

PP-Module for Software-Defined Networking Controllers



Version: 1.0

2026-04-20

National Information Assurance Partnership

Revision History

Version	Date	Comment
1.0	2026-04-20	Initial release

Contents

- 1 Introduction
 - 1.1 Overview
 - 1.2 Terms
 - 1.2.1 Common Criteria Terms
 - 1.2.2 Technical Terms
 - 1.3 Compliant Targets of Evaluation
 - 1.3.1 TOE Boundary
 - 1.4 Use Cases
- 2 Conformance Claims
- 3 Security Problem Definition
 - 3.1 Threats
 - 3.2 Assumptions
 - 3.3 Organizational Security Policies
- 4 Security Objectives
 - 4.1 Security Objectives for the Operational Environment
- 5 Security Requirements
 - 5.1 Collaborative Protection Profile for Network Devices Security Functional Requirements Direction
 - 5.1.1 Modified SFRs
 - 5.1.1.1 Class FTP: Trusted Path/Channels
 - 5.2 TOE Security Functional Requirements
 - 5.2.1 Auditable Events for Mandatory SFRs
 - 5.2.2 Class FAU: Security Audit
 - 5.2.3 Class FDP: User Data Protection
 - 5.2.4 Class FMT: Security Management
 - 5.3 TOE Security Functional Requirements Rationale
 - 5.4 TOE Security Assurance Requirements
- 6 Consistency Rationale
 - 6.1 Collaborative Protection Profile for Network Devices
 - 6.1.1 Consistency of TOE Type
 - 6.1.2 Consistency of Security Problem Definition
 - 6.1.3 Consistency of OE Objectives
 - 6.1.4 Consistency of Requirements
- Appendix A - Optional SFRs
 - A.1 Strictly Optional Requirements
 - A.2 Objective Requirements
 - A.3 Implementation-dependent Requirements
- Appendix B - Selection-based Requirements
- Appendix C - Extended Component Definitions
 - C.1 Extended Components Table

C.2 Extended Component Definitions

C.2.1 Class FMT: Security Management

C.2.1.1 FMT_API_EXT Management of API Behavior

Appendix D - Acronyms

Appendix E - Bibliography

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.

An SDN controller is a central component of an SDN system and is available as a logical or physical device. An SDN controller manages and distributes network policies, collects network routing and payload information from the control and data planes, and interfaces with user applications in the management plane for functions such as configuration and logging. Each of the planes in an SDN system is composed of multiple logical or physical components. The SDN controller logically separates the data plane from the control plane and centralizes control which enhances network flexibility and scalability through more efficient network management.

This PP-Module is intended for use with the following Base-PP.

- collaborative Protection Profile for Network Devices, Version 4.0

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. 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.
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 Profile Module (PP-Module)	An implementation-independent statement of security needs for a TOE type complementary 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

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	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).
Data Plane	Controls the forwarding and data processing capabilities for the network. This includes forwarding and processing of the data path.
Management Plane	Composed of programs that communicate behaviors and needed resources with the SDN controller via 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.
Southbound	Communications between an SDN and network devices in the data plane.

1.3 Compliant Targets of Evaluation

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. Compliant TOEs will implement the following functionality.

- Ability to support secure remote administration.
- Ability to apply software or firmware updates from a trusted source.
- Secure interfaces to remote entities such as applications, VMs, and other devices (zone of trust).
- Reporting of all security-relevant events.
- Logical separation of the management, control, and data planes.
- Protection of data that is collected by the control plane and distributed to the data plane, such as flow tables and configuration.
- Protection of data that is collected by the control plane and distributed to the management plane, such as auditable events and traffic or packet statistics.
- Protection of data that is collected, distributed, and shared within the control plane, such as when multiple SDN controllers exist in the architecture.

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. The Base-PP includes guidance on whether the evaluation boundary includes the virtualization system on which the VM runs.
- All security functionality is contained and executed within the evaluation boundary of the SDN controller.

The following figure shows the logical position of the SDN controller 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.

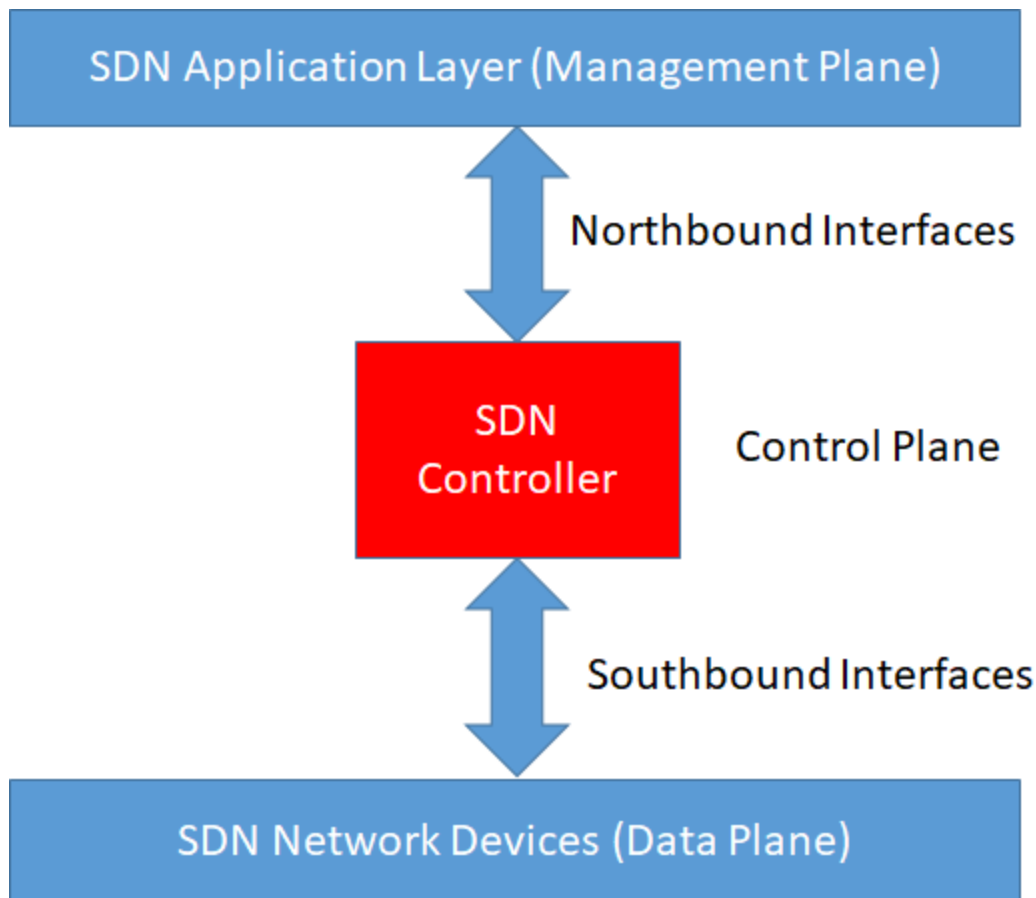


Figure 1: High-Level SDN Representation

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 its 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 standalone virtual machine, the hardware or firmware of the underlying platform.
- If the TOE boundary is a standalone 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).
- Other objects belonging in the data plane.

Other PP-Modules may exist to allow for other functions such as these to be defined as part of the TOE boundary.

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-Module.

[USE CASE 1] Physical Device

The TOE is one or more physical appliances that provide SDN controller functionality. The TOE boundary includes the entire software and firmware running on these appliances; a software-only device where the boundary does not include the underlying operating system is not a valid use case.

[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 or if the hypervisor itself includes functionality necessary for the ~~TOE~~ to implement the required security functionality. As with the physical device use case above, the ~~TOE~~ boundary for a virtual device must include an entire virtual machine and not just a collection of applications or services running on an environmental operating system.

[USE CASE 3] Cluster (stand-alone or virtual)

Regardless of whether the ~~TOE~~ is physical, virtual, or both, it may include multiple distinct instances (i.e., multiple connected physical appliances or VMs) that are combined into a distributed cluster.

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 claims conformance to the following Protection Profiles:

- collaborative Protection Profile for Network Devices, v4.0

The following PPs and PP-Modules are allowed to be specified in a PP-Configuration with this PP-Module:

- collaborative Protection Profile Module for Stateful Traffic Filter Firewalls, version 2.0
- PP-Module for Authentication Servers, version 2.0
- PP-Module for MACsec Ethernet Encryption, version 2.0
- PP-Module for VPN Gateways, version 2.0

Package Claim

- This PP-Module is Functional Package for TLS, version 2.1 conformant.
- This PP-Module is Functional Package for Secure Shell, version 2.0 conformant.
- This PP-Module does not conform to any assurance packages.

The functional packages to which the PP conforms may include SFRs that are not mandatory to claim for the sake of conformance. An ST that claims one or more of these functional packages may include any non-mandatory SFRs that are appropriate to claim based on the capabilities of the TSE and on any triggers for their inclusion based inherently on the SFR selections made.

3 Security Problem Definition

The security problem is described in terms of the threats that the T.OE 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_NORTHBOUND_API

A malicious user or process may invoke insecure application programming interfaces (APIs) to inject malicious code to the T.OE on a northbound interface or to retrieve sensitive information from it. Executing improper or unauthorized API functions or providing malicious input to achieve unintended or improper results from proper or authorized API functions can compromise network services. This can also result in sensitive data leakage.

3.2 Assumptions

All assumptions from the Base-PP also apply to the T.OE's environment when it includes this PP-Module in its conformance claims. This document does not define any additional assumptions.

3.3 Organizational Security Policies

An organization deploying the T.OE is expected to satisfy the organizational security policy listed below in addition to all organizational security policies defined by the claimed Base-PP.

This document does not define any additional OSPs.

4 Security Objectives

4.1 Security Objectives for the Operational Environment

All environmental security objectives from the ~~Base-PP~~ also apply to the ~~TOE~~'s environment when it includes this ~~PP~~-Module in its conformance claims.

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 ~~striketrough 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 Class FTP: Trusted Path/Channels

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 as defined in the Functional Package for SSH, TLS as defined in the Functional Package for TLS, DTLS as defined in the Functional Package for TLS, HTTPS*] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, **northbound components, southbound components**, [**selection:** *authentication server, external east/west components* [**assignment:** *other capabilities*], *no other capabilities*] that is logically distinct from other communication channels and provides assured identification of its endpoints 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 **TSE** shall initiate communication via the trusted channel for [**assignment: list of services for which the TSE 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 **TSE** 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 Auditable Events for Mandatory SFRs

Table 1: Auditable Events for Mandatory Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1/SDN	No events specified	N/A
FDP_ACC.1	No events specified	N/A
FDP_ACF.1	No events specified	N/A
FMT_API_EXT.1	Successful and failed definition or modification of API template	Identification of template (if successful)
FMT_MOF.1/SDN	No events specified	N/A
FMT_SMF.1/SDN	No events specified	N/A
FMT_SMR.2/SDN	Assignment of user to role	Identification of subject and assigned role

5.2.2 Class FAU: Security Audit

FAU_GEN.1/SDN Audit Data Generation (SDN)

FAU_GEN.1.1/SDN

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

- Start-up and shutdown of the audit functions;
- All auditable events for [*not specified*] level of audit;
[
 - All unauthorized usage of all **API** endpoints, including create, read, update, and delete.
 - 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 *SDN* controller *API*.
- f. All *HTTP* Response Codes returned from any *HTTP API* requests sent to the *SDN* controller *API*.
- g. All error codes and error messages from usage of the *API*.
- h. All auditable events for mandatory *SFRs* specified in [Table 1](#)].

FAU_GEN.1.2/SDN

The *TSS* 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 applicable) the outcome (success or failure) of the event;
- b. For each auditable event type, based on the auditable event definitions of the functional components included in the *PP*, *PP-Module*, functional package, or *ST*, [Additional Audit Record Contents as specified in [Table 1](#)].

Application Note: This *SFR* mandates that audit data be generated for start-up and shutdown of the audit functions, which is a duplicate of FAU_GEN.1.1 in the *Base-PP*. If the *TOE* has one single audit mechanism, then the events used to conform to the *Base-PP* requirement also suffice here. Start-up and shutdown of the audit functions in the context of the *SDN* controller portion of the *TOE* only need to be logged separately if the *SDN* controller has a different logging mechanism from what the *Base-PP* uses.

Note also that in many cases the start-up and shutdown of the audit functions cannot be forced on its own because the audit functions may be operational by default. In this case, logging for start-up and shutdown of the *TOE* itself is sufficient. The purpose of this part of the requirement is to ensure that a malicious user cannot disable auditing to evade detection of other malicious use of the *TOE*.

The auditable events for the 'not specified' level of audit is intended to communicate that this *PP-Module* does not claim one of the pre-defined audit levels specified in *CC* Part 2; this is inherently satisfied by demonstrating that the *TSS* generates appropriate audit data for all events that are explicitly required by the *PP-Module*.

Evaluation Activities ▼

[FAU_GEN.1/SDN](#)

TSS

The evaluator shall verify that the *TSS* identifies the auditable events generated by the *TOE*, and that they include the auditable events specified in this *SFR* along with the required audit data. The evaluator shall also verify that the *TSS* identifies the mechanism used for generating audit events such that it can be determined whether the auditable events required by this *PP-Module* use the same audit mechanism as what the *Base-PP* requires for FAU_GEN.1 or there are alternate mechanisms that are used.

The evaluator shall check that the requirement in the *Base-PP* to ensure that all auditable events can be transmitted to the operational environment via a secure channel is applicable to the entire *TOE*, even when *PP-Modules* are claimed. Therefore, if the *TSS* does have any audit mechanisms that are used specifically to generate audit records for this *PP-Module*'s auditable events, the evaluator shall ensure that these mechanisms are discussed in the context of the *Base-PP*'s requirements for protection of audit data in transit, and at rest if applicable.

Guidance

The evaluator shall verify that the operational guidance identifies the auditable events that are generated by the TOE in support of this SFR and includes representative audit records of each event to demonstrate the format of each record. The evaluator shall verify that each sample audit record includes the fields required for each auditable event per [FAU_GEN.1.2/SDN](#).

Tests

For each auditable event, the evaluator shall perform actions that are intended to cause a record of the event to be generated and verify that this occurs and that the required fields are present and accurate. If the audit mechanism used to generate any event differs from the mechanisms claimed in the Base-PP for [FAU_GEN.1](#), the evaluator shall also perform the associated test activities for other claimed FAU class requirements to verify that the audit data is transferred to an external entity over the claimed trusted channel and that if audit data is stored locally, that it has the appropriate data at rest protections applied.

5.2.3 Class FDP: User Data Protection

FDP_ACC.1 Subset Access Control

FDP_ACC.1.1

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

Application Note: The intent of this requirement is for the TOE to have a means to enforce access control on the invocation of APIs. The specific rules that comprise the policy itself are defined in [FDP_ACF.1](#) below. It is not expected that the TOE have a construct explicitly called the "API access control policy," only that enforcement mechanisms exist to construct such a policy.

Evaluation Activities ▼

[FDP_ACC.1](#)

TSS

The evaluator shall examine the TSS to verify that it defines the TOE's API access control policy in terms of subjects (the users, groups, or machine entities that interact with APIs), objects (the APIs used to access the TSE, operations (invocation, modification, or some other function that can be done against the object), and any attributes that may govern the enforcement of the policy. Operations may also include the ability to limit the scope of particular API access; for example, a user may be authorized to invoke an API to configure one segment of a network but may be prohibited from configuring a different one.

Guidance

The evaluator shall review the operational guidance to verify that it describes the TOE's API access control policy in terms of the subjects, objects, operations, and attributes specified in the ST, as well as which elements of this policy are enforced by default versus being configurable options. For any configurable elements of the API access control policy, the evaluator shall verify that the operational guidance describes how to perform this configuration.

Tests

There are no test activities for this component.

FDP_ACF.1 Security Attribute-Based Access Control

FDP_ACF.1.1

The TSSF 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]*.

Application Note: The purpose of the API access control policy is to identify the API objects that are controlled by the TOE, the various types of access that can be performed against this object, and whether that access is permitted based on either the validity of the request itself or the entity requesting that access.

FDP_ACF.1.2

The TSSF 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 TSSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[assignment: rules, based on security attributes, that explicitly allow access of subjects to objects]**.

Application Note: The purpose of this requirement is to define an override mechanism where an allowlist may explicitly authorize some API access that would not otherwise be permitted by the API access control policy. It may be the case that no explicit denylist is supported and the policy as defined in [FDP_ACF.1.2](#) always defines what is allowed and denied with no exceptions. If this is the case, the ST author may complete the assignment with "no additional rules."

This is intended to support the potential for operations that may be authorized on the basis of permissions; it is not intended to authorize API calls that are invalid regardless of permissions.

FDP_ACF.1.4

The TSSF shall explicitly deny access of subjects to objects based on the following additional rules: **[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]**.

Application Note: The purpose of this requirement is to define an override mechanism where a denylist may explicitly prohibit some API access that would not otherwise be permitted by the API access control policy. It may be the case that no explicit denylist is supported and the policy as defined in [FDP_ACF.1.2](#) always defines what is allowed and denied with no exceptions. If this is the case, the ST author may complete the assignment with "no additional rules."

Evaluation Activities ▼

The evaluator shall verify that the TSS describes the TOE's API access control policy in sufficient detail to conclude, for each API, the operations that can be performed on or by the API, and the authorization required to perform those operations. If the TOE implements explicit allowlisting or denylisting to override any default or configurable policy rules, the evaluator shall verify that these are described. In cases where multiple contradictory rules exist to define authorizations, the evaluator shall verify that the TSS describes the precedence in which these rules are prioritized.

Guidance

The evaluator shall verify that the operational guidance identifies, for each API, the mechanism that exist to control access to that API, and how to perform such configuration. The evaluator shall also verify that the operational guidance defines any default policy rules that may exist. The evaluator shall also verify that the operational guidance describes the ordering of rule processing such that it is clear what behavior is followed in the event that multiple contradictory rules exist. If this is configurable (e.g., because rules are enforced in a configurable priority order), the evaluator shall verify that the operational guidance provides instructions on how to perform this configuration.

Additionally, the evaluator shall verify that each API is defined in terms of the operations that can be performed by or against them and the valid parameters for these operations.

Tests

The evaluator shall perform the following tests:

- Test FDP_ACF.1:1: Default policy test: The evaluator shall review the operational guidance to determine what default policy rules (if any) are enforced by the TOE. For each such rule, the evaluator shall attempt to access the TOE's APIs in a manner prohibited by the default policy rule and verify that access is prohibited as expected. The evaluator shall then access these APIs as a subject with sufficient authorization and verify that the requested access is permitted.
- Test FDP_ACF.1:2: Configured policy test: For each API policy element that is configurable (e.g., by determining what users or groups are authorized to access a particular API), the evaluator shall follow the instructions in the operational guidance to configure positive and negative test scenarios where access is allowed and rejected. The evaluator shall then access the TOE as subjects that are and are not authorized to access the particular API, and verify that the TSS allows or rejects the access as configured.
- Test FDP_ACF.1:3: Contradictory policy test: For each situation identified in the ST where contradictory rules may exist (e.g., an operation is both allowlisted and denylisted), the evaluator shall apply these contradictory rules to the API, attempt to access the API in such a way that their authorization to do so is ambiguous (i.e., one rule permits the action and one rejects it), and verify that whether or not the attempt is authorized resolves in the manner specified in the TSS. If this functionality is configurable (e.g., the rules are enforced in a set order), the evaluator shall follow the steps necessary to change the priority of their enforcement, re-attempt the same access, and verify that the attempt resolves in the manner specified by the new configuration.
- Test FDP_ACF.1:4: Validity test: For each API, the evaluator shall connect a network traffic generator to the API and configure it to inject bad data into API requests that follow operations to legitimately assume a role that is authorized to invoke the API (in other words, attempt to perform an invalid operation as a valid subject). "Bad data" in this case is determined by the operational guidance's definition for what values are acceptable for the parameters of a particular API. In all cases, attempts to invoke an API using invalid data shall fail.

FMT_API_EXT.1 Management of API Behavior

FMT_API_EXT.1.1

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

Application Note: The intent of this requirement is to require the ST to define API endpoints that the SDN controller exposes for administering the SDN controller itself, and for inducing the SDN controller to administer the network devices that it manages. This could include, but is not limited to, APIs for the following behavior:

- Endpoints that modify the SDN controller's own configuration, including any changes to:
 - System settings
 - Operational parameters
- Endpoints that cause the SDN controller to modify network configurations, such as:
 - Routing settings
 - Switching settings
- Endpoints that change the SDN controller's power state, including:
 - Power on and off
 - Reboot operations
- Endpoints that cause the SDN controller to change the power state of managed network devices, including:
 - Power on and off
 - Reboot operations
- Endpoints that modify the SDN controller's security or hardening settings, including:
 - Security policies
 - Hardening configurations
- Endpoints that cause the SDN controller to modify security or hardening settings on managed network devices.
- Endpoints that change authentication or authorization controls on the SDN controller.
- Endpoints that change authentication or authorization controls on managed network devices.

Evaluation Activities ▼

[FMT_API_EXT.1](#)

TSS

The evaluator shall verify that the TSS identifies the API templates that can be applied to API objects, consistent with the claims made in [FMT_API_EXT.1.1](#). The evaluator shall verify that the

TSS describes any rules that are enforced on the definition of API templates.

Guidance

The evaluator shall examine the operational guidance to verify that the guidance identifies the API templates that can be applied to API objects as well as the administrative permissions necessary on the TOE to apply these. If additional templates may be defined per FMT_API_EXT.1.2, the evaluator shall verify that the operational guidance specifies the rules that are applied to allowable templates, both in terms of syntax (i.e., what constitutes a valid template) and authorization (i.e., what role or other attribute permits an administrator to perform this function).

Tests

The evaluator shall perform the following tests:

- *Test FMT_API_EXT.1:1: For each API template that can be applied to API objects, the evaluator shall verify that a sufficiently privileged administrator on the TOE is able to apply the template to the supported objects.*
- *Test FMT_API_EXT.1:2: [Conditional: The TOE supports the definition of additional templates] The evaluator shall follow the operational guidance to define a new API template that is consistent with the defined rules for both syntax and authorization and verify that the template is successfully defined.*
- *Test FMT_API_EXT.1:3: [Conditional: The TOE supports the definition of additional templates] If the ability to define additional templates is restricted based on administrative role or other attribute, the evaluator shall access the TOE as an insufficiently privileged administrator to define new API templates and verify that there is no mechanism by which a new template can be defined.*
- *Test FMT_API_EXT.1:4: [Conditional: The TOE supports the definition of additional templates] The evaluator shall access the TOE with an administrator account that is sufficiently privileged to define additional API templates and verify that a template is not created if it is invalid with respect to its syntax or composition.*

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

FMT_MOF.1.1/SDN

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.

Evaluation Activities ▼

FMT_MOF.1/SDN

TSS

The evaluator shall examine the TSS to verify that it identifies the API access and API validity functions provided by the TOE and verify that an API administrator is the only subject that is authorized to modify the behavior of these functions (e.g., by determining the subjects that are authorized to access a particular function). Note that the "API administrator" role is a necessary but

not sufficient condition to be authorized to perform management functions. If the *TSE* implements more granular controls (e.g., different *API* administrators have access to different sets of functions based on role or other subject attribute), the evaluator shall verify that the *TSS* includes sufficient data to determine that it fully describes the authorizations needed to perform management functions against the *TOE*'s APIs.

Guidance

The evaluator shall examine the operational guidance to verify that it describes the *TOE*'s claimed management functions and the privileges required to execute those functions.

Tests

For each claimed management function, the evaluator shall access the *TOE* as a legitimate user that lacks privileges to perform the function, attempt to perform the function, and verify that it fails (or that there is no mechanism by which an attempt can even be made). If multiple conditions define the ability to perform the function, the evaluator shall repeat this activity as needed to verify that the absence of each condition is sufficient to prevent the execution of the function. The evaluator shall then access the *TOE* with all necessary authorities to perform the function and verify that it is successful.

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

FMT_SMF.1.1/SDN

The *TSE* 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 *TSE* to have the ability to define both when and how a given *API* can be invoked.

Evaluation Activities ▼

[FMT_SMF.1/SDN](#)

TSS

The evaluator shall verify that the *TSS* identifies each management function that can be performed by the *TOE*, whether that is to execute a given function as defined by [FDP_ACC.1](#) or to modify the behavior of how the function is executed as defined by [FMT_MOF.1/SDN](#).

Guidance

The evaluator shall examine the operational guidance to verify that it provides instructions on how to perform each management function specified in this *SER*.

Tests

For each specified management function, the evaluator shall access the *TOE* as a subject that is sufficiently privileged to perform the function, execute the function, and verify that executing the function has the intended effect.

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

FMT_SMR.2.1/SDN

The TSSF shall maintain the roles: [API user, API administrator].

FMT_SMR.2.2/SDN

The TSSF shall be able to associate users with roles.

FMT_SMR.2.3/SDN

The TSSF shall ensure that the conditions [API user and API administrator roles cannot be held simultaneously] are satisfied.

Application Note: The Base-PP defines FMT_SMR.2 for the "Security Administrator" role that must be available on a network device in general. This iteration supplements that by making sure that API user and API administrator roles exist for SDN functionality but are logically separated. The security administrator as defined by that SFR may also be an API administrator as defined by this iteration, or management of general security functionality may be separated from API administration such that they are granted by different roles.

Evaluation Activities ▼

FMT_SMR.2/SDN

TSS

The evaluator shall verify that the TSS defines the roles available to subjects on the TOE, and that these may be categorized into API user and API administrator roles. The TOE does not need to implement roles with these specified names, but based on the description of the TOE's privilege model it should be clear how the roles and permissions on the TOE map to the concept of an API user that may only execute functions and an API administrator that has the ability to modify the behavior of those functions.

Guidance

The evaluator shall verify that the operational guidance identifies the administrative roles available for subjects on the TOE and the mechanism by which subjects are associated with roles. The evaluator shall also verify that this identifies how the API user and API administrator role cannot be held simultaneously (e.g., because a user may only have one role and these are defined as two separate roles; because write privilege and execute privilege cannot be held by the same subject for any given API).

Tests

The evaluator shall follow the operational guidance to associate subjects with the roles defined by the TOE to verify that subjects may be given an API user role or an API administrator role. The evaluator shall attempt to assign a subject to both roles and verify that this fails or that no mechanism exists to attempt this.

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: SER Rationale

Threat	Addressed by	Rationale
T.INSECURE_NORTHBOUND_API	FAU_GEN.1/SDN	Mitigates the threat by generating audit records of activities performed on or by the TSF that could affect the northbound interface.
	FDP_ACC.1	Mitigates the threat by enforcing access control on the APIs offered by the TOE.
	FDP_ACF.1	Mitigates the threat by defining the mechanism for how API access control is enforced.
	FMT_API_EXT.1	Mitigates the threat by allowing API templates to be defined against objects based on a set of defined rules.
	FMT_MOF.1/SDN	Mitigates the threat by limiting the ability to modify API functions.
	FMT_SMF.1/SDN	Mitigates the threat by identifying the management functions subject to access restrictions.
	FMT_SMR.2/SDN	Mitigates the threat by defining a role separation mechanism for administrators to enforce least privilege.
	FTP_ITC.1 (modified from ND PP)	Mitigates the threat by defining the trusted communications channel used for SDN functionality.

5.4 TOE Security Assurance Requirements

This PP-Module does not define any Security Assurance requirements. The SARs from the Base-PP must be satisfied.

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_NORTHBOUND_API	This threat represents the same attack scenario as T.UNAUTHORISED_ADMINISTRATOR_ACCESS and T.UNTRUSTED_COMMUNICATION_CHANNELS, but applies to the north-south interface that is outside the scope originally defined by the Base-PP.

6.1.3 Consistency of OE Objectives

This PP-Module does not define any environmental objectives beyond those already defined in the NDcPP.

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 4: 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	This SFR iterates the Base-PP to account for if the SDN controller has a different logging mechanism from the Base-PP use.
FDP_ACC.1	This SFR defines access control policies that are specific to the PP-Module .
FDP_ACF.1	This SFR defines access control policies to subjects and objects based on a set of rules and is specific to the PP-Module .
FMT_API_EXT.1	This SFR defines API templates and the set of rules they must follow, which is specific to this PP-Module .
FMT_MOF.1/SDN	This SFR iterates the Base-PP to add specific abilities of the API administrator to modify API function behavior.
FMT_SMF.1/SDN	This SFR iterates the Base-PP to add management functions for API access control and validity, which is not defined in the Base-PP .
FMT_SMR.2/SDN	This SFR iterates the Base-PP to define specific roles for the API user and administrator, which is not defined in the Base-PP .

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 SERs.

Appendix C - Extended Component Definitions

This appendix contains the definitions for all extended requirements specified in the PP-Module.

C.1 Extended Components Table

All extended components specified in the PP-Module are listed in this table:

Table 5: Extended Component Definitions	
Functional Class	Functional Components
Class FMT: Security Management	FMT_API_EXT Management of API Behavior

C.2 Extended Component Definitions

C.2.1 Class FMT: Security Management

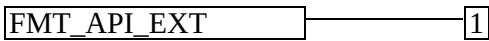
This PP-Module defines the following extended components as part of the class originally defined by CC Part 2:

C.2.1.1 FMT_API_EXT Management of API Behavior

Family Behavior

This family defines requirements for management of API behavior.

Component Leveling



FMT_API_EXT.1, Management of API Behavior, requires the TSF to define API templates and determine whether they meet specified rules.

Management: FMT_API_EXT.1

There are no management activities foreseen.

Audit: FMT_API_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package, or ST:

- Successful and failed definition of API template.

FMT_API_EXT.1 Management of API Behavior

Hierarchical to: No other components.

Dependencies to: FCS_CKM.6 Timing and Event of Cryptographic Key Destruction

FMT_API_EXT.1.1

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

FMT_API_EXT.1.2

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

Appendix D - Acronyms

Table 6: Acronyms	
Acronym	Meaning
API	Application Programming Interface
Base-PP	Base Protection Profile
CC	Common Criteria
CEM	Common Evaluation Methodology
cPP	Collaborative Protection Profile
DRBG	Deterministic Random Bit Generator
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
IT	Information Technology
OE	Operational Environment
OMB	Office of Management and Budget
OS	Operating System
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
REC	Request for Comment
SAR	Security Assurance Requirement
SDN	Software-Defined Networking
SFR	Security Functional Requirement

ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSE	TOE Security Functionality
TSEI	TSE Interface
TSS	TOE Summary Specification
USB	Universal Serial Bus
VM	Virtual Machine
VPN	Virtual Private Network
XOR	Exclusive Or

Appendix E - 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-2022-11-005, CC:2022, Revision 1, November 2022.
[GEM]	Common Methodology for Information Technology Security Evaluation - • Evaluation methodology, CCMB-2022-11-006, CC:2022, Revision 1, November 2022.
[NDcPP]	collaborative Protection Profile for Network Devices, Version 4.0, December 22, 2025