the documentation will not be considered complete.
The entropy justification shall not include any data added from any third-party application or from any state saving between restarts.

C.3 Operating Conditions

The entropy rate may be affected by conditions outside the control of the entropy source itself. For example, voltage, frequency, temperature, and elapsed time after power-on are just a few of the factors that may affect the operation of the entropy source. As such, documentation will also include the range of operating conditions under which the entropy source is expected to generate random data. It will clearly describe the measures that have been taken in the system design to ensure the entropy source continues to operate under those conditions. Similarly, documentation shall describe the conditions under which the entropy source is known to malfunction or become inconsistent. Methods used to detect failure or degradation of the source shall be included.

C.4 Health Testing

More specifically, all entropy source health tests and their rationale will be documented. This includes a description of the health tests, the rate and conditions under which each health test is performed (e.g., at start, continuously, or on-demand), the expected results for each health test, and rationale indicating why each test is believed to be appropriate for detecting one or more failures in the entropy source.

Appendix D - Bibliography

Appendix E - Acronyms

Table 6: Acronyms

Acronym	Meaning
ABAC	Attribute-Based Access Control
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
app	Application
APT	Advanced Persistent Threats
ASLR	Address Space Layout Randomization
Base-PP	Base Protection Profile
CC	Common Criteria
CEM	Common Evaluation Methodology
CESG	Communications-Electronics Security Group
CLI	Command-Line Interface
CMC	Certificate Management over CMS
CMS	Cryptographic Message Syntax
CN	Common Names
cPP	Collaborative Protection Profile
CRL	Certificate Revocation List
CRUD	Create, Read, Update, Delete
CSA	Computer Security Act
CSP	Critical Security Parameters
DAR	Data At Rest
DEP	Data Execution Prevention
DES	Data Encryption Standard
DHE	Diffie-Hellman Ephemeral
DNS	Domain Name System
DRBG	Deterministic Random Bit Generator
DSS	Digital Signature Standard
DT	Date/Time Vector
DTLS	Datagram Transport Layer Security
EAP	Extensible Authentication Protocol
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
ECDSA	Elliptic Curve Digital Signature Algorithm
EP	Extended Package
ESR	Equivalent Series Resistance
EST	Enrollment over Secure Transport
FIPS	Federal Information Processing Standards
FP	Functional Package

HMAC	Hash-based Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISO	International Organization for Standardization
IT	Information Technology
ITSEF	Information Technology Security Evaluation Facility
MFA	Multi-Factor Authentication
MITM	Man-in-the-Middle
NAT	Network Address Translation
NFC	Near Field Communication
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
OE	Operational Environment
OID	Object Identifier
OMB	Office of Management and Budget
OS	Operating System
OWASP	Open Worldwide Application Securtiy Project
PII	Personally Identifiable Information
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PP	Protection Profile
PP	Protection Profile
PP-Configuration	Protection Profile Configuration
PP-Module	Protection Profile Module
RBAC	Role-Based Access Control
RBG	Random Bit Generator
REST	Representational State Transfer
RFC	Request for Comment
RNG	Random Number Generator
S/MIME	Secure/Multi-purpose Internet Mail Extensions
SAN	Subject Alternative Name
SAR	Security Assurance Requirements
SAR	Security Assurance Requirement
SDN	Software Defined Networking
SFR	Security Functional Requirements
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm

SIP	Session Initiation Protocol
ST	Security Target
SWID	Software Identification
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USB	Universal Serial Bus
VM	Virtual Machine
VPN	Virtual Private Network
XCCDF	eXtensible Configuration Checklist Description Format
XOR	Exclusive Or

Appendix F - Bibliography

Table 7: Bibliography

Identifier	Title
[CC]	Common Criteria for Information Technology Security Evaluation - • Part 1: Introduction and General Model, CCMB-2022-11-001, CC:2022, Revision 1, November 2022 • Part 2: Security Functional Requirements, CCMB-2022-11-002, CC:2022, Revision 1, November 2022. • Part 3: Security Assurance Requirements, CCMB-2022-11-003, CC:2022, Revision 1, November 2022. • Part 4: Framework for the Specification of Evaluation Methods and Activities, CCMB-2022-11-004, CC:2022, Revision 1, November 2022. • Part 5: Pre-defined Packages of Security Requirements, CCMB-2022-11-004, CC:2022, Revision 1, November 2022.
[CC]	Common Criteria for Information Technology Security Evaluation - • Part 1: Introduction and general model, CCMB-2022-11-001, CC:2022, Revision 1, November 2022. • Part 2: Security functional requirements, CCMB-2022-11-002, CC:2022, Revision 1, November 2022. • Part 3: Security assurance requirements, CCMB-2022-11-003, CC:2022, Revision 1, November 2022. • Part 4: Framework for the specification of evaluation methods and activities, CCMB-2022-11-004, CC:2022, Revision 1, November 2022. • Part 5: Pre-defined packages of security requirements, CCMB-2022-11-005, CC:2022, Revision 1, November 2022.
[CEM]	Common Common Methodology for Information Technology Security Evaluation, CCMB-2022-11-006, CC:2022, Revision 1, November 2022.
[CEM]	Common Methodology for Information Technology Security Evaluation - • Evaluation methodology, CCMB-2022-11-006, CC:2022, Revision 1, November 2022.
[OMB]	Reporting Incidents Involving Personally Identifiable Information and Incorporating the Cost for Security in Agency Information Technology Investments, OMB M-06-19, July 12, 2006.