Assurance Activities Report

for

Palo Alto Networks PA-400 Series, PA-800 Series, PA-1400 Series, PA-3200 Series, PA-3400 Series, PA-5200 Series, PA-5400 Series, PA-5450, PA-7000 Series, and VM Series Next-Generation Firewall with PAN-OS 11.1

Version 1.0

18 September 2024

Prepared by:

Leidos Inc.
https://www.leidos.com/CC-FIPS140
Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive
Columbia, MD 21046

Prepared for:

Palo Alto Networks, Inc.
3000 Tannery Way
Santa Clara, CA 95054

© 2024 Leidos. All rights reserved.

The Developer of the TOE:

Palo Alto Networks, Inc.
3000 Tannery Way
Santa Clara, CA 95054

The TOE Evaluation was Sponsored by:

Palo Alto Networks, Inc.
3000 Tannery Way
Santa Clara, CA 95054

Evaluation Personnel:

Anthony Apted
Dawn Campbell
Justin Fisher
Josh Marciante
Armin Najafabadi
Kofi Owusu
Pascal Patin
Allen Sant
Srilekha Vangala
Kevin Zhang

© 2024 Leidos. All rights reserved.

Contents

© 2024 Leidos. All rights reserved.

© 2024 Leidos. All rights reserved.

© 2024 Leidos. All rights reserved.

1 Introduction

This document presents results from performing evaluation activities associated with the evaluation of Palo Alto Networks PA-400 Series, PA-800 Series, PA-1400 Series, PA-3200 Series, PA-3400 Series, PA-5200 Series, PA-5400 Series, PA-5450, PA-7000 Series, and VM Series Next-Generation Firewall with PAN-OS 11.1. The TOE claims conformance to the following CC specifications:

- PP-Configuration for Network Devices, Intrusion Prevention Systems, Stateful Traffic Filter Firewalls, and Virtual Private Network Gateways, Version 2.0, 25 April 2024 [CFG_NDcPP-IPS-FW-VPNGW_V2.0] consisting of the following components:
  - collaborative Protection Profile for Network Devices, Version 3.0e, 06 December 2023 [NDcPP]
  - PP-Module for Stateful Traffic Filter Firewalls, Version 1.4 + Errata 20200625, 25 June 2020 [FW-Module]
  - PP-Module for Virtual Private Network (VPN) Gateways, Version 1.3, 16 August 2023 [VPNGW-Module].
  - PP-Module for Intrusion Prevention System (IPS), Version 1.0, 11 May 2021 [IPS-Module]
  - Functional Package for Secure Shell (SSH) Version 1.0, 2021-05-13 [SSHPKG].

This report contains sections documenting the performance of evaluation activities associated with each of the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) as specified in the following:

- Evaluation Activities for Network Device cPP, Version 3.0e, 06-December 2023 [ND-SD]
- Supporting Document Mandatory Technical Document, Evaluation Activities for Stateful Traffic Filter Firewalls PP-Module, Version 1.4 + Errata 20200625, June-2020 [FW-SD]
- Supporting Document Mandatory Technical Document, PP-Module for VPN Gateways, Version 1.3, 2023-08-16 [VPNGW-SD]
- Supporting Document Mandatory Technical Document, PP-Module for Intrusion Prevention Systems (IPS), Version 1.0, 2021-05-11 [IPS-SD]
- Functional Package for Secure Shell (SSH) Version 1.0, 2021-05-13 [SSHPKG].

Note that, in accordance with NIAP Policy Letter #5, all cryptography in the TOE for which NIST provides validation testing of FIPS-approved and NIST-recommended cryptographic algorithms and their individual components must be NIST validated. The CCTL will verify that the claimed NIST validation complies with the NIAP-approved PP requirements the TOE claims to satisfy. The CCTL verification of the NIST validation will constitute performance of the associated assurance activity. As such, Test Activities associated with functional requirements within the scope of Policy Letter #5 are performed by verification of the relevant CAVP certification and not through performance of any testing as specified in the PP or its supporting document.

1.1 Technical Decisions

NIAP has issued the following Technical Decisions, applicable to [NDcPP], [ND-SD], [FW-Module], [FW-SD], [VPNGW-Module], [VPNGW-SD], [IPS-Module], [IPS-SD], and [SSHPKG]. Rationale is included for those Technical Decisions that do not apply to this evaluation.

© 2024 Leidos. All rights reserved.

Network Device cPP

TD0836: NIT Technical Decision: Redundant Requirements in FPT_TST_EXT.1

This TD is applicable to the TOE.

Firewall Module

TD0545: NIT Technical Decision for Conflicting FW rules cannot be configured (extension of Rfl#201837)

The TD updates test evaluation activities that apply to the TOE.

TD0551: NIT Technical Decision for Incomplete Mappings of OEs in FW Module v1.4+Errata

This TD is applicable to the TOE but does not affect the ST or evaluation activities. The TD updates Security Problem Definition mappings and rationale.

TD0827: Aligning MOD_CPP_FW_v1.4E with CPP_ND_V3.0E

This TD is applicable to the TOE.

VPN Gateway Module

TD0781: Correction to FIA_PSK_EXT.3 EA for MOD_VPNGW_v1.3

This TD does not apply to the TOE since the SFR is not claimed.

TD0811: Correction to Referenced SFR in FIA_PSK_EXT.3 Test

This TD does not apply to the TOE since the SFR is not claimed.

TD0824: Aligning MOD_VPNGW 1.3 with NDcPP 3.0E

This TD is applicable to the TOE.

TD0838: PPK Configurability in FIA_PSK_EXT.1.1

This TD is applicable to the TOE.

IPS Module

TD0595: Administrative corrections to IPS PP-Module

This TD is applicable to the TOE.

TD0722: IPS_SBD_EXT.1.1 EA Correction

This TD is applicable to the TOE.

TD0828: Aligning MOD_IPS_V1.0 with CPP_ND_V3.0E

This TD is applicable to the TOE.

SSH Package

TD0682: Addressing Ambiguity in FCS_SSHS_EXT.1 Tests

TD updates test evaluation activities that apply to the TOE.

© 2024 Leidos. All rights reserved.

TD0695: Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package

This TD is not applicable to the TOE because it supports both 128 and 256 bits for AES-CTR.

TD0732: FCS_SSHS_EXT.1.3 Test 2 Update

TD updates test evaluation activities that apply to the TOE.

TD0777: Clarification to Selections for Auditable Events for FCS_SSH_EXT.1

This TD is not applicable to the TOE because it logs failure to establish SSH connection.

1.2 Evidence

[ST] Palo Alto Networks PA-400 Series, PA-800 Series, PA-1400 Series, PA-3200 Series, PA-3400 Series, PA-5200 Series, PA-5400 Series, PA-5450, PA-7000 Series, and VM Series Next-Generation Firewall with PAN-OS 11.1 Security Target, Version 1.0, July 8, 2024

[CCECG] Palo Alto Networks Common Criteria Evaluated Configuration Guide (CCECG) for Next-Generation Firewall with PAN-OS 11.1, Revision Date: July 9, 2024

[ADMIN] PAN-OS® Administrator's Guide Version 11.1, Last Revised April 17, 2024

[CLI] PAN-OS CLI Quick Start Version 11.1, Last Revised October 20, 2023

[GUI] PAN-OS Web Interface Help Version 11.1, Last Revised March 13, 2024

[API] PAN-OS® and Panorama™ API Usage Guide Version 11.1, Last Revised December 6, 2023

[VM] VM-Series Deployment Guide Version 11.0, Last Revised November 14, 2022

Note: *The vendor confirmed this is the correct document reference and version—the manner in which the administrator deploys VM-Series virtual devices is identical for PAN-OS 11.0 and PAN-OS 11.1.*

[Test] Palo Alto PAN-OS v11.1 Common Criteria Test Report and Procedures For Network Device collaborative PP Version 3.0e, Version 1.0, July 11, 2024

[VA] Palo Alto Networks PA-400 Series, PA-800 Series, PA-1400 Series, PA-3200 Series, PA-3400 Series, PA-5200 Series, PA-5400 Series, PA-5450, PA-7000 Series, and VM Series Next-Generation Firewall with PAN-OS 11.1 Vulnerability Assessment, Version 1.2, September 17, 2024

[ALC] Palo Alto Networks PAN-OS 11.1, Panorama 11.1, WildFire 11.1 Flaw Remediation Procedures, Document Version 0.1, Revision Date November 14, 2023

1.3 Conformance Claims

Common Criteria Versions

- Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, Version 3.1, Revision 5, dated: April 2017.
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, dated: April 2017.
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Revision 5, dated: April 2017.

© 2024 Leidos. All rights reserved.

Common Evaluation Methodology Versions

- Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 3.1, Revision 5, dated: April 2017.

© 2024 Leidos. All rights reserved.

1.4 SAR Evaluation

The following Security Assurance Requirements (SARs) were evaluated during the evaluation of the TOE:

SAR	Verdict
ASE_CCL.1	Pass
ASE_ECD.1	Pass
ASE_INT.1	Pass
ASE_OBJ.1	Pass
ASE_REQ.1	Pass
ASE_SPD.1	Pass
ASE_TSS.1	Pass
ADV_FSP.1	Pass
AGD_OPE.1	Pass
AGD_PRE.1	Pass
ALC_CMC.1	Pass
ALC_CMS.1	Pass
ALC_FLR.3	Pass
ATE_IND.1	Pass
AVA_VAN.1	Pass

The evaluation work units are listed in the proprietary ETR. The evaluators note per the PP, PP-Modules, and Package evaluation activities that many of the SARs were successfully evaluated through completion of the associated evaluation activities present in the claimed PP, PP-Modules, and Package.

© 2024 Leidos. All rights reserved.

2 Security Functional Requirement Assurance Activities

This section describes the evaluation activities associated with the SFRs defined in the ST and the results of those activities as performed by the evaluation team. The evaluation activities are derived from [ND-SD], [FW-SD], [VPNGW-SD], [IPS-SD], and [SSHPKG] and modified by applicable NIAP Technical Decisions. Evaluation activities for SFRs not claimed by the TOE have been omitted.

Evaluator notes, such as changes made as a result of NIAP Technical Decisions, are highlighted in bold text, as are changes made as a result of NIAP Technical Decisions. Bold text is also used within evaluation activities to identify when they are mapped to individual SFR elements rather than the component level.

2.1 Security Audit (FAU)

2.1.1 FAU_GEN.1 Audit Data Generation

2.1.1.1 TSS Activities

For the administrative task of generating/import of, changing, or deleting of cryptographic keys as defined in FAU_GEN.1.1c, the TSS should identify what information is logged to identify the relevant key.

Section 6.1 of [ST] (“Security Audit”) states for identification of relevant keys in audit logs, the TOE logs the key name (or the certificate name if the key is embedded in a certificate or certificate request).

For distributed TOEs the evaluator shall examine the TSS to ensure that it describes which of the overall required auditable events defined in FAU_GEN.1.1 are generated and recorded by which TOE components. The evaluator shall ensure that this mapping of audit events to TOE components accounts for, and is consistent with, information provided in Table 1, as well as events in Tables 2, 4, and 5 (where applicable to the overall TOE). This includes that the evaluator shall confirm that all components defined as generating audit information for a particular SFR should also contribute to that SFR as defined in the mapping of SFRs to TOE components, and that the audit records generated by each component cover all the SFRs that it implements.

The TOE is not distributed so this evaluation activity is not applicable.

FW-SD

No additional Evaluation Activities are specified

2.1.1.2 Guidance Activities

The evaluator shall check the guidance documentation and ensure that it provides an example of each auditable event required by FAU_GEN.1 (i.e. at least one instance of each auditable event, comprising the mandatory, optional and selection-based SFR sections as applicable, shall be provided from the actual audit record).

Section 4 of [CCECG] (“Required Auditable Events”) identifies and provides the format for the logs stored in the Configuration Logs, the System Logs, and the Traffic and Threat Logs. A brief description of each field has been provided that contains the information required in FAU_GEN.1.2, and the additional information specified in the table of audit events. Taken together, these log files provide the security audit trail that satisfies the auditing requirements specified in the PPs to which the TOE claims conformance.

© 2024 Leidos. All rights reserved.

Section 4 of [CCECG] includes a table that provides an example of each auditable event required by FAU_GEN.1, including optional and selection-based SFRs included in the ST. Examples are provided for both the Web UI (HTTPS) and CLI (SSH). API calls (over HTTPS) used to make a configuration change will generate a log that is the same as those where the administrator logs in through the web interface. Since logs generated from the use of API calls are the same as the UI/Web configuration logs, examples of these are not provided. Note that the required auditable events for IPS are identified along with those for FMT_SMF.1/IPS, including the auditable events specified in [ST] Table 6 ("Auditable Events (IPS)").

The evaluator shall also make a determination of the administrative actions related to TSF data related to configuration changes. The evaluator shall examine the guidance documentation and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the cPP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are related to TSF data related to configuration changes. The evaluator may perform this activity as part of the activities associated with ensuring that the corresponding guidance documentation satisfies the requirements related to it.

The evaluator examined the supplied guidance documentation, identifying all mechanisms available to the administrator for configuring and managing the capabilities of the TOE. Those mechanisms related to the SFRs specified in the ST were identified and mapped to the applicable SFRs. In addition, the evaluator sought to confirm that all SFRs that would be expected to have a management capability related to them had appropriate management capabilities identified in the guidance documentation.

The administrative actions identified as auditable are:

- Login/Logout
- Resetting passwords
- Start and reboot TOE
- Set time
- Configure communication with external syslog
- Configure local audit settings
- Configure the authentication failure parameters for FIA_AFL.1
- Configure behavior of authentication failure lockout mechanism
- Enable and configure TLS/HTTPS/SSH
- Configure thresholds for SSH rekeying
- Create/Manage user accounts
- Configure local authentication
- Initiate and verify software updates
- Configure time interval of session inactivity
- Configure the login banner
- Configure firewall rules, stateful packet filtering, IPS functionality (signatures, lists, thresholds)
- Configure IPsec including the lifetime for IPsec SAs, reference identifier for peer
- Ability to generate Certificate Signing Request (CSR) and process CA certificate response
- Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1
- Configure the cryptographic functionality
- Configure X.509 certificate profiles

© 2024 Leidos. All rights reserved.

- Import X.509v3 certificates to the TOE's trust store
- Manage the TOE's trust store and designate X.509v3 certificates as trust anchor
- Ability to start and stop services
- Ability to manage the trusted public keys database.

The audit records for the administrator management actions are identified in Table 8 of [CCECG] ("Required Auditable Events"), in rows FAU_GEN.1, FMT_SMF.1, FPT_TUD_EXT.1, FFW_RUL_EXT.*, and FPF_RUL_EXT.*.

FW-SD

In addition to the Evaluation Activities specified in the Supporting Document for the Base-PP, the evaluator shall check the guidance documentation to ensure that it describes the audit records specified in Table 2 of the PP-Module in addition to those required by the Base-PP. If the optional SFR FFW_RUL_EXT.2 is claimed by the TOE, the evaluator shall also check the guidance documentation to ensure that it describes the relevant audit record specified in Table 3 of the PP-Module.

Section 4 of [CCECG] includes audit record examples for FFW_RUL_EXT.1: Application of rules configured with the 'log' operation; FFW_RUL_EXT.2: Establishment of a dynamic session (using App-ID); and configuring the firewall rules (FMT_SMF.1/FFW).

2.1.1.3 Test Activities

The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the table of audit events and administrative actions listed above. This should include all instances of an event: for instance, if there are several different identity and authentication (I&A) mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the guidance documentation, and that the fields in each audit record have the proper entries.

The evaluator performed actions, either independently or as part of another test activity, to generate all audit records for the events listed in the table of audit events and administrative actions listed above. The evaluator confirmed the audit records were generated in the format specified in guidance and that the fields in each audit record have the proper entries.

For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of auditable events to TOE components in the Security Target. For all events involving more than one TOE component when an audit event is triggered, the evaluator has to check that the event has been audited on both sides (e.g. failure of building up a secure communication channel between the two components). This is not limited to error cases but includes also events about successful actions like successful build up/tear down of a secure communication channel between TOE components.

Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly.

The TOE is not distributed and therefore this evaluation activity is not applicable.

© 2024 Leidos. All rights reserved.

FW-SD

In addition to the Evaluation Activities specified in the Supporting Document for the Base-PP, the evaluator shall perform tests to demonstrate that audit records are generated for the auditable events as specified in Table 2 of the PP-Module and, if the optional SFR FFW_RUL_EXT.2 is claimed by the TOE, Table 3.

The evaluator performed the actions described and observed that the audit records were generated in the described manner for each of the additional auditable events.

2.1.2 FAU_GEN.1/VPN Audit Data Generation (VPNGW-SD)

2.1.2.1 TSS Activities

The evaluator shall examine the TSS to verify that it describes the audit mechanisms that the TOE uses to generate audit records for VPN gateway behavior. If any audit mechanisms the TSF uses for this are not used to generate audit records for events defined by FAU_GEN.1 in the Base-PP, the evaluator shall ensure that any VPN gateway-specific audit mechanisms also meet the relevant functional claims from the Base-PP.

For example, FAU_STG_EXT.1 requires all audit records to be transmitted to the OE over a trusted channel. This includes the audit records that are required by FAU_GEN.1/VPN. Therefore, if the TOE has an audit mechanism that is only used for VPN gateway functionality, the evaluator shall ensure that the VPN gateway related audit records meet this requirement, even if the mechanism used to generate these audit records does not apply to any of the auditable events defined in the Base-PP.

Section 6.1 of [ST] (“Security Audit”) states the audit trail generated by the standalone TOE comprises several logs, which are locally stored in the TOE file system on the hard disk. The logs include Configuration, System, Traffic, and Threat logs. The TSS does not identify any separate audit mechanism for VPN gateway functionality specifically and therefore the remaining activities are not applicable.

2.1.2.2 Guidance Activities

The evaluator shall examine the operational guidance to verify that it identifies all security-relevant auditable events claimed in the ST and includes sample records of each event type. If the TOE uses multiple audit mechanisms to generate different sets of records, the evaluator shall verify that the operational guidance identifies the audit records that are associated with each of the mechanisms such that the source of each audit record type is clear.

Section 4 of [CCECG] (“Required Auditable Events”) identifies and provides the format for the logs stored in the Configuration Logs, the System Logs, and the Traffic and Threat Logs. A brief description of each field has been provided that contains the information required in FAU_GEN.1.2, and the additional information specified in the table of audit events. Taken together, these log files provide the security audit trail that satisfies the auditing requirements specified in the PPs to which the TOE claims conformance.

Section 4 of [CCECG] includes a table that provides an example of each auditable event required by FAU_GEN.1, including optional and selection-based SFRs included in the ST.

Section 4 of [CCECG] states the audit trail consists of the following log files:

- Configuration logs—record events such as when an administrator configures the security policies
- System logs—record user login and logout, system, and session information
- Traffic logs—record traffic flow events and information

© 2024 Leidos. All rights reserved.

- Threat logs—record detection and blocking of threats.

The information in [CCECG] is sufficient to identify the audit records associated with each of the log files, such that the source of each record type is clear.

2.1.2.3 Test Activities

The evaluator shall test the audit functionality by performing actions that trigger each of the claimed audit events and verifying that the audit records are accurate and that their format is consistent with what is specified in the operational guidance. The evaluator may generate these audit events as a consequence of performing other tests that would cause these events to be generated.

The evaluator observed through other testing of [VPNGW-Module] requirements that the TOE can generate all auditable events required by the PP-Module.

2.1.3 FAU_GEN.1/IPS Audit Data Generation (IPS-SD)

2.1.3.1 TSS Activities

The evaluator shall verify that the TSS describes how the TOE can be configured to log IPS data associated with applicable policies.

Section 6.11 of [ST] ("Intrusion Prevention") states the TOE supports three types of rules associated with its IPS functionality: security policy rules (i.e., firewall rules); L3 & L4 header rules; and Vulnerability/Threat signature rules. The Security Administrator can configure each type of rule and each type of rule includes logging options.

The evaluator shall verify that the TSS describes what (similar) IPS event types the TOE will combine into a single audit record along with the conditions (e.g., thresholds and time periods) for so doing. The TSS shall also describe to what extent (if any) that may be configurable.

Section 6.1 of [ST] ("Security Audit") states the TOE supports "suppression", which is used to log a fixed number of similar threats over a period of time once with repeat counter to indicate total number of threats. Log suppression, when enabled, is a feature that instructs the TOE to combine multiple similar logs into a single log entry on the Traffic or Threat Monitor page

For IPS_SBD_EXT.1, for each field, the evaluator shall verify that the TSS describes how the field is inspected and if logging is not applicable, any other mechanism such as counting that is deployed.

Section 6.11 of [ST] ("Intrusion Prevention") states the TOE supports string-based matching in header and payload via custom and predefined vulnerability signature rules. Every header field in IP/ICMP/TCP/UDP packets can be inspected. The vulnerability signature rules are assigned to a Vulnerability Protection profile, which is then associated with a security policy rule or rules. Each vulnerability signature rule must have an action (Allow, Alert, Reset Client, Reset Server, Reset Both, or Drop) defined and will trigger if the signature matches traffic that has otherwise been allowed by the security policy. Some attacks are always dropped, such as land, ping of death, teardrop, IP spoof, MAC spoof, and ICMP fragment attacks. These are tracked with counters per network interface.

© 2024 Leidos. All rights reserved.

2.1.3.2 Guidance Activities

The evaluator shall verify that the operational guidance describes how to configure the TOE to result in applicable IPS data logging.

Section 7.11 of [CCECG] ("Configure Threat Prevention") describes how to set the Action for a Vulnerability/Threat signature rule, which includes "default (alert)" and "alert". The description says to use "alert" in order to allow traffic and generate a log, and use "allow" to allow traffic without any logging. "Drop" will deny traffic and automatically generate a log.

The evaluator shall verify that the operational guidance provides instructions for any configuration that may be done in regard to logging similar events (e.g., setting thresholds, defining time windows, etc.).

Section 4 of [CCECG] ("Required Auditable Events") provides instructions for configuring log suppression, which is used to log a fixed number of similar threats over a period of time once with repeat counter to indicate total number of threats.

2.1.3.3 Test Activities

The evaluator shall test that the interfaces used to configure the IPS polices yield expected IPS data in association with the IPS policies. A number of IPS policy combination and ordering scenarios need to be configured and tested by attempting to pass both allowed and anomalous network traffic matching configured IPS policies in order to trigger all required IPS events.

Note the following:

- This activity should have been addressed with a combination of the Test EAs for the other IPS requirements.
- As part of testing this activity, the evaluator shall also ensure that the audit data generated to address this SFR can be handled in the manner that FAU_STG_EXT.1 requires for all audit data.

The evaluator performed actions associated with the test activities for the other IPS requirements and confirmed the TOE generated the expected IPS data. The evaluator confirmed the IPS audit records were generated in the format specified in guidance and that the fields in each audit record have the proper entries. The evaluator also confirmed the TOE handles the generated IPS audit events in the same manner as all other audit data.

2.1.4 FAU_GEN.2 User Identity Association

2.1.4.1 TSS Activities

The TSS and Guidance Documentation requirements for FAU_GEN.2 are already covered by the TSS and Guidance Documentation requirements for FAU_GEN.1.

2.1.4.2 Guidance Activities

The TSS and Guidance Documentation requirements for FAU_GEN.2 are already covered by the TSS and Guidance Documentation requirements for FAU_GEN.1.

© 2024 Leidos. All rights reserved.

2.1.4.3 Test Activities

This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

For distributed TOEs the evaluator shall verify that where auditable events are instigated by another component, the component that records the event associates the event with the identity of the instigator. The evaluator shall perform at least one test on one component where another component instigates an auditable event. The evaluator shall verify that the event is recorded by the component as expected and the event is associated with the instigating component. It is assumed that an event instigated by another component can at least be generated for building up a secure channel between two TOE components. If for some reason (could be e.g. TSS or Guidance Documentation) the evaluator would come to the conclusion that the overall TOE does not generate any events instigated by other components, then this requirement shall be omitted.

The TOE is not distributed so this evaluation activity is not applicable.

2.1.5 FAU_STG_EXT.1 Protected Audit Event Storage

2.1.5.1 TSS Activities

The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided.

Section 6.1 of [ST] (“Security Audit”) states the TOE can be configured to send generated audit records to an external syslog server using TLS or IPsec.

The evaluator shall examine the TSS to ensure it describes whether the TOE is a standalone TOE that stores audit data locally or a distributed TOE that stores audit data locally on each TOE component or a distributed TOE that contains TOE components that cannot store audit data locally on themselves but need to transfer audit data to other TOE components that can store audit data locally. The evaluator shall examine the TSS to ensure that for distributed TOEs it contains a list of TOE components that store audit data locally. The evaluator shall examine the TSS to ensure that for distributed TOEs that contain components which do not store audit data locally but transmit their generated audit data to other components it contains a mapping between the transmitting and storing TOE components.

Section 6.1 of [ST] states the audit trail is generated by the standalone TOE comprising logs, which are locally stored in the TOE file system on the hard disk.

As the TOE is not distributed, the remaining activities are not applicable.

The evaluator shall examine the TSS to ensure that it details whether the transmission of audit data to an external IT entity can be done in real-time, periodically, or both. In the case where the TOE is capable of performing transmission periodically, the evaluator shall verify that the TSS provides details about what event stimulates the transmission to be made as well as the possible acceptable frequency for the transfer of audit data.

Section 6.1 of [ST] states when the TOE is configured to send audit records to a syslog server, audit records are also written to the external syslog in real-time as they are written locally to the internal logs.

For distributed TOEs the evaluator shall examine the TSS to ensure it describes to which TOE components this SFR applies and how audit data transfer to the external audit server is implemented

© 2024 Leidos. All rights reserved.

among the different TOE components (e.g. every TOE components does its own transfer or the data is sent to another TOE component for central transfer of all audit events to the external audit server).

The TOE is not distributed, so this activity is not applicable.

The evaluator shall examine the TSS to ensure it describes the amount of audit data that can be stored locally and how these records are protected against unauthorized modification or deletion.

Section 6.1 of [ST] states the amount of locally-stored audit data is configured based on the percentage of available disk space that the administrator wants to allocate to audit records, with the total size of the audit storage being dependent on this setting as well as the hard drives that are being used by the TOE. By default, the TOE allocates 1-5% to System logs, 1-5% to Configuration logs, 20-35% to Traffic logs, and 10-20% to Threat logs. For example, for a 120GB drive, approximately 100GB is allocated for logging. Platform capabilities range from a limit of 3-4GB for the PA-410 (which has a 16GB flash drive) and up for the larger platforms (for example, PA-5220 has 1.70 TB drive). The absolute minimum on the smallest supported drive is 45MB. This section also states that when the audit storage is exhausted, the oldest records are automatically overwritten by the newest ones. The oldest records are also purged if the administrator reconfigures the amount of allocated audit storage to be insufficient to store all of the records that are currently present.

Section 6.1 of [ST] states audit records are protected from unauthorized access and removal by ensuring that only the pre-defined Audit Administrator role has access to them, and that there is no interface to modify the contents of the audit storage (other than deleting records) regardless of administrator privilege.

The evaluator shall examine the TSS to ensure it describes the method implemented for local logging, including format (e.g. buffer, log file, database) and whether the logs are persistent or non-persistent.

Section 6.1 of [ST] states the logs are stored locally in the TOE file system on the hard disk.

The evaluator shall examine the TSS to ensure it describes the conditions that must be met for authorized deletion of audit records.

Section 6.1 of [ST] states audit records are protected from unauthorized access and removal by ensuring that only the pre-defined Audit Administrator role has access to them.

The evaluator shall examine the TSS to ensure it details the behaviour of the TOE when the storage space for audit data is full. When the option ‘overwrite previous audit record’ is selected this description should include an outline of the rule for overwriting audit data. If ‘other actions’ are chosen such as sending the new audit data to an external IT entity, then the related behaviour of the TOE shall also be detailed in the TSS.

[ST] Section 6.1 states when the audit storage is exhausted, the oldest records are automatically overwritten by the newest ones. The oldest records are also purged if the administrator reconfigures the amount of allocated audit storage to be insufficient to store all of the records that are currently present.

For distributed TOEs the evaluator shall examine the TSS to ensure it describes which TOE components are storing audit information locally and which components are buffering audit information and forwarding the information to another TOE component for local storage. For every component the TSS shall describe the behaviour when local storage space or buffer space is exhausted.

© 2024 Leidos. All rights reserved.

The TOE is not distributed, so this activity is not applicable.

2.1.5.2 Guidance Activities

The evaluator shall also examine the guidance documentation to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server.

Section 6.8.1 of [CCECG] ("Syslog Server Connection Settings (Required)") states the TOE can be configured to forward generated audit records to an external syslog server in real-time. It provides guidance to configure the TOE to establish a trusted channel to the external syslog server in order to forward the audit records over the trusted channel. Guidance is provided for establishing a trusted channel using TLS v1.2 and alternatively for IKE/IPsec. The example in [CCECG] provides specific guidance for using the GUI and the CLI. Section 6.8.1 of [CCECG] also provides information on how to configure the external syslog server to receive the audit records from the TOE. The guidance recommends the use of syslog-ng version 3.7 or later for the external syslog server.

Section 6.7 of [CCECG] ("Configure Audit Settings (Required)") instructs the administrator to use TLS Session Logging, CA (OCSP/CRL) Session Establishment Logging, and IKE Session Establishment Logging.

The evaluator shall also examine the guidance documentation to ensure it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server.

Section 6.8.1 of [CCECG] states the TOE forwards generated audit records to an external syslog server in real-time. The TOE converts audit records and forwards them to the external syslog as it writes them to its local log files.

The evaluator shall examine the guidance documentation to ensure it describes any configuration required for protection of the locally stored audit data against unauthorized modification or deletion.

The TOE uses role-based access control (RBAC) to restrict access to the management functions, including the locally stored audit data, to the Administrator role. Section 7.3 of [CCECG] ("Role-Based Access Control (RBSC)") describes how every administrator must have a user account that specifies a role and authentication method. By default, every TOE appliance (PA-Series or VM-Series) has a predefined administrative account (admin) that provides full read-write access (superuser access) to all. Section 7 of [CCECG] ("Management Activity") identifies the audit management functions as restricted to the admin.

If the storage size is configurable, the evaluator shall review the Guidance Documentation to ensure it contains instructions on specifying the required parameters.

Section 4 of [CCECG] ("Required Auditable Events") describes how to configure the size of the log database and provides instructions and screenshots for specifying the required parameters.

If more than one selection is made for FAU_STG_EXT.1.5, the evaluator shall review the Guidance Documentation to ensure it contains instructions on specifying which action is performed when the local storage space is full.

© 2024 Leidos. All rights reserved.

The statement of FAU_STG_EXT.1.5 in [ST] specifies a single configuration—the TOE overwrites the oldest audit records first when the local storage space for audit data is full. Since there is only one selection made for the SFR, this activity is not applicable.

2.1.5.3 Test Activities

Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s).

The evaluator shall perform the following additional test for this requirement:

Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing. The evaluator shall verify that the TOE is capable of transferring audit data to an external audit server automatically without administrator intervention.

The evaluator configured the TOE to send audit records to a remote syslog server. The evaluator then observed the communication channel between the TOE and the syslog server. The evaluator verified that the audit records were not transmitted in the clear and were automatically being transmitted as events occurred.

Test 2: For distributed TOEs, Test 1 defined above shall be applicable to all TOE components that forward audit data to an external audit server.

The TOE is not distributed, so this activity is not applicable.

Test 3: The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall then make note of whether the TSS claims persistent or non-persistent logging and perform one of the following actions:

i. If persistent logging is selected, the evaluator shall perform a power cycle of the TOE and ensure that following power on operations the log events generated are still maintained within the local audit storage.

ii. If non-persistent logging is selected, the evaluator shall perform a power cycle of the TOE and ensure that following power on operations the log events generated are no longer present within the local audit storage.

The TSS claims persistent logging for the TOE. The evaluator performed a number of activities intended to generate audit records and confirmed the audit records were generated as expected. The evaluator then power-cycled the TOE, logged on to the TOE, and confirmed the TOE maintained the previously generated audit records in local audit storage.

Test 4: The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in FAU_STG_EXT.1.5. Depending on the configuration this means that the evaluator shall check the content of the audit data when the audit data is just filled to the maximum and then verifies that:

© 2024 Leidos. All rights reserved.

i. The audit data remains unchanged with every new auditable event that should be tracked but that the audit data is recorded again after the local storage for audit data is cleared (for the option 'drop new audit data' in FAU_STG_EXT.1.5).
ii. The existing audit data is overwritten with every new auditable event that should be tracked according to the specified rule (for the option 'overwrite previous audit records' in FAU_STG_EXT.1.5)
iii. The TOE behaves as specified (for the option 'other action' in FAU_STG_EXT.1.5).

The ST selects the option 'overwrite previous audit records' in FAU_STG_EXT.1.5. The evaluator performed actions that generated sufficient audit records to just fill the local storage space for data. The evaluator checked the content of the audit trail, then generated additional audit events. The evaluator then checked the content of the audit trail again and confirmed the TOE had overwritten the oldest audit records first.

Test 5: For distributed TOEs, for the local storage according to FAU_STG_EXT.1.4, Test 1 specified above shall be applied to all TOE components that store audit data locally. For all TOE components that store audit data locally and comply with FAU_STG_EXT.2, Test 2 specified above shall be applied. The evaluator shall verify that the transfer of audit data to an external audit server is implemented.

The TOE is not distributed, so this activity is not applicable.

Test 6 [Conditional]: In case manual export or ability to view locally is selected in FAU_STG_EXT.1.6, during interruption the evaluator shall perform a TSF-mediated action and verify the event is recorded in the audit trail.

The ST selects the option 'ability to view locally' in FAU_STG_EXT.1.6. The evaluator verified the TOE stores audit events locally and that stored audit records are accessible to be viewed by an administrator.

2.1.6 FAU_STG.1 Protected Audit Trail Storage

2.1.6.1 TSS Activities

The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally and how these records are protected against unauthorized modification or deletion. The evaluator shall ensure that the TSS describes the conditions that must be met for authorized deletion of audit records.

Section 6.1 of [ST] ("Security Audit") states the amount of locally-stored audit data is configured based on the percentage of available disk space that the administrator wants to allocate to audit records, with the total size of the audit storage being dependent on this setting as well as the hard drives that are being used by the TOE. By default, the TOE allocates 1-5% to System logs, 1-5% to Configuration logs, 20-35% to Traffic logs, and 10-20% to Threat logs. For example, for a 120GB drive, approximately 100GB is allocated for logging. Platform capabilities range from a limit of 3-4GB for the PA-410 (which has a 16GB flash drive) and up for the larger platforms (for example, PA-5220 has 1.70 TB drive). This section also states that when the audit storage is exhausted, the oldest records are automatically overwritten by the newest ones. The oldest records are also purged if the administrator reconfigures the amount of allocated audit storage to be insufficient to store all the records that are currently present.

Section 6.1 of [ST] states audit records are protected from unauthorized access and removal by ensuring that only the pre-defined Audit Administrator role has access to them, and that there is no interface to

© 2024 Leidos. All rights reserved.

modify the contents of the audit storage (other than deleting records) regardless of administrator privilege.

For distributed TOEs the evaluator shall examine the TSS to ensure it describes to which TOE components this SFR applies and how local storage is implemented among the different TOE components (e.g. every TOE component does its own local storage or the data is sent to another TOE component for central local storage of all audit events).

The TOE is not distributed, so this activity is not applicable.

2.1.6.2 Guidance Activities

The evaluator shall examine the guidance documentation to determine that it describes any configuration required for protection of the locally stored audit data against unauthorized modification or deletion.

The TOE restricts access to the management functions, including the locally stored audit data, to the Administrator role using role-based access control (RBAC). Section 7.3 of [CCECG] ("Role-Based Access Control (RBAC)") describes how every administrator must have a user account that specifies a role and authentication method. By default, every TOE appliance (PA-Series or VM-Series) has a predefined administrative account (admin) that provides full read-write access (superuser access) to all. Section 7 of [CCECG] ("Management Activity") identifies the audit management functions as restricted to the admin.

2.1.6.3 Test Activities

Test 1: The evaluator shall attempt to access the audit trail without authentication as Security Administrator (either by authentication as a non-administrative user, if supported, or without authentication at all) and attempt to modify and delete the audit records. The evaluator shall verify that these attempts fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to access the audit trail can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

The evaluator logged on to the TOE as a user without administrative privileges. The evaluator viewed the audit trail and then attempted to clear the system logs. The evaluator confirmed the non-administrative user was unable to modify or delete audit records.

Test 2: The evaluator shall access the audit trail as an authenticated Security Administrator and attempt to delete the audit records (if supported by the TOE, and to the extent described in the TSS). The evaluator shall verify that these attempts succeed. The evaluator shall verify that only the records authorized for deletion are deleted.

The evaluator logged on to the TOE as a user with full administrative privileges. The evaluator viewed the audit trail and then attempted to clear the system logs. The evaluator confirmed the Security Administrator was able to delete the audit records as described in the TSS.

For distributed TOEs the evaluator shall perform test 1 and test 2 for each component that is defined by the TSS to be covered by this SFR.

© 2024 Leidos. All rights reserved.

The TOE is not distributed, so this activity is not applicable.

2.2 Cryptographic Support (FCS)

The following table lists the cryptographic functions supported by the TOE and associated SFRs, the specific algorithms that are claimed for these functions, and the relevant CAVP certificate validation lists and certificate numbers for each.

Functions	Standards	Certificates
Asymmetric key generation (FCS_CKM.1, FCS_CKM.1/IKE)		
RSA Schemes (2048, 3072, 4096 bits)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	**Appliances:** CAVP #A3453 RSA KeyGen (FIPS186-4) **VMs:** CAVP #A3454 RSA KeyGen (FIPS186-4)
ECC Schemes (NIST Curves P-256, P-384, P-521) ***Note:*** *P-521 is selected only in FCS_CKM.1/IKE, not in FCS_CKM.1.*	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	**Appliances:** CAVP #A3453 ECDSA KeyGen (FIPS186-4) **VMs:** CAVP #A3454 ECDSA KeyGen (FIPS186-4)
FFC Schemes ('safe-prime' groups (2048-bit MODP, 3072-bit MODP, 4096-bit MODP)	NIST SP 800-56A Revision 3; RFC 3526	**Appliances:** CAVP #A3453 Safe Primes Key Generation **VMs:** CAVP #A3454 Safe Primes Key Generation
Key Establishment (FCS_CKM.2)		
Elliptic curve-based scheme (ECDSA) NIST P-256, P-384, P-521	NIST Special Publication 800-56A Revision 3	**Appliances:** CAVP #A3453 KAS-ECC-SSC-Sp800-56Ar3 **VMs:** CAVP #A3454 KAS-ECC-SSC-Sp800-56Ar3
FFC Schemes using 'safe-prime' groups (2048-bit MODP, 3072-bit MODP, 4096-bit MODP)	NIST Special Publication 800-56A Revision 3, RFC 3526	**Appliances:** CAVP #A3453 KAS-ECC-SSC-Sp800-56Ar3 **VMs:** CAVP #A3454 KAS-ECC-SSC-Sp800-56Ar3

© 2024 Leidos. All rights reserved.

Functions	Standards	Certificates
Encryption/Decryption (FCS_COP.1/DataEncryption)		
AES in CBC mode (128, 192, 256 bits)	AES as specified in ISO 18033-3 CBC as specified in ISO 10116	**Appliances:** CAVP #A3453 AES-CBC **VMs:** CAVP #A3454 AES-CBC
AES in CTR mode (128, 256 bits)	AES as specified in ISO 18033-3 CTR as specified in ISO 10116	**Appliances:** CAVP #A3453 AES-CTR **VMs:** CAVP #A3454 AES-CTR
AES in GCM mode (128, 256 bits)	AES as specified in ISO 18033-3 GCM as specified in ISO 19772	**Appliances:** CAVP #A3453 AES-GCM **VMs:** CAVP #A3454 AES-GCM
Cryptographic signature services (Signature Generation and Verification) (FCS_COP.1/SigGen)		
RSA Digital Signature Algorithm (rDSA) (2048, 3072, 4096-bit modulus)	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	**Appliances:** CAVP #A3453 RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4) **VMs:** CAVP #A3454 RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)
ECDSA (NIST curves P-256, P-384, P-521)	ECDSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves”	**Appliances:** CAVP #A3453 ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4) **VMs:** CAVP #A3454 ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4)

© 2024 Leidos. All rights reserved.

Functions	Standards	Certificates
Cryptographic hashing (FCS_COP.1/Hash)		
SHA-1 (digest size 160 bits) SHA-256 (digest size 256 bits) SHA-384 (digest size 384 bits) SHA-512 (digest size 512 bits)	ISO/IEC 10118-3:2004	**Appliances:** CAVP #A3453 SHA-1, SHA2-256, SHA2-384, SHA2-512 **VMs:** CAVP #A3454 SHA-1, SHA2-256, SHA2-384, SHA2-512
Keyed-hash message authentication (FCS_COP.1/KeyedHash)		
HMAC-SHA-1 (key/digest sizes 160 bits) HMAC-SHA-256 (key/digest sizes 256 bits) HMAC-SHA-384 (key/digest sizes 384 bits) HMAC-SHA-512 (key/digest sizes 512 bits)	ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2	**Appliances:** CAVP #A3453 HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-512 **VMs:** CAVP #A3454 HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-512
Random bit generation (FCS_RBG_EXT.1)		
CTR-DRBG (AES) – 256 bits entropy	ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions	**Appliances:** CAVP #A3453 Counter DRBG **VMs:** CAVP #A3454 Counter DRBG

2.2.1 FCS_CKM.1 Cryptographic Key Generation

2.2.1.1 TSS Activities

The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

Section 6.2 of [ST] ("Cryptographic Support") lists the key sizes and schemes supported by the TOE as follows: 2048/3072/4096-bit RSA; ECC with P-256/P-384/P-521; and Diffie-Hellman MODP-2048/MODP-3072/MODP-4096.

© 2024 Leidos. All rights reserved.

Upon review of the TSS, the evaluator identified the schemes are used for the following functions:

- X.509 key pair generation: ECDSA (256, 384), RSA (2048, 3072, 4096)
- SSH RSA key pair generation: RSA (2048, 3072, 4096)
- SSH: ECDSA (256, 384, 521)
- TLS: ECDSA (256, 384, 521), FFC (2048 or greater)
- IPsec: FFC MODP-2048, MODP-3072, MODP-4096 (corresponding to Groups 14, 15, and 16), ECDSA NIST curves P-256, P-384, P-521) (corresponding to Groups 19, 20, and 21).

Upon review of section 6.2 and section 6.3 of [ST], it is also clear that the RSA and ECC key generation schemes are used in support of key generation for X.509 certificate requests.

2.2.1.2 Guidance Activities

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all cryptographic protocols defined in the Security Target.

Section 6 of [CCECG] (“Evaluated Configuration”) states “HTTPS, IKE/IPsec, SSH and TLS connection settings (TLS ciphersuites, IKE/IPsec algorithms, SSH key exchange algorithms, key sizes, etc.) are configured or restricted automatically when FIPS-CC mode is enabled. For the remaining settings such as SSH encryption and rekey, please follow the guide in sections 6.4 and 6.5. While not required by the NDcPP, the administrator should configure the Permitted IP feature to restrict which computers can access the TOE and from specific IP addresses.”

Section 6.2 of [CCECG] (“Enable FIPS-CC Mode (Required)”) states the administrator must enable FIPS-CC mode. In this mode, the TOE restricts SSH key exchange algorithms and the TLS version and cipher suites (including elliptical curves) to the approved ones claimed in [ST]. The TLS ciphersuites are negotiated based on the public key algorithm (RSA vs ECDSA) in the TLS certificate and the TLS version(s) supported in the SSL/TLS Service Profile (TLSv1.2 vs TLSv1.3 [SHA-256 and SHA-384]).

Section 6.6 of [CCECG] (“Configure SSH Public-Key Authentication (Recommended)”) provides procedures to configure an administrative login using SSH Public-key Authentication, including generating a key pair.

Section 6.8.2 of [CCECG] (“Certificate-Based Authentication for Web UI (Optional)”) provides procedures to configure certificate-based authentication for administrator accounts (including use of CAC), while

Section 6.8.1 of [CCECG] (“Syslog Server Connection Settings (Required)”) describes how the administrator can specify the algorithms and associated key size for an asymmetric key pair for use in digital certificates.

2.2.1.3 Test Activities

Key Generation for FIPS PUB 186-4 RSA Schemes Performed in accordance with NIAP Policy Letter #5. **Key Generation for Elliptic Curve Cryptography (ECC)** Performed in accordance with NIAP Policy Letter #5. **Key Generation for Finite Field Cryptography (FFC)** Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] identifies the CAVP certifications verifying asymmetric key generation, as follows.

© 2024 Leidos. All rights reserved.

Algorithm	Tested Capabilities	Certificates
RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3.	Key Generation Mode: B.3.6 Properties: Modulo: 2048 Primality Tests: C.2 Properties: Modulo: 3072 Primality Tests: C.2 Properties: Modulo: 4096 Primality Tests: C.2 Public Exponent Mode: Fixed Fixed Public Exponent: 010001 Public Key Format: Standard	**Appliances:** CAVP #A3453 RSA KeyGen (FIPS186-4) **VMs:** CAVP #A3454 RSA KeyGen (FIPS186-4)
ECC schemes using “NIST curves” P-256, P-384, P-521, that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4. **Note:** P-521 is included here although it is selected only in FCS_CKM.1/IKE, not in FCS_CKM.1.	Curve: P-256, P-384, P-521 Secret Generation Mode: Testing Candidates	**Appliances:** CAVP #A3453 ECDSA KeyGen (FIPS186-4) ECDSA KeyVer (FIPS186-4) **VM:** CAVP #A3454 ECDSA KeyGen (FIPS186-4) ECDSA KeyVer (FIPS186-4)

FFC Schemes using “safe-prime” groups
Testing for FFC Schemes using safe-prime groups is done as part of testing in CKM.2.1.

2.2.2 FCS_CKM.1/IKE Cryptographic Key Generation (for IKE Peer Authentication) (VPNGW-SD)

2.2.2.1 TSS Activities

The evaluator shall check to ensure that the TSS describes how the key-pairs are generated. In order to show that the TSF implementation complies with FIPS PUB 186-4, the evaluator shall ensure that the TSS contains the following information:

- The TSS shall list all sections of Appendix B to which the TOE complies
- For each applicable section listed in the TSS, for all statements that are not "shall" (that is, "shall not," "should," and "should not"), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "shall not" or "should not" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE
- For each applicable section of Appendix B, any omission of functionality related to "shall" or “should” statements shall be described

© 2024 Leidos. All rights reserved.

Any TOE-specific extensions, processing that is not included in the Appendices, or alternative implementations allowed by the Appendices that may impact the security requirements the TOE is to enforce shall be described.

Section 6.2 of [ST] ("Cryptographic Support"), Table 9 ("FIPS 186-4 Conformance") describes the TOE's compliance to FIPS 186-4. Each section of Appendix B that the TOE complies with is listed along with "should" and "shall not" statements. The table indicates that each is implemented according to the standard and there are no deviations in the TOE's implementation. The rationale preceding the table indicates that the TOE fulfills all FIPS PUB 186-4 requirements without extensions. The TOE implements all "should" statements and complies with all "shall not" statements.

2.2.2.2 Guidance Activities

The evaluator shall check that the operational guidance describes how the key generation functionality is invoked, and describes the inputs and outputs associated with the process for each signature scheme supported. The evaluator shall also check that guidance is provided regarding the format and location of the output of the key generation process.

The "Certificate Management" section of [ADMIN], "Obtain Certificates" sub-section, "Generate a Certificate" topic, describes how the administrator can specify the algorithm and associated key size for generating an asymmetric key pair for use in digital certificates, and describes the format and location of the output of the key generation process.

Section 7.10 of [CCECG] ("Configure IKE/IPsec VPN Gateway") describes how the administrator configures the TOE for establishing IKE and IPsec connections, including all allowed algorithms and associated key sizes.

2.2.2.3 Test Activities

For FFC Schemes using "safe-prime" groups:

Testing for FFC Schemes using safe-prime groups is done as part of testing in FCS_CKM.2.

For all other selections:

The evaluator shall perform the corresponding tests for FCS_CKM.1 specified in the NDcPP SD, based on the selections chosen for this SFR. If IKE key generation is implemented by a different algorithm than the NDcPP key generation function, the evaluator shall ensure this testing is performed using the correct implementation.

See results for Key Generation for FIPS PUB 186-4 RSA Schemes and Key Generation for Elliptic Curve Cryptography (ECC) in FCS_CKM.1 above.

2.2.3 FCS_CKM.2 Cryptographic Key Establishment

2.2.3.1 TSS Activities

The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme. It is sufficient to provide the scheme, SFR, and service in the TSS.

© 2024 Leidos. All rights reserved.

The intent of this activity is to be able to identify the scheme being used by each service. This would mean, for example, one way to document scheme usage could be as shown in the table below. The information provided in this example does not necessarily have to be included as a table but can be presented in other ways as long as the necessary data is available.

Scheme	SFR	Service
RSA	FCS_TLSS_EXT.1	Administration
ECDH	FCS_IPSEC_EXT.1	Authentication Server

Section 6.2 of [ST] ("Cryptographic Support") identifies the TOE uses the following key establishment schemes for each cryptographic service mapped to a claimed protocol:

- SSH: ECC (256, 384, 521)
- TLS: ECC (256, 384, 521), FFC (2048 or greater)
- IKE/IPsec: FFC DH Groups 14, 15, 16, ECC (256, 384, 521) corresponding to Groups 19, 20, and 21

The supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1.

2.2.3.2 Guidance Activities

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment scheme(s).

Section 6 of [CCECG] ("Evaluated Configuration") states that for HTTPS, IKE/IPsec, SSH and TLS connection settings, the TLS cipher suites, IKE/IPsec algorithms, SSH key exchange algorithms, key sizes, etc. are configured automatically when FIPS-CC mode is enabled. FIPS-CC Mode is identified as required and the instructions to enable the mode are provided in Section 6.2 of [CCECG] ("Enable FIPS-CC Mode (Required)").

Section 7.10 of [CCECG] ("Configure IKE/IPsec VPN Gateway") describes how the administrator configures the TOE for establishing IKE and IPsec connections, including all allowed algorithms and associated key sizes.

Section 7.2 lists the cipher suites supported for GlobalProtect.

Section 6.8.1 of [CCECG] ("Syslog Server Connection Settings (Required)") describes how to generate valid certificates and identifies the TLS cipher suites used for the syslog server connection.

Section 6.2 of [CCECG] states the TLS cipher suites the TOE negotiates are based on the public key algorithm (RSA vs ECDSA) in the TLS certificate and the TLS version(s) supported in the SSL/TLS Service Profile (TLSv1.2 and TLSv1.3 [SHA-256 and SHA-384]).

2.2.3.3 Test Activities

Key Establishment Schemes

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment scheme(s).

ECC and FIPS 186-type SP800-56A Key Establishment Schemes

Performed in accordance with NIAP Policy Letter #5.

© 2024 Leidos. All rights reserved.

Section 6.2 of [ST] identifies the CAVP certifications verifying SP 800-56A key establishment schemes, as follows.

Algorithm	Tested Capabilities	Certificates
Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	Domain Parameter Generation Methods: P-256, P-384, P-521 Scheme: Ephemeral Unified: KAS Role: Initiator, Responder	**Appliances:** CAVP #A3453 KAS-ECC-SSC **VM:** CAVP #A3454 KAS-ECC-SSC

RSA-based Key Establishment

The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses RSAES-PKCS1-v1_5.

The ST does not select RSA-based key establishment schemes in FCS_CKM.2, so this activity is not applicable.

FFC Schemes using "safe-prime" groups

The evaluator shall verify the correctness of the TSF's implementation of safe-prime groups by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses safe-prime groups. This test must be performed for each safe-prime group that each protocol uses.

The only protocol which claims to utilize Safe-Primes is IPsec. The functionality of the IPsec channel was tested as part of FCS_IPSEC_EXT.1.4, where the TOE is shown to be able to successfully establish an IKE key exchange with a peer using the known good implementation of Safe-Prime groups, specifically 'StrongSwan' version 5.8.2. Additionally, the evaluator examined the ACVP certificate provided and observed that the TOE was tested against KAS-FFC-SSC SP800-56Ar3 algorithm by the ACVTS.

2.2.4 FCS_CKM.4 Cryptographic Key Destruction

2.2.4.1 TSS Activities

The evaluator shall examine the TSS to ensure it lists all relevant keys (describing the origin and storage location of each), all relevant key destruction situations (e.g. factory reset or device wipe function, disconnection of trusted channels, key change as part of a secure channel protocol), and the destruction method used in each case. For the purpose of this Evaluation Activity the relevant keys are those keys that are relied upon to support any of the SFRs in the Security Target. The evaluator shall confirm that the description of keys and storage locations is consistent with the functions carried out by the TOE (e.g. that all keys for the TOE-specific secure channels and protocols, or that support FPT_APW.EXT.1 and FPT_SKP_EXT.1, are accounted for[2]). In particular, if a TOE claims not to store plaintext keys in non-volatile memory then the evaluator shall check that this is consistent with the operation of the TOE.

© 2024 Leidos. All rights reserved.

Section 6.2 of [ST] ("Cryptographic Support") lists all keys/CSPs used by the TOE by their function (e.g., RSA/ECDSA private keys, SSH session keys) and describes their usage and composition. The description covers the cryptographic algorithm the key/CSP acts as input or output data for, how it is generated/used, the type of storage medium it resides on, and its method of destruction. In all cases, the key storage locations and destruction methods are consistent with the claims made in the SFR.

Section 6.2 of [ST] states all keys stored in volatile memory are zeroized following their use and describes the zeroization method consistent with the claims made in FCS_CKM.4.1. Specifically, plaintext key data in volatile memory is overwritten with a pseudo random pattern generated by an approved DRBG.

The evaluator shall check to ensure the TSS identifies how the TOE destroys keys stored as plaintext in non-volatile memory, and that the description includes identification and description of the interfaces that the TOE uses to destroy keys (e.g., file system APIs, key store APIs).

Section 6.2 of [ST] states the TOE does not store any plaintext private keys, secret keys, or other critical security parameters in non-volatile memory. The TOE stores persistent secret and private keys in encrypted form when not in use, using a 256-bit AES Key Encrypting Key (KEK) known as the Firmware Content Encryption Key, or Master Key. The KEK is not stored encrypted but is protected using Cryptod (Palo Alto's proprietary key storage module) and destroyed by the TOE's overwriting method.

Note that where selections involve 'destruction of reference' (for volatile memory) or 'invocation of an interface' (for non-volatile memory) then the relevant interface definition is examined by the evaluator to ensure that the interface supports the selection(s) and description in the TSS. In the case of non-volatile memory, the evaluator includes in their examination the relevant interface description for each media type on which plaintext keys are stored. The presence of OS-level and storage device-level swap and cache files is not examined in the current version of the Evaluation Activity.

The statement of FCS_CKM.4 does not select 'destruction of reference' or 'invocation of an interface' so this evaluation activity is not applicable to the TOE.

Where the TSS identifies keys that are stored in a non-plaintext form, the evaluator shall check that the TSS identifies the encryption method and the key-encrypting-key used, and that the key-encrypting-key is either itself stored in an encrypted form or that it is destroyed by a method included under FCS_CKM.4.

Section 6.2 of [ST] identifies the KEK as a 256-bit AES key that encrypts all key data stored in non-volatile memory. The KEK is protected using Cryptod and destroyed by the TOE's overwriting method.

The evaluator shall check that the TSS identifies any configurations or circumstances that may not conform to the key destruction requirement (see further discussion in the Guidance Documentation section below). Note that reference may be made to the Guidance Documentation for description of the detail of such cases where destruction may be prevented or delayed.

The evaluators examined the TSS and did not identify any exceptions to the behavior described by FCS_CKM.4.1.

Where the ST specifies the use of "a value that does not contain any CSP" to overwrite keys, the evaluator examines the TSS to ensure that it describes how that pattern is obtained and used, and that this justifies the claim that the pattern does not contain any CSPs.

© 2024 Leidos. All rights reserved.

The TSS does not claim any instances where key/CSP data is overwritten with a value that does not contain any CSP.

2.2.4.2 Guidance Activities

A TOE may be subject to situations that could prevent or delay key destruction in some cases. The evaluator shall check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS (and any other supporting information used). The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer.

For example, when the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-levelling and garbage collection. This may result in additional copies of the key that are logically inaccessible but persist physically. Where available, the TOE might then describe use of the TRIM command and garbage collection to destroy these persistent copies upon their deletion (this would be explained in TSS and Operational Guidance).

[Where TRIM is used then the TSS and/or guidance documentation is also expected to describe how the keys are stored such that they are not inaccessible to TRIM, (e.g. they would need not to be contained in a file less than 982 bytes which would be completely contained in the master file table).]

Section 6.2 of [CCECG] (“Enable FIPS-CC Mode (Required)”) states when FIPS-CC mode is enabled, all key destruction activities occur in the manner specified by FCS_CKM.4. To be in the evaluated configuration, the administrator must enable FIPS-CC Mode.

The [CCECG] does not define any circumstances that would cause key destruction to be delayed or prevented. The evaluators reviewed the TSS and test evidence and observed that no such cases should be expected.

2.2.4.3 Test Activities

None

2.2.5 FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/Decryption)

This SFR is modified by MOD_VPNGW_v1.3, but there are no modified or additional evaluation activities.

2.2.5.1 TSS Activities

The evaluator shall examine the TSS to ensure it identifies the key size(s) and mode(s) supported by the TOE for data encryption/decryption.

Section 6.2 of [ST] (“Cryptographic Support”) identifies that the TOE supports AES with key sizes of 128 bits and 256 bits in CBC, CTR, and GCM modes. This section also identifies that 192-bit CBC is supported for IPsec. The application note for FCS_COP.1.1/DataEncryption specifies that CBC is the only mode that supports 192-bit keys and that this is only used for IPsec, so the TSS is consistent with the SFR claim.

© 2024 Leidos. All rights reserved.

2.2.5.2 Guidance Activities

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected mode(s) and key size(s) defined in the Security Target supported by the TOE for data encryption/decryption.

Section 6.2 of [CCECG] ("Enable FIPS-CC Mode (Required)") describes how to enable FIPS-CC Mode on the TOE and states that it is required for the evaluated configuration. This process will restrict the TLS version and cipher suites to the approved ones claimed in the ST, including the allowed data encryption modes and key sizes.

Section 6.4 of [CCECG] ("Configure SSH Encryption and Integrity Algorithms (Required)") instructs the administrator how to configure the data encryption algorithms, including modes and key sizes, to be used by the TOE in SSH connections.

Section 7.10 of [CCECG] ("Configure IKE/IPsec VPN Gateway") describes how the administrator configures the TOE for establishing IKE and IPsec connections, including all allowed modes and key sizes for data encryption/decryption.

2.2.5.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] identifies the CAVP certifications verifying AES encryption and decryption, as follows.

Algorithm	Tested Capabilities	Certificates
AES as specified in ISO 18033-3, CBC as specified in ISO 10116	Direction: Decrypt, Encrypt Key Length: 128, 192, 256	**Appliances:** CAVP #A3453 AES-CBC **VM:** CAVP #A3454 AES-CBC
AES as specified in ISO 18033-3, GCM as specified in ISO 19772	Direction: Decrypt, Encrypt IV Generation: Internal Key Length: 128, 256	**Appliances:** CAVP #A3453 AES-GCM **VM:** CAVP #A3454 AES-GCM
AES as specified in ISO 18033-3, CTR as specified in ISO 10116	Direction: Decrypt, Encrypt Key Length: 128, 256	**Appliances:** CAVP #A3453 AES-CTR **VM:** CAVP #A3454 AES-CTR

© 2024 Leidos. All rights reserved.

2.2.6 FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)

2.2.6.1 TSS Activities

The evaluator shall examine the TSS to determine that it specifies the cryptographic algorithm and key size supported by the TOE for signature services.

Section 6.2 of [ST] ("Cryptographic Support") states the TOE supports RSA (2048/3072/4096 bit modulus) and ECDSA (P-256/P-384/P-521 curves) for digital signatures.

2.2.6.2 Guidance Activities

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected cryptographic algorithm and key size defined in the Security Target supported by the TOE for signature services.

Section 6.2 of [CCECG] ("Enable FIPS-CC Mode (Required)") describes how to enable FIPS-CC Mode on the TOE and states that it is required for the evaluated configuration. This process will restrict the TLS version and cipher suites to the approved ones claimed in the ST, including the allowed cryptographic algorithms and key sizes used by the TOE for signature services.

Section 6.6 of [CCECG] ("Configure SSH Public-Key Authentication (Recommended)") instructs the administrator how to configure SSH public key authentication, including the public key algorithm and key sizes.

2.2.6.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] identifies the CAVP certifications verifying digital signature services, as follows.

Algorithm	Tested Capabilities	Certificates
RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSAPKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3.	RSA Signature Generation Signature Type: PKCS 1.5 Modulo: 2048 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512 Modulo: 3072 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512 Modulo: 4096 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512	**Appliances:** CAVP #A3453 RSA SigGen (186-4) **VM:** CAVP #A3454 RSA SigGen (186-4)

© 2024 Leidos. All rights reserved.

Algorithm	Tested Capabilities	Certificates
	Signature Type: PKCSPSS Modulo: 2048 Hash: SHA2-256; Salt Length: 32 Hash: SHA2-384; Salt Length: 48 Hash: SHA2-512; Salt Length: 64 Modulo: 3072 Hash: SHA2-256; Salt Length: 32 Hash: SHA2-384; Salt Length: 48 Hash: SHA2-512; Salt Length: 64 Modulo: 4096 Hash: SHA2-256; Salt Length: 32 Hash: SHA2-384; Salt Length: 48 Hash: SHA2-512; Salt Length: 64	
	RSA Signature Verification Signature Type: PKCS 1.5 Modulo: 2048 Hash Algorithm: SHA1 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512 Modulo: 3072 Hash Algorithm: SHA1 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512 Modulo: 4096 Hash Algorithm: SHA1 Hash Algorithm: SHA2-256 Hash Algorithm: SHA2-384 Hash Algorithm: SHA2-512	**Appliances:** CAVP #A3453 RSA SigVer (186-4) **VM:** CAVP #A3454 RSA SigVer (186-4)

© 2024 Leidos. All rights reserved.

Algorithm	Tested Capabilities	Certificates
	Signature Type: PKCPSS Modulo: 2048 Hash: SHA1; Salt Length: 20 Hash: SHA2-256; Salt Length: 32 Hash: SHA2-384; Salt Length: 48 Hash: SHA2-512; Salt Length: 64 Modulo: 3072 Hash: SHA1; Salt Length: 20 Hash: SHA2-256; Salt Length: 32 Hash: SHA2-384; Salt Length: 48 Hash: SHA2-512; Salt Length: 64 Modulo: 4096 Hash: SHA1; Salt Length: 20 Hash: SHA2-256; Salt Length: 32 Hash: SHA2-384; Salt Length: 48 Hash: SHA2-512; Salt Length: 64	
ECDSA schemes using "NIST curves" P-256, P-384 and P-521 that meet the following: ISO/IEC 14888-3, Section 6.4.	ECDSA Signature Generation Curve: P-256, P-384, P-521 Hash Algorithm: SHA2-256, SHA2-384, SHA2-512 ECDSA Signature Verification Curve: P-256, P-384, P-521 Hash Algorithm: SHA-1, SHA2-256, SHA2-384, SHA2-512	**Appliances:** CAVP #A3453 ECDSA SigGen (186-4) ECDSA SigVer (186-4) **VM:** CAVP #A3454 ECDSA SigGen (186-4) ECDSA SigVer (186-4)

2.2.7 FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

2.2.7.1 TSS Activities

The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS.

Sections 6.2 ("Cryptographic Support") and 6.6 ("Protection of the TSF") of [ST] indicate that the hash function is associated with digital signature generation/verification, with HMAC, with software integrity verification, with IKE peer authentication, and with protection of user passwords.

2.2.7.2 Guidance Activities

The evaluator shall check the AGD documents to determine that any configuration that is required to configure the required hash sizes is present.

Section 6 of [CCECG] ("Evaluated Configuration") states the connection settings for HTTPS, IKE/IPsec, SSH and TLS (comprising TLS cipher suites, IKE/IPsec algorithms, SSH key exchange algorithms, key sizes, etc.) are configured or restricted automatically when FIPS-CC mode is enabled. FIPS-CC Mode is identified as

© 2024 Leidos. All rights reserved.

required and the instructions to enable the mode are provided in Section 6.2 of [CCECG] (“Enable FIPS-CC Mode (Required)”).

Section 7.10 of [CCECG] (“Configure IKE/IPsec VPN Gateway”) describes how the administrator configures the TOE for establishing IKE and IPsec connections, including all allowed hash algorithms and associated digest sizes.

Section 6.8 of [CCECG] (“Secure Connection Settings”) describes how to generate valid certificates to be used for the syslog server connection and identifies the valid hash algorithms and digest sizes.

2.2.7.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] identifies the CAVP certifications verifying cryptographic hashing, as follows.

Algorithm	Tested Capabilities	Certificates
SHS as defined in ISO/IEC 10118-3:2004.	SHA-1 SHA-256 SHA-384 SHA-512	**Appliances:** CAVP #A3453 SHA-1, SHA2-256, SHA2-384, SHA2-512 **VM:** CAVP #A3454 SHA-1, SHA2-256, SHA2-384, SHA2-512

2.2.8 FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)

2.2.8.1 TSS Activities

The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used.

Table 8 of [ST] (“Cryptographic Functions”) includes a list of the HMAC functions that specifies the key length, block size, and digest size (which implicitly identifies the hash function and output MAC length).

2.2.8.2 Guidance Activities

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the values used by the HMAC function: key length, hash function used, block size, and output MAC length used defined in the Security Target supported by the TOE for keyed hash function.

Section 6.2 of [CCECG] (“Enable FIPS-CC Mode (Required)”) describes how to enable FIPS-CC Mode on the TOE and states that it is required for the evaluated configuration. This process will configure cryptographic parameters to ensure that only the required keyed-hash algorithms are used for trusted channel communications.

Section 7.10 of [CCECG] (“Configure IKE/IPsec VPN Gateway”) describes how the administrator configures the TOE for establishing IKE and IPsec connections, including all allowed authentication algorithms and associated digest sizes.

© 2024 Leidos. All rights reserved.

2.2.8.3 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] identifies the CAVP certifications verifying cryptographic keyed hashing, as follows.

Algorithm	Tested Capabilities	Certificates
HMAC that meets ISO/IEC 9797-2:2011.	HMAC-SHA1 MAC: 160 Key Length: 256-2048 Increment 8 HMAC-SHA2-256 MAC: 256 Key Length: 256-2048 Increment 8 HMAC-SHA2-384 MAC: 384 Key Length: 256-2048 Increment 8 HMAC-SHA2-512 MAC: 512 Key Length: 256-2048 Increment 8	**Appliances:** CAVP #A3453 HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-512 **VM:** CAVP #A3454 HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-512

2.2.9 FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

2.2.9.1 Evaluation Activity

Documentation shall be produced—and the evaluator shall perform the activities—in accordance with Appendix D of [NDcPP].

The vendor produced a proprietary Entropy Analysis Report (EAR) that the evaluators determined was suitable to meet the requirements specified in Appendix D of [NDcPP].

2.2.9.2 TSS Activities

The evaluator shall examine the TSS to determine that it specifies the DRBG type, identifies the entropy source(s) seeding the DRBG, and state the assumed or calculated min-entropy supplied either separately by each source or the min-entropy contained in the combined seed value.

Section 6.2 of [ST] (“Cryptographic Support”) states the TSF uses an AES-256 CTR_DRBG that receives entropy from a hardware source (identified in the proprietary EAR), and states that the min-entropy of the combined seed value is no less than 256 bits.

2.2.9.3 Guidance Activities

The evaluator shall confirm that the guidance documentation contains appropriate instructions for configuring the RNG functionality.

Section 6.2 of [CCECG] (“Enable FIPS-CC Mode (Required)”) states that enabling FIPS-CC mode configures the DRBG to use the algorithm claimed in [ST].

© 2024 Leidos. All rights reserved.

2.2.9.4 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] identifies the CAVP certification verifying deterministic random bit generation, as follows.

Algorithm	Tested Capabilities	Certificates
CTR_DRBG in accordance with ISO/IEC 18031:2011	Counter DRBG Mode: AES-256	**Appliances:** CAVP #A3453 Counter DRBG **VMs:** CAVP #A3454 Counter DRBG

2.2.10 FCS_HTTPS_EXT.1 HTTPS Protocol

2.2.10.1 TSS Activities

The evaluator shall examine the TSS and determine that enough detail is provided to explain how the implementation complies with RFC 2818.

Section 6.2 of [ST] ("Cryptographic Support") states the TOE's HTTPS protocol complies with RFC 2818 and is implemented using TLS 1.2 (RFC 5246) and TLS 1.3 (RFC 8446). It includes statements for each section of RFC 2818, sufficient to understand how the implementation complies with the RFC.

2.2.10.2 Guidance Activities

The evaluator shall examine the guidance documentation to verify it instructs the Administrator how to configure TOE for use as an HTTPS client or HTTPS server.

Section 6 of [CCECG] ("Evaluated Configuration") states the TOE by default supports only secure protocols (including HTTPS) for remote administrative access. It further states the TOE automatically configures TLS and HTTPS settings when the administrator enables FIPS-CC mode, which the administrator must do as part of the procedure for establishing the evaluated configuration. Section 6.2 of [CCECG] ("Enable FIPS-CC Mode (Required)") instructs the administrator how to enable FIPS-CC mode.

2.2.10.3 Test Activities

This test is now performed as part of FIA_X509_EXT.1/Rev testing. Tests are performed in conjunction with the TLS evaluation activities. If the TOE is an HTTPS client or an HTTPS server utilizing X.509 client authentication, then the certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1.

This test is performed in conjunction with FIA_X509_EXT.1/Rev and FCS_TLSS_EXT.1 evaluation activities.

© 2024 Leidos. All rights reserved.

2.2.11 FCS_SSH_EXT.1 SSH Protocol (SSHPKG)

2.2.11.1 TSS Activities

FCS_SSH_EXT.1.1
The evaluator shall ensure that the selections indicated in the ST are consistent with selections in this and subsequent components. Otherwise, this SFR is evaluated by activities for other SFRs.

The ST selects “server” and specifies compliance with RFCs 4251, 4252, 4253, 4254, 4256, 4344, 5647, 5656, 6668, 8308, and 8332.

FCS_SSH_EXT.1.2
The evaluator shall check to ensure that the authentication methods listed in the TSS are identical to those listed in this SFR component; and, ensure if password-based authentication methods have been selected in the ST then these are also described; and, ensure that if keyboard-interactive is selected, it describes the multifactor authentication mechanisms provided by the TOE.

Section 6.2 of [ST] identifies public-key (RSA), keyboard-interactive, and password-based authentication. The authentication methods listed in the TSS are identical to those listed in FCS_SSH_EXT.1.2.

Section 6.2 of [ST] states keyboard-interactive is a generic authentication method that can be used to implement different types of authentication mechanisms such as password and public-key, which are the authentication mechanisms provided by the TOE.

FCS_SSH_EXT.1.3
The evaluator shall check that the TSS describes how “large packets” are detected and handled.

Section 6.2 of [ST] describes a tracking mechanism limiting SSH packets to 262,105 bytes, where any packet over that size will be dropped (i.e., not processed farther and buffer containing the packet will be freed).

FCS_SSH_EXT.1.4
The evaluator will check the description of the implementation of SSH in the TSS to ensure the encryption algorithms supported are specified. The evaluator will check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component.

Section 6.2 of [ST] describes the supported AES encryption/decryption algorithms in CBC, CTR, or GCM mode with key sizes of 128 and 256 bits. No optional characteristics are supported.

FCS_SSH_EXT.1.5
The evaluator will check the description of the implementation of SSH in the TSS to ensure the hashing algorithms supported are specified. The evaluator will check the TSS to ensure that the hashing algorithms specified are identical to those listed for this component.

Section 6.2 of [ST] states the TOE supports HMAC-SHA-256, HMAC-SHA-512, and implicit MAC (aes128-gcm@openssh.com and aes256-gcm@openssh.com) for integrity and authenticity. These are identical to the list in the SFR component.

FCS_SSH_EXT.1.6

© 2024 Leidos. All rights reserved.

The evaluator will check the description of the implementation of SSH in the TSS to ensure the shared secret establishment algorithms supported are specified. The evaluator will check the TSS to ensure that the shared secret establishment algorithms specified are identical to those listed for this component.

Section 6.2 of [ST] states the TOE only supports key exchange algorithms ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521, identical to the list in the SFR component.

FCS_SSH_EXT.1.7

The evaluator will check the description of the implementation of SSH in the TSS to ensure the KDFs supported are specified. The evaluator will check the TSS to ensure that the KDFs specified are identical to those listed for this component.

Section 6.2 of [ST] states the TOE supports SSH KDF as specified in section of RFC 5656, which comprises the Elliptic Curve Diffie-Hellman (ECDH) key exchange method (i.e., ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521). This is identical to the KDFs listed in FCS_SSH_EXT.1.7 and consistent with the key exchange algorithms specified in FCS_SSH_EXT.1.6.

FCS_SSH_EXT.1.8

The evaluator shall check the TSS to ensure that if the TOE enforces connection rekey or termination limits lower than the maximum values that these lower limits are identified.

In cases where hardware limitation will prevent reaching data transfer threshold in less than one hour, the evaluator shall check the TSS to ensure it contains:

a. An argument describing this hardware-based limitation and
b. Identification of the hardware components that form the basis of such argument.

For example, if specific Ethernet Controller or Wi-Fi radio chip is the root cause of such limitation, these subsystems shall be identified.

Section 6.2 of [ST] states for each SSH session, the TOE initiates a new key exchange (rekey) when either a configurable amount of data (10 – 4000 MBs) or time (10 – 3600 seconds) has passed, whichever threshold occurs first. In the evaluated configuration, the administrator should not configure the SSH data rekey threshold to be more than 1024 MBs and the threshold limits apply to both transmitted and received data.

The ST does not identify any hardware limitations and therefore this activity is not applicable.

2.2.11.2 Guidance Activities

FCS_SSH_EXT.1.1

There are no guidance evaluation activities for this component. This SFR is evaluated by activities for other SFRs.

FCS_SSH_EXT.1.2

The evaluator shall check the guidance documentation to ensure the configuration options, if any, for authentication mechanisms provided by the TOE are described.

The ST identifies the authentication methods: password; keyboard-interactive; and publickey (ssh-rsa, rsa-sha2-256, rsa-sha2-512). Section 6.6 of [CCECG] (“Configure SSH Public-Key Authentication

© 2024 Leidos. All rights reserved.

(Recommended)") describes how to enable public key authentication for SSH and states that only RSA keypair of 2048 bits or higher is supported.

FCS_SSH_EXT.1.3
None listed.

FCS_SSH_EXT.1.4
The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Section 6.4 of [CCECG] ("Configure SSH Encryption and Integrity Algorithms (Required)") provides steps to configure the SSH encryption and integrity algorithms and states to configure AES 128 and 256 bits and hmac-sha2 only. It warns not to select hmac-sha-1 as a MAC algorithm.

FCS_SSH_EXT.1.5
The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Section 6.4 of [CCECG] provides instructions to configure hmac-sha2 only. In addition, AES-GCM has the integrity algorithm built-in so no other MAC algorithm is needed. This is the 'implicit' option in FCS_SSHS_EXT.1.5. This is consistent with the hmac-sha-256, hmac-sha-512, and implicit integrity algorithms supported.

FCS_SSH_EXT.1.6
The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

The SSH key exchange algorithms are not separately configurable; enabling FIPS-CC mode as described in section 6.2 of [CCECG] ("Enable FIPS-CC Mode (Required)") is sufficient to ensure this function is configured properly.

FCS_SSH_EXT.1.7
None listed.

FCS_SSH_EXT.1.8
The evaluator shall check the guidance documentation to ensure that if the connection rekey or termination limits are configurable, it contains instructions to the administrator on how to configure the relevant connection rekey or termination limits for the TOE.

Section 6.5 of [CCECG] ("Configure SSH Rekey Interval (Required)") states when FIPS-CC mode is enabled, the SSH rekeying will occur approximately at 1 hour of time or after 1 GB of data has been transmitted, whichever occurs first. Section 6.5 also describes how the administrator can configure the rekeying thresholds for time and amount of data and specifies the allowed values for the evaluated configuration (less than 1 hour for time and less than 1 Gb for data). Section 6.2 of [CCECG] instructs the administrator how to enable FIPS-CC mode.

© 2024 Leidos. All rights reserved.

2.2.11.3 Test Activities

FCS_SSH_EXT.1.1

There are no test evaluation activities for this component. This SFR is evaluated by activities for other SFRs.

FCS_SSH_EXT.1.2

Test 1: [conditional] If the TOE is acting as SSH Server:

a. The evaluator shall use a suitable SSH Client to connect to the TOE, enable debug messages in the SSH Client, and examine the debug messages to determine that only the configured authentication methods for the TOE were offered by the server.

The evaluator configured an SSH client to enable debug messages and connected to the TOE. The evaluator verified that the TOE offered only the public key, password, and keyboard-interactive authentication mechanisms as claimed in the ST.

b. [conditional] If the SSH server supports X509 based Client authentication options:
   a. The evaluator shall initiate an SSH session from a client where the username is associated with the X509 certificate. The evaluator shall verify the session is successfully established.
   b. Next the evaluator shall use the same X509 certificate as above but include a username not associated with the certificate. The evaluator shall verify that the session does not establish.
   c. Finally, the evaluator shall use the correct username (from step a above) but use a different X509 certificate which is not associated with the username. The evaluator shall verify that the session does not establish.

The TOE does not claim to support X.509 based client authentication, so this test activity is not applicable.

FCS_SSH_EXT.1.2

Test 2: [conditional] If the TOE is acting as SSH Client, the evaluator shall test for a successful configuration setting of each authentication method as follows:

a. The evaluator shall initiate a SSH session using the authentication method configured and verify that the session is successfully established.
b. Next, the evaluator shall use bad authentication data (e.g. incorrectly generated certificate or incorrect password) and ensure that the connection is rejected.

Steps a-b shall be repeated for each independently configurable authentication method supported by the server.

The TOE does not act as an SSH Client, so this test activity is not applicable.

FCS_SSH_EXT.1.2

Test 3: [conditional] If the TOE is acting as SSH Client, the evaluator shall verify that the connection fails upon configuration mismatch as follows:

a. The evaluator shall configure the Client with an authentication method not supported by the Server.

© 2024 Leidos. All rights reserved.

b. The evaluator shall verify that the connection fails.

If the Client supports only one authentication method, the evaluator can test this failure of connection by configuring the Server with an authentication method not supported by the Client. In order to facilitate this test, it is acceptable for the evaluator to configure an authentication method that is outside of the selections in the SFR.

The TOE does not act as an SSH Client, so this test activity is not applicable.

FCS_SSH_EXT.1.3

Test 1: The evaluator shall demonstrate that the TOE accepts the maximum allowed packet size.

The evaluator used a proprietary testing tool to send a packet of the maximum size permitted by the TOE. The evaluator verified that the TOE accepted this packet.

Modified by TD0732

FCS_SSH_EXT.1.3

Test 2: This test is performed to verify that the TOE drops packets that are larger than size specified in the component.

a. The evaluator shall establish a successful SSH connection with the peer.

b. Next the evaluator shall craft a packet that is ~~**one byte**~~ **slightly** larger than the maximum size specified in this component and send it through the established SSH connection to the TOE. **The packet should not be greater than the maximum packet size + 16 bytes. If the packet is larger, the evaluator shall justify the need to send a larger packet.**

c. **The e**valuator shall verify that the packet was dropped by the TOE**. The method of verification will vary by the TOE. Examples include ~~by~~** reviewing the TOE audit log for a dropped packet audit **or observing the TOE terminates the connection**.

The evaluator utilized a proprietary testing tool to send a packet just larger than the maximum packet size. The evaluator observed that the TOE terminated the connection.

FCS_SSH_EXT.1.4

If the TOE can be both a client and a server, these tests must be performed for both roles.

Test 1: The evaluator must ensure that only claimed algorithms and cryptographic primitives are used to establish an SSH connection. To verify this, the evaluator shall establish an SSH connection with a remote endpoint. The evaluator shall capture the traffic exchanged between the TOE and the remote endpoint during protocol negotiation (e.g. using a packet capture tool or information provided by the endpoint, respectively). The evaluator shall verify from the captured traffic that the TOE offers only the algorithms defined in the ST for the TOE for SSH connections. The evaluator shall perform one successful negotiation of an SSH connection and verify that the negotiated algorithms were included in the advertised set. If the evaluator detects that not all algorithms defined in the ST for SSH are advertised by the TOE or the TOE advertises additional algorithms not defined in the ST for SSH, the test shall be regarded as failed.

The data collected from the connection above shall be used for verification of the advertised hashing and shared secret establishment algorithms in FCS_SSH_EXT.1.5 and FCS_SSH_EXT.1.6 respectively.

The evaluator established an SSH connection while capturing packets. The evaluator inspected the packet capture and verified that only the aes128-cbc, aes256-cbc, aes128-ctr, aes256-ctr, aes128-

© 2024 Leidos. All rights reserved.

gcm@openssh.com, and aes256-gcm@openssh.com encryption algorithms were included in the advertisement, as specified in the ST.

FCS_SSH_EXT.1.4

If the TOE can be both a client and a server, these tests must be performed for both roles.

Test 2: For the connection established in Test 1, the evaluator shall terminate the connection and observe that the TOE terminates the connection.

The evaluator issued the 'exit' command on the connection established in Test 1 and verified that the session was terminated.

FCS_SSH_EXT.1.4

If the TOE can be both a client and a server, these tests must be performed for both roles.

Test 3: The evaluator shall configure the remote endpoint to only allow a mechanism that is not included in the ST selection. The evaluator shall attempt to connect to the TOE and observe that the attempt fails.

The evaluator configured an SSH client to allow only the non-selected aes192-cbc cipher. The evaluator attempted to connect to the TOE and verified that the connection failed.

FCS_SSH_EXT.1.5

Test 1: The evaluator shall use the test data collected in FCS_SSH_EXT.1.4, Test 1 to verify that appropriate mechanisms are advertised.

The evaluator inspected the test data collected in FCS_SSH_EXT.1.4 Test 1 and verified that only the hmac-sha2-256 and hmac-sha2-512 MAC algorithms were present, as claimed in the ST.

FCS_SSH_EXT.1.5

Test 2: The evaluator shall configure an SSH peer to allow only a hashing algorithm that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection and observe that the connection is rejected.

The evaluator configured an SSH peer to allow only the non-claimed hmac-sha1 MAC algorithm. The evaluator verified that the connection failed.

FCS_SSH_EXT.1.6

Test 1: The evaluator shall use the test data collected in FCS_SSH_EXT.1.4, Test 1 to verify that appropriate mechanisms are advertised.

The evaluator inspected the data captured in the evidence for FCS_SSH_EXT.1.4 Test 1 and verified that only the ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp512 kex algorithms were advertised, as claimed in the ST

© 2024 Leidos. All rights reserved.

FCS_SSH_EXT.1.6

Test 2: The evaluator shall configure an SSH peer to allow only a key exchange method that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection and observe that the connection is rejected.

The evaluator configured an SSH peer to allow only the non-claimed diffie-hellman-group1-sha1 key exchange method. The evaluator verified that the connection failed.

FCS_SSH_EXT.1.7

None listed.

FCS_SSH_EXT.1.8

The test harness needs to be configured so that its connection rekey or termination limits are greater than the limits supported by the TOE -- it is expected that the test harness should not be initiating the connection rekey or termination.

Test 1: Establish an SSH connection. Wait until the identified connection rekey limit is met. Observed that a connection rekey or termination is initiated. This may require traffic to periodically be sent, or connection keep alive to be set, to ensure that the connection is not closed due to an idle timeout.

The evaluator configured the TOE with a 3-minute rekey threshold and a 10MB data rekey threshold. The evaluator used a modified SSH client to connect to the TOE and started a timer awaiting a rekey while capturing packets. The evaluator observed that after 3 minutes had passed, the SSH client indicated a rekey had occurred.

Test 2: Establish an SSH connection. Transmit data from the TOE until the identified connection rekey or termination limit is met. Observe that a connection rekey or termination is initiated.

The evaluator used a proprietary tool to execute a command repeatedly in an SSH session to receive data from the TOE. The evaluator selected a command that would return a large amount of data. The evaluator observed that when the TOE had sent 10 MB in the established session that a rekey occurred.

Test 3: Establish an SSH connection. Send data to the TOE until the identified connection rekey limit or termination is met. Observe that a connection rekey or termination is initiated.

The evaluator used a proprietary tool to execute a command repeatedly in an SSH session to receive data from the TOE. The evaluator selected a command that would return a large amount of data. The evaluator observed that when the TOE had sent 10 MB in the established session that a rekey occurred.

2.2.12 FCS_SSHS_EXT.1 SSH Server Protocol (SSHPKG)

2.2.12.1 TSS Activities

No activities.

© 2024 Leidos. All rights reserved.

2.2.12.2 Guidance Activities

The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Section 6.4 of [CCECG] (“Configure SSH Encryption and Integrity Algorithms (Required)”) together with section 6.2 of [CCECG] (“Enable FIPS-CC Mode (Required)”) provides the instructions the administrator needs to ensure only the allowed mechanisms are used in SSH connection with the TOE.

2.2.12.3 Test Activities

Modified by TD0682

The evaluator shall ~~**repeat Test 1 and Test 2 from FCS_SSH_EXT.1.4 for each of the authentication mechanisms supported by the TOE.**~~ perform the following tests:

Test 1: The evaluator shall use a suitable SSH Client to connect to the TOE and examine the list of server host key algorithms in the SSH_MSG_KEXINIT packet sent from the server to the client to determine that only the configured server authentication methods for the TOE were offered by the server.

Test 2: The evaluator shall test for a successful configuration setting of each server authentication method as follows. The evaluator shall initiate a SSH session using the authentication method configured and verify that the session is successfully established. Repeat this process for each independently configurable server authentication method supported by the server.

Test 3: ~~**Next**~~ The evaluator shall configure the ~~**remote**~~ peer to only allow an authentication mechanism that is not included in the ST selection. The evaluator shall attempt to connect to the TOE and observe that the ~~**attempt fails**~~ **TOE sends a disconnect message**.

The evaluator connected to the TOE using an SSH client while capturing packets. The evaluator observed that only the rsa-sha2-256, rsa-sha2-512, ssh-rsa , ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and ecdsa-sha2-nistp521 key algorithms were present, consistent with the ST.

2.2.13 FCS_TLSC_EXT.1 TLS Client Protocol

2.2.13.1 TSS Activities

FCS_TLSC_EXT.1.1

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified include those listed for this component.

Section 6.2 of [ST] (“Cryptographic Support”) lists the supported TLS cipher suites for the TOE’s TLS client implementation, which is consistent with those that are claimed in the SFR.

FCS_TLSC_EXT.1.2

The evaluator shall ensure that the TSS describes the client’s method of establishing all reference identifiers from the administrator/application-configured reference identifier, including which types of reference identifiers are supported (e.g. application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported.

© 2024 Leidos. All rights reserved.

Section 6.2 of [ST] states the TOE verifies that the presented identifier matches the reference identifier according to RFC 6125. The TOE compares the external server's presented identifier to the reference identifier by matching the certificate FQDN (hostname) of the server certificate. The SAN is checked first and if there is any match, the connection is allowed. The TOE supports wildcards (for FQDN only) for peer authentication.

FCS_TLSC_EXT.1.2

Note that where a TLS channel is being used between components of a distributed TOE for FPT_ITT.1, the requirements to have the reference identifier established by the administrator are relaxed and the identifier may also be established through a "Gatekeeper" discovery process. The TSS shall describe the discovery process and highlight how the reference identifier is supplied to the "joining" component. Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1 and the ST author selected attributes from RFC 5280, the evaluator shall ensure the TSS describes which attribute type, or combination of attributes types, are used by the client to match the presented identifier with the configured identifier. The evaluator shall ensure the TSS presents an argument how the attribute type, or combination of attribute types, uniquely identify the remote TOE component; and the evaluator shall verify the attribute type, or combination of attribute types, is sufficient to support unique identification of the maximum supported number of TOE components.

The TOE is not distributed, so this evaluation activity is not applicable.

FCS_TLSC_EXT.1.2

If IP addresses are supported in the CN as reference identifiers, the evaluator shall ensure that the TSS describes the TOE's conversion of the text representation of the IP address in the CN to a binary representation of the IP address in network byte order. The evaluator shall also ensure that the TSS describes whether canonical format (RFC 5952 for IPv6, RFC 3986 for IPv4) is enforced.

The TOE does not support IP addresses in the CN as reference identifiers, so this activity is not applicable.

FCS_TLSC_EXT.1.3

None

FCS_TLSC_EXT.1.4

If "present the Supported Groups Extension" is selected, the evaluator shall verify that TSS describes the Supported Groups Extension and whether the required behaviour is performed by default or may be configured. If TLS 1.2 is claimed and DHE ciphers are claimed, then the TSS must also specify whether the TOE is capable of negotiating DHE ciphers and whether the TOE client will terminate if an unsupported DHE parameter set is returned in the Server Key Exchange or whether all valid server-generated DHE parameters are accepted.

Section 6.2 of [ST] states the TOE supports secp256r1, secp384r1, and secp521r1 by default.

FCS_TLSC_EXT.1.5

[Conditional]: The evaluator shall verify that TSS describes the signature_algorithms extension and whether the required behavior is performed by default or may be configured.

[Conditional]: The evaluator shall verify that TSS describes the signature_algorithms_cert extension and whether the required behavior is performed by default or may be configured.

© 2024 Leidos. All rights reserved.

Section 6.2 of [ST] states the TOE supports the signature_algorithms extension and that no configuration is required, other than to enable FIPS-CC mode.

FCS_TLSC_EXT.1.6
The evaluator shall verify that TSS describes whether the list of supported ciphersuites can be configured or not.

Section 6.2 of [ST] states when FIPS-CC mode is enabled, only the TLS ciphersuites specified in the requirement are supported. The TOE does not provide the ability to configure the list of supported ciphersuites in the evaluated configuration.

FCS_TLSC_EXT.1.7
None

FCS_TLSC_EXT.1.8
The evaluator shall verify in the TSS that, for TLS 1.3, the TOE shall not permit out-of-band provisioning of pre-shared keys (PSKs) in the evaluated configuration.

The ST specifies in FCS_TLSC_EXT.1.8 the TOE shall not use PSKs. As such, the TOE does not permit out-of-band provisioning of PSKs for TLS 1.3.

FCS_TLSC_EXT.1.9
None

2.2.13.2 Guidance Activities

FCS_TLSC_EXT.1.1
The evaluator shall check the guidance documentation to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS.

Section 6.2 of [CCECG] (“Enable FIPS-CC Mode (Required)”) describes how to enable FIPS-CC mode and states that this is sufficient to ensure that the supported TLS versions and cipher suites are limited to those claimed in [ST].

FCS_TLSC_EXT.1.2
The evaluator shall ensure that the operational guidance describes all supported identifiers, explicitly states whether the TOE supports the SAN extension or not and includes detailed instructions on how to configure the reference identifier(s) used to check the identity of peer(s). If the identifier scheme implemented by the TOE includes support for IP addresses, the evaluator shall ensure that the operational guidance provides a set of warnings and/or CA policy recommendations that would result in secure TOE use.

Section 6.2 of [CCECG] indicates the TOE supports reference identifiers according to section 6 of RFC 6125 in the SAN or CN.

© 2024 Leidos. All rights reserved.

Section 6.8.1 (“Syslog Server Connection Settings (Required)”) provides detailed instructions on how to configure the reference identifiers used to check the identity of peers and provides warnings and CA policy recommendations. It states the TOE supports the SAN extension and that it takes priority over the CN.

FCS_TLSC_EXT.1.2
Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1, the SFR selects attributes from RFC 5280, and FCO_CPC_EXT.1.2 selects “no channel”; the evaluator shall verify the guidance provides instructions for establishing unique reference identifiers based on RFC 5280 attributes.

The TOE is not distributed so this evaluation activity is not applicable.

FCS_TLSC_EXT.1.3
None

FCS_TLSC_EXT.1.4
If the TSS indicates that the Supported Groups Extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the Supported Groups Extension.

Section 6.2 of [ST] states the TOE presents the Supported Elliptic Curves/Supported Groups Extension in the Client Hello with the secp256r1, secp384r1, and secp521r1 NIST curves and when FIPS-CC mode is enabled.

Section 6 of [CCECG] (“Evaluated Configuration”) states the administrator must enable FIPS-CC mode as part of the procedure for establishing the evaluated configuration. Section 6.2 of [CCECG] instructs the administrator how to enable FIPS-CC mode.

FCS_TLSC_EXT.1.5
If the TSS indicates that the signature_algorithms extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the signature_algorithms extension.

Section 6.2 of [ST] states the TOE supports the signature_algorithms extension and that no configuration is required, other than to enable FIPS-CC mode. Section 6.2 of [CCECG] instructs the administrator how to enable FIPS-CC mode.

FCS_TLSC_EXT.1.6
If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall verify that AGD guidance includes configuration of the list of supported ciphersuites.

The TSF does not provide the ability to configure the list of supported ciphersuites. Therefore, this activity is not applicable.

FCS_TLSC_EXT.1.7
None

FCS_TLSC_EXT.1.8

© 2024 Leidos. All rights reserved.

The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

Section 6 of [CCECG] states the administrator must enable FIPS-CC mode as part of the procedure for establishing the evaluated configuration. Section 6.2 of [CCECG] instructs the administrator how to enable FIPS-CC mode. The TOE does not use PSKs with TLS and there is no specific configuration to ensure this.

FCS_TLSC_EXT.1.9
None

2.2.13.3 Test Activities

FCS_TLSC_EXT.1.1
Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

The evaluator configured a test TLS server to restrict its supported ciphersuites to one claimed ciphersuite and verified that the TOE established a connection. The evaluator repeated this for each ciphersuite specified in the requirement.

FCS_TLSC_EXT.1.1
The goal of the following test is to verify that the TOE accepts only certificates with appropriate values in the extendedKeyUsage extension, and implicitly that the TOE correctly parses the extendedKeyUsage extension as part of X.509v3 server certificate validation.
Test 2: The evaluator shall establish the connection with a server presenting a certificate that contains the serverAuth (OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage extension and verify that the connection successfully negotiated. The evaluator shall then verify that when the same server presents an otherwise valid server certificate that contains the extendedKeyUsage extension without serverAuth the client rejects the connection. Ideally, the two certificates should be identical except for the OID values.

The evaluator confirmed that a TLS server presenting a certificate with the Server Authentication purpose in the extendedKeyUsage field resulted in a successful connection. The evaluator then configured the TLS server with a certificate without the Server Authentication purpose in the extendedKeyUsage field and attempted a connection and verified that the TOE did not accept the connection.

FCS_TLSC_EXT.1.1
Test 3: [conditional]: Perform this test only if support of TLS 1.2 is claimed. The evaluator shall send a server certificate in the TLS connection that does not match the server-selected ciphersuite (for example, send an ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite). The evaluator shall verify that the TOE disconnects after receiving the server's Certificate handshake message.

© 2024 Leidos. All rights reserved.

The evaluator configured the TLS server to present a server certificate that did not match the server-selected ciphersuite and attempted a connection from the TOE. The evaluator confirmed that the TOE did not accept the connection.

FCS_TLSC_EXT.1.1

Test 4: The evaluator shall perform the following 'negative tests':

i. [conditional]: Perform this test only if support of TLS 1.2 is claimed. The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the TOE TLS client denies the connection.

The evaluator configured the TLS server to present the TLS_NULL_WITH_NULL_NULL cipher suite in the Server Hello message and attempted a connection from the TOE. The evaluator confirmed that the TOE did not accept the connection.

ii. Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite (compatible with the server-selected version of TLS) not presented in the Client Hello handshake message. The evaluator shall verify that the TOE TLS client rejects the connection after receiving the Server Hello.

The evaluator configured a remote TLS server to present a Server Hello message that contained a cipher suite that did not match any presented in the Client Hello handshake. The evaluator confirmed that the TOE did not accept the connection.

iii. The evaluator shall attempt to establish a TLS connection using each valid TLS/SSL version (i.e. TLS 1.3, TLS 1.2, TLS 1.1, TLS 1.0, SSL 3.0, SSL 2.0). The evaluator shall verify that the version(s) specified in FCS_TLSC_EXT.1.1 are successfully established and all other versions are rejected by the TOE TLS client. If a supported_versions extension is not sent by the TOE in the ClientHello, then the evaluator must ensure the test server responds with a ServerHello that is valid for the TLS version being negotiated. If the TOE includes the Supported Versions extension in its ClientHello, the evaluator shall also ensure the version(s) specified in the extension match the version(s) in FCS_TLSC_EXT.1.1. NOTE: For TLS 1.3 aware test servers, it is appropriate for the test server to issue a TLS Alert. The TOE client must not attempt to continue the connection.

The evaluator configured a remote TLS server to negotiate one valid version of TLS or SSL and observed the TOE responded appropriately. For TLS 1.1, TLS 1.0, SSL 3.0, and SSL 2.0, the TOE rejected the connection. Execution of Test 1 for FCS_TLSC_EXT.1.1 showed the TOE successfully establishing TLS 1.2 and TLS 1.3 connections.

FCS_TLSC_EXT.1.1

Test 5: The evaluator shall perform the following modifications to the traffic (i.e. Man-in-the-middle modifications that result in invalid signatures and MACs):

i. [conditional]: Perform this test only if support of TLS 1.2 is claimed. If using DHE or ECDH ciphersuites, modify the signature block in the Server's Key Exchange handshake message, and verify that the client denies the connection and no application data flows. The handshake shall be valid (e.g. the Finished message is calculated using the modified signature), with the exception of the invalid signature. This test does not apply to ciphersuites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.

© 2024 Leidos. All rights reserved.

The evaluator modified the TLS server to send a Server Key Exchange message with a modified signature block. The evaluator confirmed that the TOE did not accept the connection.

ii. [conditional]: Perform this test only if support of TLS 1.3 is claimed. Modify the signature block in the Server's Certificate Verify handshake message, and verify that the client denies the connection and no application data flows. The handshake shall be valid (e.g. the Finished message is calculated using the modified signature), with the exception of the invalid signature.

The evaluator configured a TLS server to modify a byte within the TLS 1.3 Certificate Verify signature. The evaluator confirmed that the TOE did not accept the connection.

FCS_TLSC_EXT.1.1

Test 6: The evaluator shall perform the following 'scrambled message tests':

i. Modify a byte in the Server Finished handshake message and verify that the handshake does not finish successfully and no application data flows. (Note: This modification must be performed prior to the contents of the Finished message being encrypted.)

The evaluator modified a byte in the Server Finished record before encrypting and sending it to the client and observed the TOE terminating the connection after receiving the Server Finished message.

ii. [conditional]: Perform this test only if support of TLS 1.2 is claimed. Send a garbled message from the server after the server has issued the ChangeCipherSpec message and verify that the handshake is not finished successfully and no application data flows. (Note: TLS 1.3 provides for a dummy ChangeCipherSpec message to aid in middlebox compatibility if such an option is enabled in the specific implementation [see Section D.4 in RFC 8446]. If TLS 1.3 middlebox compatibility mode is enabled a ChangeCipherSpec message may appear in packet traces, but it does not influence the protocol. To be clear: for TLS 1.3, this test does not need to be performed.)

The evaluator configured a TLS server to send a garbled application data message instead of a Finished record after the ChangeCipherSpec message and confirmed the TOE rejected the connection.

iii. [conditional]: Perform this test only if support of TLS 1.3 is claimed. Send a plaintext EncryptedExtensions message from the server and verify that the handshake is not finished successfully and no application data flows. (Note: Under TLS 1.3, the EncryptedExtensions message is the first message to be encrypted with the handshake traffic secret.)

The evaluator configured a TLS server to modify the nonce sent in the Server Hello handshake message and confirmed the TOE terminated the connection after receiving the Server Key Exchange message.

iv. [conditional]: Perform this test only if support of TLS 1.2 is claimed. Modify at least one byte in the server's nonce in the Server Hello handshake message and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE ciphersuite) or that the server denies the client's Finished handshake message.

The evaluator configured a TLS server to modify the nonce sent in the Server Hello handshake message and confirmed the TOE terminated the connection after receiving the Server Key Exchange message.

FCS_TLSC_EXT.1.2

Note that the following tests are marked conditional and are applicable under the following conditions:

© 2024 Leidos. All rights reserved.

a. For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.

or

b. For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable

or

c. For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.

Note that for some tests additional conditions apply.

IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:

- IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.
- IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.

The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection:

Test 1 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the CN.

Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1.

The evaluator configured the TLS server to present a certificate with a CN that does match the reference identifier and no SAN extension. The evaluator attempted a connection from the TOE and verified that the TOE did not accept the connection. For this test, the evaluator generated a certificate with an invalid FQDN.

FCS_TLSC_EXT.1.2

Test 2 [conditional]: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, URI). When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the SAN.

The evaluator configured the TLS server to present a certificate with a CN that does match the reference identifier and a SAN extension with a value that does not match the reference identifier. The evaluator attempted a connection from the TOE and verified that the TOE did not accept the connection. The evaluator tested a bad FQDN in the SAN extension.

FCS_TLSC_EXT.1.2

© 2024 Leidos. All rights reserved.

Test 3 [conditional]: If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted.

The evaluator configured the TLS server to present a certificate with a valid CN and no SAN extension. The evaluator attempted a connection from the TOE and verified that the TOE accepted the connection. The evaluator tested FQDN in the CN field.

FCS_TLSC_EXT.1.2

Test 4 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, SRV).

The evaluator configured the TLS server to present a certificate with a CN that does not match the reference identifier and a SAN extension with a value that does match the reference identifier. The evaluator attempted a connection from the TOE and verified that the TOE accepted the connection. The evaluator tested FQDN in the SAN extension.

FCS_TLSC_EXT.1.2

Test 5 [conditional]: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URI-ID):

i. [conditional]: The evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails.

The evaluator configured the TLS server to present a certificate with a DNS value of tlss.*.ate in the CN field. The evaluator attempted a connection from the TOE and verified that the TOE did not accept the connection.

ii. [conditional]: The evaluator shall present a server certificate containing a wildcard in the left-most label (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds, if wildcards are supported, or fails if wildcards are not supported. The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.com) and verify that the connection fails. (Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities.)

The evaluator configured the TLS server to present a certificate with a DNS value of *.leidos.ate and configured the TOE with a reference identifier of tlss.leidos.ate. The evaluator attempted a connection from the TOE and verified that the TOE accepted the connection. The evaluator then configured the TOE to communicate to a server with no left-most label (tlss.ate) and two left-most labels (test.tlss.leidos.ate) independently and attempted a connection for each. The evaluator confirmed that the TOE did not accept either connection.

© 2024 Leidos. All rights reserved.

FCS_TLSC_EXT.1.2

Test 6: Objective: The objective of this test is to ensure the TOE is able to differentiate between IP address identifiers that are not allowed to contain wildcards and other types of identifiers that may contain wildcards.

[conditional]: If IP address identifiers are supported in the SAN or CN, the evaluator shall present a server certificate that contains a CN that matches the reference identifier, except one of the groups has been replaced with a wildcard asterisk (*) (e.g. CN=*.168.0.1 when connecting to 192.168.0.1, CN=2001:0DB8:0000:0000:0008:0800:200C:* when connecting to 2001:0DB8:0000:0000:0008:0800:200C:417A). The certificate shall not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported IP address version (e.g. IPv4, IPv6).

The TOE does not claim to support IP address identifiers in the CN or SAN. Thus, this test is not applicable.

FCS_TLSC_EXT.1.2

Test 7: [conditional]: If the secure channel is used for FPT_ITT, and RFC 5280 is selected, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator shall modify each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier):

i. The evaluator shall present a server certificate that does not contain an identifier in the Subject (DN) attribute type(s) that matches the reference identifier. The evaluator shall verify that the connection fails.

ii. The evaluator shall present a server certificate that contains a valid identifier as an attribute type other than the expected attribute type (e.g. if the TOE is configured to expect id-at-serialNumber=correct_identifier, the certificate could instead include id-at-name=correct_identifier), and does not contain the SAN extension. The evaluator shall verify that the connection fails. Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass this test.

iii. The evaluator shall present a server certificate that contains a Subject attribute type that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.

iv. The evaluator shall confirm that all use of wildcards results in connection failure regardless of whether the wildcards are used in the left or right side of the presented identifier. (Remark: Use of wildcards is not addressed within RFC 5280.)

This test is not applicable because the TOE is not distributed and does not claim FPT_ITT.1.

FCS_TLSC_EXT.1.3

The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows:

Test 1: Using the administrative guidance, the evaluator shall load a CA certificate or certificates needed to validate the presented certificate used to authenticate an external entity and demonstrate that the function succeeds, and a trusted channel can be established.

© 2024 Leidos. All rights reserved.

Initially, the evaluator configured a TLS server to present a certificate that did not chain back to a certificate in the TOE's trust store. The evaluator observed that the connection failed. The evaluator then loaded into the TOE's trust store the root CA certificate and intermediate CA certificates needed to validate the TLS server's certificate and attempted a connection from the TOE to the TLS server. The evaluator confirmed that the connection succeeded.

FCS_TLSC_EXT.1.3

The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows:

Test 2 [conditional]: If "except with the following administrator override" is selected, the evaluator shall change the presented certificate(s) or modify the operational environment, so that certificate validation fails due to the TSF's inability to determine revocation status. The evaluator shall verify that the certificate is not accepted by the TSF until the Security Administrator authorizes the TSF to establish the connection and this action results in the Trusted Channel being successfully established.

The TOE does not claim any administrator override mechanisms, so this test is not applicable.

FCS_TLSC_EXT.1.3

The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows:

Test 3: While performing testing of invalid TLS Client Reference Identifiers, expired X.509 certificates, and invalid X.509 trust chains; the evaluator shall ensure the TSF does not present an administrator override option, with the exception of failure to determine revocation status (if selected). Note: This should be a review of behavior observed while performing other tests.

The TOE does not claim any administrator override mechanisms, so this test is not applicable.

FCS_TLSC_EXT.1.4

Test 1 [conditional]: If "not present the Supported Groups Extension" is selected, the evaluator shall examine the Client Hello message and verify it does not contain the Supported Groups extension.

The TOE claims to present the Supported Groups Extension, so this test is not applicable.

FCS_TLSC_EXT.1.4

Test 2 [conditional]: If "present the Supported Groups Extension" is selected, the evaluator shall configure the server to perform ECDHE or DHE (as applicable) key exchange using each of the TOE's supported groups. The evaluator shall verify that the connection succeeds. This test shall be repeated for each type of key exchange message/extension supported (i.e. Key Share extension for TLS 1.3 and Server Key Exchange Message for TLS 1.2).

The evaluator configured a test TLS server to allow only one supported group and verified that the TOE could establish a connection successfully. The evaluator repeated this test for each claimed key exchange group (secp256r1, secp384r1, secp521r1) and supported version of TLS (TLS v1.2, TLS v1.3).

FCS_TLSC_EXT.1.4

Test 3 [conditional]: If secp curves are selected, the evaluator shall configure the server to perform an ECDHE key exchange in the TLS connection using a non-supported curve and shall verify that the connection fails and no application data flows. The non-supported curve shall be as similar to the selected curve(s) as possible (i.e. a non-selected curve when not all curves are selected or P-224). This

© 2024 Leidos. All rights reserved.

test shall be repeated for each type of key exchange message/extension supported (i.e. Key Share extension for TLS 1.3 and Server Key Exchange Message for TLS 1.2).

The evaluator configured a TLS server to negotiate only the unsupported secp192r1 curve. The evaluator had the TOE attempt to connect to the server and verified that this connection failed. The evaluator performed this test for both TLS v1.2 and TLS v1.3.

FCS_TLSC_EXT.1.4

Test 4a [conditional, for TLS 1.3 only]: If ffdhe curves are selected, the evaluator shall configure the server to perform a DHE key exchange in the TLS connection using a non-supported group and shall verify that the connection fails and no application data flows. The non-supported group shall be as similar to the selected group(s) as possible (i.e. a non-selected group when not all groups are selected or undefined Codepoint 0x0105 (ffdhe8192 + 1)).

The TOE does not select any ffdhe groups, so this test is not applicable.

FCS_TLSC_EXT.1.4

Test 4b [conditional, for TLS 1.2 only]: If ffdhe curves are selected, the evaluator shall configure the server to return DHE parameters in the Server Key Exchange in the TLS connection that do not meet the construction for any claimed ffdhe group. The evaluator shall verify that the connection fails and no application data flows. If the TOE client supports any server-returned DHE parameter set, then this test is not applicable.

The TOE does not select any ffdhe groups, so this test is not applicable.

FCS_TLSC_EXT.1.5

Test 1 [conditional]: The evaluator shall perform the following tests if "present the signature_algorithms extension" is selected:

i. The evaluator shall examine the Client Hello message and verify it contains the signature_algorithms extension and the SignatureSchemes match the SignatureSchemes specified in the requirement.

ii. The evaluator shall establish a TLS connection using each of the SignatureSchemes specified by the requirement and observes the session is successfully completed. The evaluator shall ensure the test server sends a leaf Certificate that has a public key algorithm that is consistent with the SignatureScheme being tested. For TLS 1.2 and if the ciphersuite is DHE or ECDHE, the evaluator shall ensure that the server sends Server Key Exchange messages consistent with the SignatureScheme being tested. For TLS 1.3, the evaluator shall ensure that the server sends Certificate Verify messages consistent with the SignatureScheme being tested.

The evaluator configured a TLS server to force the use of only one signature algorithm. The evaluator verified that the TOE successfully connected using each supported signature algorithm.

FCS_TLSC_EXT.1.5

Test 2 [conditional]: The evaluator shall perform the following tests if "present the signature_algorithms_cert extension" is selected:

i. The evaluator shall examine the Client Hello message and verify it contains the signature_algorithms_cert extension and the SignatureSchemes match the SignatureSchemes specified in the requirement.

© 2024 Leidos. All rights reserved.

ii. The evaluator shall establish a TLS connection using a certificate chain using each of the SignatureSchemes specified by the requirement. The evaluator shall ensure the signatures used in the certificate chain are consistent with the SignatureScheme being tested.

The TOE does not select “present the signature_algorithms_cert extension”, so this test is not applicable.

FCS_TLSC_EXT.1.6

[conditional]: If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall establish a TLS connection using one of the possible configurations of the list of supported ciphersuites. The evaluator shall then change the configuration and repeat the test. The evaluator shall verify that the behavior of the TOE has changed according to the modification of the list of ciphers. This test shall be repeated for all supported TLS versions. If the TSF does not provide the ability of configuring the list of supported ciphersuites, this test shall be omitted.

The TOE does not claim to provide the ability to configure the list of supported ciphersuites, so this test is not applicable.

FCS_TLSC_EXT.1.7

The evaluator shall establish a TLS connection with a server and observe that the early data extension and the post-handshake client authentication extension according to RFC 8446 Section 4.2 are not advertised in the Client Hello Message. This test shall be executed for all TLS versions supported by the TOE.

The evaluator had the TOE connect to a compliant TLS server while capturing packets. The evaluator observed the packet trace and verified that the TOE did not send the early_data or post_handshake_auth extensions in the Client Hello. The evaluator performed this test for TLS v1.2 and TLS v1.3.

FCS_TLSC_EXT.1.8

None

FCS_TLSC_EXT.1.9

Test 1 [conditional]: If "support TLS 1.2 secure renegotiation…" is selected, the evaluator shall use a network packet analyzer/sniffer to capture a TLS 1.2 handshake between the two TLS endpoints. The evaluator shall verify that either the “renegotiation_info” field or the SCSV ciphersuite is included in the ClientHello message during the initial handshake.

The TOE does not select “support TLS 1.2 secure renegotiation”, so this test is not applicable.

FCS_TLSC_EXT.1.9

Test 2 [conditional]: If "support TLS 1.2 secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and verify the TOE TLS Client’s handling of ServerHello messages received during the initial handshake that include the “renegotiation_info” extension. The evaluator shall modify the length portion of this field in the ServerHello message to be non-zero and verify that the TOE TLS client sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field results in a successful TLS connection.

The TOE does not select “support TLS 1.2 secure renegotiation”, so this test is not applicable.

© 2024 Leidos. All rights reserved.

FCS_TLSC_EXT.1.9

Test 3 [conditional]: If "support TLS 1.2 secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and verify that ServerHello messages received during secure renegotiation contain the "renegotiation_info" extension. The evaluator shall modify either the "client_verify_data" or "server_verify_data" value and verify that the TOE TLS client terminates the connection.

The TOE does not select "support TLS 1.2 secure renegotiation", so this test is not applicable.

FCS_TLSC_EXT.1.9

Test 4 [conditional]: If "reject…renegotiation attemps" is selected, then for each selected TLS version, the evaluator shall initiate a TLS session between the so-configured TSF and a test server that is configured to perform a compliant handshake, followed by a hello reset request. The evaluator shall confirm that the TSF completes the initial handshake successfully but terminates the TLS session after receiving the hello reset request. Note: It is preferred that the TSF sends a fatal error alert message (e.g., unexpected message) in response to this, but it is acceptable that the TSF terminates the connection silently (i.e., without sending a fatal error alert).

The evaluator configured a TLS server to establish a handshake and attempt to renegotiate. The evaluator observed that the TOE refused to renegotiate the connection and remained on the existing negotiated parameters and session. The evaluator performed this test for both TLS 1.2 and TLS 1.3 connections.

2.2.14 FCS_TLSS_EXT.1 TLS Server Protocol without mutual authentication

2.2.14.1 TSS Activities

FCS_TLSS_EXT.1.1

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component.

The evaluator shall verify that the TSS contains a description of how the TOE technically prevents the use of unsupported and undefined SSL and TLS versions.

Section 6.2 of [ST] ("Cryptographic Support") lists the supported TLS cipher suites for the TOE's TLS server implementation, consistent with the claims made in FCS_TLSS_EXT.1. It also states the TOE denies connections from clients requesting connections using SSL 2.0, SSL 3.0, TLS 1.0, or TLS 1.1 by default (when FIPS-CC mode is enabled).

FCS_TLSS_EXT.1.2

The evaluator shall verify that the TSS describes the algorithms and key sizes the TSF supports for authenticating itself to TLS clients. The evaluator shall ensure these algorithms are consistent with the selected ciphersuites.

Section 6.2 of [ST] states the TOE authenticates itself to TLS clients using a server certificate with RSA key sizes 2048 bits or greater and ECDSA key sizes 256 bits or greater.

© 2024 Leidos. All rights reserved.

FCS_TLSS_EXT.1.3

The evaluator shall verify that the TSS lists all EC Diffie-Hellman curves and/or Diffie-Hellman groups used in the key establishment by the TOE when acting as a TLS Server. The evaluator shall ensure these algorithms are consistent with the selected ciphersuites.

Section 6.2 of [ST] states the key agreement parameters of the server key exchange message consist of the key establishment parameters generated by the TOE: Diffie-Hellman Ephemeral parameters with key size 2048 bits; and ECDHE implementing NIST curves secp256r1, secp384r1, and secp521r1. These algorithms are consistent with the selected ciphersuites.

FCS_TLSS_EXT.1.4

The evaluator shall verify that the TSS describes if session resumption based on session IDs is supported (RFC 4346 and/or RFC 5246), if session resumption based on session tickets is supported (RFC 5077) and/or if session resumption according to RFC 8446 is supported.

If session tickets are supported, the evaluator shall verify that the TSS describes that the session tickets are encrypted using symmetric algorithms consistent with FCS_COP.1/DataEncryption. The evaluator shall verify that the TSS identifies the key lengths and algorithms used to protect session tickets.

If session tickets are supported, the evaluator shall verify that the TSS describes that session tickets adhere to the structural format provided in Section 4 of RFC 5077 and if not, a justification shall be given of the actual session ticket format.

Section 6.2 of [ST] states the TOE supports session resumption based on session tickets (RFC 5077) for TLS 1.2. Session tickets are encrypted using 128-bit AES encryption and a 256-bit HMAC-SHA-256 key, and the session tickets themselves adhere to the structural format specified by RFC 5077. This is consistent with the algorithms identified in FCS_COP.1/DataEncryption.

FCS_TLSS_EXT.1.4

If the TOE claims a TLS server capable of session resumption (as a single context, or across multiple contexts), the evaluator shall verify that the TSS describes how session resumption operates (i.e. what would trigger a full handshake, e.g. checking session status, checking Session ID, etc.). If multiple contexts are used, the TSS describes how session resumption is coordinated across those contexts. In case session establishment and session resumption are always using a separate context, the TSS shall describe how the contexts interact with respect to session resumption (in particular regarding the session ID). It is acceptable for sessions established in one context to be resumable in another context.

Section 6.2 of [ST] states the TOE supports session resumption using tickets for a single context (i.e., multiple contexts are not used). A full handshake is triggered if the session ticket expires.

FCS_TLSS_EXT.1.5

The evaluator shall verify that TSS describes whether the list of supported ciphersuites can be configured or not.

Section 6.2 of [ST] states when FIPS-CC mode is enabled, only the TLS ciphersuites specified in the requirement are supported.

FCS_TLSS_EXT.1.6

None

© 2024 Leidos. All rights reserved.

FCS_TLSS_EXT.1.7
The evaluator shall verify in the TSS that, for TLS 1.3, the TOE shall not permit out-of-band provisioning of pre-shared keys (PSKs) in the evaluated configuration.

The ST specifies in FCS_TLSS_EXT.1.7 the TOE shall not use PSKs. As such, the TOE does not permit out-of-band provisioning of PSKs for TLS 1.3.

FCS_TLSS_EXT.1.8
None

2.2.14.2 Guidance Activities

FCS_TLSS_EXT.1.1
The evaluator shall check the guidance documentation to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE or TLS version supported by the TOE may have to be restricted to meet the requirements).

Section 6.2 of [CCECG] ("Enable FIPS-CC Mode (Required)") describes how to enable FIPS-CC mode and states that this is sufficient to ensure that the TLS version and TLS cipher suites are limited to those claimed in [ST].

FCS_TLSS_EXT.1.2
The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

Section 6.2 of [CCECG] describes how to enable FIPS-CC mode and states TLS ciphersuites are negotiated based on the public key algorithm (RSA or ECDSA) in the TLS certificate.

FCS_TLSS_EXT.1.3
The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

Section 6.2 of [CCECG] states the administrator must enable FIPS-CC mode in order to restrict the TLS cipher suites to the values claimed in [ST] and provides the necessary instructions.

Section 7.2 of [CCECG] ("Configure Custom HTTPS or TLS Server Certificate") notes the TOE automatically derives the key establishment parameters specified in FCS_TLSS_EXT.1.3 from the negotiated TLS cipher suite.

FCS_TLSS_EXT.1.4
The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

Section 6.2 of [ST] states the TOE's support of session resumption does not require any configuration.

FCS_TLSS_EXT.1.5

© 2024 Leidos. All rights reserved.

If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall verify that AGD guidance includes configuration of the list of supported ciphersuites.

Section 6.2 of [CCECG] describes how the administrator configures the list of supported ciphersuites by enabling FIPS-CC mode, which restricts the ciphersuites to those listed in the requirement and ensures the TOE accepts only TLS v1.2 and TLS v1.3 connection requests.

FCS_TLSS_EXT.1.6
None

FCS_TLSS_EXT.1.7
The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

Section 6 of [CCECG] ("Evaluated Configuration") states the administrator must enable FIPS-CC mode as part of the procedure for establishing the evaluated configuration. Section 6.2 of [CCECG] instructs the administrator how to enable FIPS-CC mode. The TOE does not use PSKs and there is no specific configuration to ensure this.

FCS_TLSS_EXT.1.8
None

2.2.14.3 Test Activities

FCS_TLSS_EXT.1.1
Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

The evaluator configured a TLS client to request each of the ciphersuites claimed in [ST] and attempted a connection to the TOE. The evaluator confirmed that negotiation of each ciphersuite was successful.

FCS_TLSS_EXT.1.1
Test 2: The evaluator shall perform the following tests:

i. The evaluator shall send a Client Hello to the server with a list of ciphersuites that does not contain any of the ciphersuites in the server's ST and verify that the server denies the connection.
ii. [conditional]: Perform this test only if support of TLS 1.2 is claimed. The evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the server denies the connection.

The evaluator configured a TLS client to request a list of cipher suites not supported by the TOE (based on the list of claimed cipher suites in the ST) and attempted a connection to the TOE. The evaluator confirmed that the TOE did not accept the connection.

© 2024 Leidos. All rights reserved.

Next, the evaluator configured a TLS client to request the TLS_NULL_WITH_NULL_NULL cipher suite and attempted a connection to the TOE. The evaluator confirmed that the TOE did not accept the connection.

FCS_TLSS_EXT.1.1

Test 3: The evaluator shall perform the following modifications to the traffic:

i. [conditional]: Perform this test only if support of TLS 1.2 is claimed. Modify a byte in the Client Finished handshake message, and verify that the server rejects the connection and does not send any application data.

(The intent of this test is to ensure that the server's TLS implementation immediately makes use of the key exchange and authentication algorithms to: a) Correctly encrypt TLS Finished message and b) Encrypt every TLS message after session keys are negotiated.)

The evaluator configured a TLS client to modify the last byte in the Client Finished handshake message and sent the modified Client Finished message to the TOE. The evaluator observed the TOE terminated the connection and did not send any application data.

ii. [conditional]: Perform this test only if support of TLS 1.2 is claimed. The evaluator shall use one of the claimed ciphersuites to complete a successful handshake and observe transmission of properly encrypted application data. The evaluator shall verify that no Alert with alert level Fatal (2) messages were sent.

The evaluator shall verify that the Finished message (Content type hexadecimal 16 and handshake message type hexadecimal 14) is sent immediately after the server's ChangeCipherSpec (Content type hexadecimal 14) message. The evaluator shall examine the Finished message (encrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 11 22 33 44 55…) and confirm that it does not contain unencrypted data (unencrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 14 00 00 0c…), by verifying that the first byte of the encrypted Finished message does not equal hexadecimal 14 for at least one of three test messages. There is a chance that an encrypted Finished message contains a hexadecimal value of '14' at the position where a plaintext Finished message would contain the message type code '14'. If the observed Finished message contains a hexadecimal value of '14' at the position where the plaintext Finished message would contain the message type code, the test shall be repeated three times in total. In case the value of '14' can be observed in all three tests it can be assumed that the Finished message has indeed been sent in plaintext and the test has to be regarded as 'failed'. Otherwise it has to be assumed that the observation of the value '14' has been due to chance and that the Finished message has indeed been sent encrypted. In that latter case the test shall be regarded as 'passed'.

The evaluator used a TLS client to carry out a compliant handshake with the TOE and send application data while capturing packets and verified that the Server's Encrypted Handshake Message was truly encrypted and no Alert messages were sent. The evaluator examined the frame number 6 and found that the data bytes for Encrypted Handshake Message did not contain "16 03 03 00 40 14 00 00 0c" but were truly encrypted.

iii. [conditional]: Perform this test only if support of TLS 1.3 is claimed. The evaluator shall use a client to send a Client Hello message containing a single curve in the Supported Groups extension. The curve that is selected to be presented in this extension should not be supported

© 2024 Leidos. All rights reserved.

by the TOE. The evaluator shall verify that the TOE disconnects after receiving the Client Hello message.

The evaluator configured a TLS client to send a Client Hello containing only an unsupported curve. The evaluator verified that the TOE rejected the handshake.

iv. [conditional]: Perform this test only if support of TLS 1.3 is claimed. The evaluator shall use a client to send a Client Hello message containing multiple curves in the Supported Groups extension. These curves should be chosen such that only one of these curves is supported by the TOE. The evaluator shall verify that the TOE responds with a Hello Retry Request message selecting the supported curve. This shall be reflected in the Key Share extension of the Hello Retry Request message.

The evaluator configured a TLS client to send a list of multiple curves to the TOE, only one of which was supported by the TOE. The evaluator verified that the TOE responded with a Hello Retry Request selecting the supported curve in the key_share extension.

FCS_TLSS_EXT.1.1

Test 4: The evaluator shall attempt to establish a TLS/SSL connection using each of the supported TLS/SSL versions (i.e., TLS 1.3, TLS 1.2, TLS 1.1, TLS 1.0, SSL 3.0, SSL 2.0). The client shall be configured so it only supports the version being tested. The evaluator shall verify that the versions specified in FCS_TLSS_EXT.1.1 are successfully established and all other versions not successfully established. If the TOE attempts to downgrade the version, it is acceptable for the test client to terminate the connection; however, the version selected by the TOE shall always be a version specified in FCS_TLSS_EXT.1.1.

The evaluator configured a TLS client to attempt a connection with versions of TLS not supported by the TOE (i.e., SSL 2.0, SSL 3.0, TLS 1.0, and TLS 1.1). The evaluator observed that the TOE rejected all connection attempts. The ability of the TOE to accept and successfully complete connection attempts using TLS v1.2 and TLS v1.3 was demonstrated in FCS_TLSS_EXT.1.1 Test 1. In that test, the evaluator iterated through all TOE-supported ciphersuites across TLS 1.2 and TLS 1.3 and verified that the TOE accepted connections with these versions.

FCS_TLSS_EXT.1.2 and FCS_TLSS_EXT.1.3

Test 1 [conditional]: If ECDHE ciphersuites/group are supported:

The evaluator shall repeat this test for each supported elliptic curve. The evaluator shall attempt a connection using a supported ECDHE ciphersuite (TLS 1.2) or group (TLS 1.3) and a single supported elliptic curve specified in the supported groups extension. The Evaluator shall verify (through a packet capture or instrumented client) that the TOE selects the same curve in the Server Key Exchange (TLS 1.2) or Server Hello (key_share, for TLS 1.3) message and successfully establishes the connection.

For TLS 1.2, the evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single unsupported elliptic curve (e.g., secp192r1 (0x13)) specified in RFC 4492, Section 5.1.1. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.

For TLS 1.3, the evaluator shall attempt a connection using a supported ciphersuite and a single unsupported group. Both the key_share and supported_groups extensions must be set to the same unsupported group. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.

© 2024 Leidos. All rights reserved.

The evaluator configured a TLS client to support the given version of TLS with a supported ECDHE ciphersuite/group and verified that each supported curve could be used to establish a connection.

The evaluator configured the TLS client to send a supported TLS 1.2 ECDHE ciphersuite but only an unsupported elliptic curve and verified that the TOE rejected the connection.

Evidence for TLS 1.3 with an unsupported group can be found in the evidence for FCS_TLSS_EXT.1.1 test 3c. In that test, the evaluator configured a TLS client to send only a group that was not supported by the TOE and verified that the TOE rejected the connection.

FCS_TLSS_EXT.1.2 and FCS_TLSS_EXT.1.3

Test 2 [conditional]: If DHE ciphersuites are supported, the evaluator shall repeat the following test for each supported parameter size. If any configuration is necessary, the evaluator shall configure the TOE to use a supported Diffie-Hellman parameter size. The evaluator shall attempt a connection using a supported DHE ciphersuite.

For TLS 1.2, the evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Exchange Message where p Length is consistent with the message are the ones configured Diffie-Hellman parameter size(s).

For TLS 1.3, the evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Share Extension Message where the KeyShareServerHello structure contains a KeyShareEntry structure with an opaque key_exchange value whose Length is consistent with the configured Diffie-Hellman parameter size(s).

The evaluator configured a test TLS client to connect to the TOE utilizing a DHE ciphersuite while capturing packets. The evaluator inspected the packet capture and found that the p-length was consistent with the selection (256 bytes, or 2048 bits).

FCS_TLSS_EXT.1.2 and FCS_TLSS_EXT.1.3

Test 3 [conditional]: If RSA key establishment ciphersuites are supported, the evaluator shall repeat this test for each RSA key establishment key size. If any configuration is necessary, the evaluator shall configure the TOE to perform RSA key establishment using a supported key size (e.g. by loading a certificate with the appropriate key size). The evaluator shall attempt a connection using a supported RSA key establishment ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a certificate whose modulus is consistent with the configured RSA key size.

The TOE does not support RSA key establishment cipher suites, so this test is not applicable.

FCS_TLSS_EXT.1.4

Test Objective: To demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption).

Test 1 [conditional]: If the TOE does not support session resumption based on session IDs according to RFC 5246 (TLS 1.2) or session tickets according to RFC 5077 (TLS 1.2) or session resumption according to RFC 8446 (TLS 1.3), the evaluator shall perform the following test:

i. For all supported TLS versions the client shall send a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket. A non-zero length session identifier for TLS 1.3 would result in testing compatibility mode which is not the objective of this test. For TLS 1.3, the evaluator shall ensure that a 'psk_key_exchange_modes' extension is included in the Client Hello.

© 2024 Leidos. All rights reserved.

ii. The client verifies the server does not send a NewSessionTicket handshake message (at any point in the handshake).

iii. The client verifies the Server Hello message contains a zerolength session identifier. For TLS 1.2 the client could alternatively pass the following steps (not applicable for TLS 1.3):

Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID.

iv. The client completes the TLS handshake and captures the SessionID from the ServerHello.

v. The client sends a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d).

vi. The client verifies the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

Remark: If multiple contexts are supported for session resumption, the session ID or session ticket may be obtained in one context for resumption in another context. It is possible that one or more contexts may only permit the construction of sessions to be reused in other contexts but not actually permit resumption themselves. For contexts which do not permit resumption, the evaluator is required to verify this behaviour subject to the description provided in the TSS. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

The TOE supports session resumption based on session tickets according to RFC 5077 and session resumption according to RFC 8446, so this test is not applicable.

Test 2 [conditional]: If the TOE supports session resumption using session IDs according to RFC 5246 (TLS 1.2), the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

i. The evaluator shall conduct a successful handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then initiate a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages (as shown in Figure 2 of RFC 4346 or RFC 5246). When the session is resumed, the evaluator shall verify on the TLS Client used for performing this test, that the TOE (TLS Server) has not advertised support for the early data extension.

ii. The evaluator shall initiate a handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then, within the same handshake, generate or force an unencrypted fatal Alert message immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake. The evaluator shall then initiate a new Client Hello using the previously captured session ID, and verify that the server (1) implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

Remark: If multiple contexts are supported for session resumption, for each of the above test cases, the session ID may be obtained in one context for resumption in another context. There is no requirement that the session ID be obtained and replayed within the same context subject to the description provided in the TSS. All contexts that can reuse a session ID constructed in another context must be

© 2024 Leidos. All rights reserved.

tested. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

The TOE does not support session resumption based on session IDs, so this test is not applicable.

Test 3 [conditional]: If the TOE supports session tickets according to RFC 5077 (supported only by TLS 1.2), the evaluator shall carry out the following steps:

i. The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the session ticket in the ClientHello. The evaluator shall confirm that the TOE responds with an abbreviated handshake described in Section 3.1 of RFC 5077 and illustrated with an example in figure 2. Of particular note: if the server successfully verifies the client's ticket, then it may renew the ticket by including a NewSessionTicket handshake message after the ServerHello in the abbreviated handshake (which is shown in figure 2). This is not required, however as further clarified in Section 3.3 of RFC 5077. When the session is resumed, the evaluator shall verify on the TLS Client used for performing this test, that the TOE (TLS Server) has not advertised support for the early data extension.

ii. The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then modify the session ticket and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data.

Remark: If multiple contexts are supported for session resumption, for each of the above test cases, the session ticket may be obtained in one context for resumption in another context. There is no requirement that the session ticket be obtained and replayed within the same context subject to the description provided in the TSS. All contexts that can reuse a session ticket constructed in another context must be tested. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

The evaluator configured a TLS client to attempt session resumption via session ticket. The evaluator verified that the TOE accepted the ticket and resumed the session.

The evaluator configured a TLS client to establish a connection and attempt to resume with a modified session ticket. The evaluator verified that the TOE rejected the ticket and initiated a new TLS session handshake.

Test 4 [conditional]: If the TOE supports session resumption according to RFC 8446 (supported only by TLS 1.3), the evaluator shall carry out the following steps:

i. The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the pre-shared key in the ClientHello. The evaluator shall confirm that the TOE responds similarly to figure 3 of RFC 8446 after successfully reusing the pre-shared-key to resume the session. Specifically, the server must not send back a Certificate message if the session is correctly resumed. When the session is resumed, the evaluator shall verify on the

© 2024 Leidos. All rights reserved.

TLS Client used for performing this test, that the TOE (TLS Server) has not advertised support for the early data extension.

The evaluator configured a TLS client to complete a TLS handshake and then attempt to correctly reuse the session. The evaluator observed that the session was resumed. This evaluator determined the TOE did not send a Certificate message when correctly resuming the session. The evaluator additionally observed that the server did not advertise the early data extension.

ii. The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then modify the pre-shared key and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake, or (2) terminates the connection in some way that prevents the flow of application data.

The evaluator configured a TLS client to complete a TLS handshake and attempt to resume the session but modify the pre-shared key first. The evaluator observed that the TOE rejected the session ticket and terminated the connection.

iii. The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then force the non-TOE client to attempt to establish a new connection using the previous session ticket material as a pre-shared key, but set psk_key_exchange_modes with a value of psk_ke in the Client Hello message and omit the psk_ke_dhe. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake, or (2) terminates the connection in some way that prevents the flow of application data.

The evaluator configured a TLS client to complete a TLS handshake and attempt to resume the session with the correct PSK, but omit the psk_ke_dhe value within psk_key_exchange_modes. The evaluator observed that the TOE terminated the session when it received the ticket.

FCS_TLSS_EXT.1.5

Test 1 [conditional]: If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall establish a TLS connection using one of the possible configurations of the list of supported ciphersuites. The evaluator shall then change the configuration and repeat the test. The evaluator shall verify that the behavior of the TOE has changed according to the modification of the list of ciphers. This test shall be repeated for all supported TLS versions. If the TSF does not provide the ability of configuring the list of supported ciphersuites, this test shall be omitted.

The evaluator observed the configuration of the TOE's ciphersuites. The evaluator established a connection to the TOE using a supported ciphersuite with the AES128-GCM cipher. The evaluator configured the list of supported ciphersuites to exclude those using AES128-GCM. The evaluator attempted the connection again and observed that the connection now failed.

FCS_TLSS_EXT.1.6

According to RFC 8446 Section 4.2.10, a PSK is required to use the early data extension. As NDcPP only allows the use of PSK in conjunction with session resumption, a NDcPP conformant TOE which acts as TLS Server cannot use the early data extension if session resumption is not supported. For TOEs that do not support session resumption, execution of test FCS_TLSS_EXT.1.4 Test 1 is regarded as sufficient that the TOE does not support the early data extension. For TOEs that support session resumption,

© 2024 Leidos. All rights reserved.

FCS_TLSS_EXT.1.4 Test 2(i), 3(i) or 4(i) (depending on the supported TLS versions and the way session resumption is implemented) ensure that the TOE does not support the early data extension.

Per the Test Activity, execution of FCS_TLSS_EXT.3(i) and 4(i) satisfy this requirement.

FCS_TLSS_EXT.1.7
None

FCS_TLSS_EXT.1.8
Test 1 [conditional]: If "support secure renegotiation…" is selected, the evaluator shall use a network packet analyzer/sniffer to capture a TLS 1.2 handshake between the two TLS endpoints. The evaluator shall verify that the “renegotiation_info” extension is included in the ServerHello message.

The ST does not select “support secure renegotiation…”, so this test is not applicable.

Test 2 [conditional]: If "support secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and modify the length portion of the field in the ClientHello message in the initial handshake to be non-zero. The evaluator shall verify that the TOE TLS server sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field results in a successful TLS connection.

The ST does not select “support secure renegotiation…”, so this test is not applicable.

Test 3 [conditional]: If "support secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and modify the "client_verify_data" or "server_verify_data" value in the ClientHello message received during secure renegotiation. The evaluator shall verify that the TOE TLS server terminates the connection.

The ST does not select “support secure renegotiation…”, so this test is not applicable.

Test 4 [conditional]: If "reject…renegotiation attemps" is selected, then for each selected TLS version, the evaluator shall follow the operational guidance as necessary to configure the TSF to negotiate the version and reject renegotiation. The evaluator shall initiate a valid initial session for the specified version, send a valid ClientHello on the non-renegotiable TLS channel, and observe that the TSF terminates the session. Note: It is preferred that the TSF sends a fatal error alert message (e.g., unexpected message) in response to this, but it is acceptable that the TSF terminates the connection silently (i.e., without sending a fatal error alert).

The evaluator configured a TLS client to establish a valid TLS session and then initiated a renegotiation. After receiving the renegotiation attempt, the TOE sent an alert indicating that it does not support renegotiation and terminated the connection. This was done for TLS 1.2 and TLS 1.3.

2.2.15 FCS_IPSEC_EXT.1 IPsec Protocol (VPNGW-SD)

This SFR is modified by MOD_VPNGW_v1.3 and includes the following additional evaluation activities.

TSS

© 2024 Leidos. All rights reserved.

All existing activities regarding "Pre-shared keys" apply to all selections including pre-shared keys. If any selection with "Pre-shared keys" is included, the evaluator shall check to ensure that the TSS describes how the selection works in conjunction with the authentication of IPsec connections.

Guidance

If any selection with “Pre-shared Keys” is selected, the evaluator shall check that the operational guidance describes any configuration necessary to enable any selected authentication mechanisms.

Test

There are no additional testing activities.

Section 6.2 of [ST] (“Cryptographic Support”) states the TOE supports PPK (Post-Quantum Pre-shared Key) as specified in RFC 8784 for authentication with the regular IKEv2 key exchange. The TOE will generate PPK keys with sizes of 128 bits (default) and 256 bits upon request. The bit-based keys are generated from the TOE’s approved DRBG in FIPS-CC mode, and keys must be securely transported via an out of band mechanism.

Section 7.10 of [CCECG] (“Configure IKE/IPsec VPN Gateway”) provides instructions for configuring Post-Quantum Pre-shared Key (PPK).

2.2.15.1 TSS Activities

FCS_IPSEC_EXT.1.1

The evaluator shall examine the TSS and determine that it describes what takes place when a packet is processed by the TOE, e.g., the algorithm used to process the packet. The TSS describes how the SPD is implemented and the rules for processing both inbound and outbound packets in terms of the IPsec policy. The TSS describes the rules that are available and the resulting actions available after matching a rule. The TSS describes how those rules and actions form the SPD in terms of the BYPASS (e.g., no encryption), DISCARD (e.g., drop the packet), and PROTECT (e.g., encrypt the packet) actions defined in RFC 4301.

As noted in Section 4.4.1 of RFC 4301, the processing of entries in the SPD is non-trivial and the evaluator shall determine that the description in the TSS is sufficient to determine which rules will be applied given the rule structure implemented by the TOE. For example, if the TOE allows specification of ranges, conditional rules, etc., the evaluator shall determine that the description of rule processing (for both inbound and outbound packets) is sufficient to determine the action that will be applied, especially in the case where two different rules may apply. This description shall cover both the initial packets (that is, no SA is established on the interface or for that particular packet) as well as packets that are part of an established SA.

Section 6.2 of [ST] describes the TOE’s SPD implementation. This section describes how the administrator can configure access control lists to process traffic using PROTECT, BYPASS, and DISCARD rules. This section also addresses potential conflicting rules by saying that the rules are processed in a strict order such that the first matching rule is processed. If no rules are found to match the traffic, it is denied by default.

FCS_IPSEC_EXT.1.2

None.

FCS_IPSEC_EXT.1.3

© 2024 Leidos. All rights reserved.

The evaluator shall check the TSS to ensure it states that the VPN can be established to operate in transport mode and/or tunnel mode (as identified in FCS_IPSEC_EXT.1.3).

Section 6.2 of [ST] states the TOE implements tunnel mode, which is consistent with the selection made in FCS_IPSEC_EXT.1.3.

FCS_IPSEC_EXT.1.4

The evaluator shall examine the TSS to verify that the selected algorithms are implemented. In addition, the evaluator shall ensure that the SHA-based HMAC algorithm conforms to the algorithms specified in FCS_COP.1/KeyedHash Cryptographic Operations (for keyed-hash message authentication) and if the SHA-based HMAC function truncated output is utilized it must also be described.

Section 6.2 of [ST] identifies the supported IPsec algorithms as AES-CBC (128, 192, and 256 bits) and AES-GCM (128 and 256 bits), consistent with FCS_IPSEC_EXT.1.4. This section also states that the HMAC algorithms claimed in FCS_IPSEC_EXT.1.4 (HMAC-SHA-256/384/512) are supported, all of which are claimed in FCS_COP.1/KeyedHash. This section also states that truncated output for the HMAC function is not used.

FCS_IPSEC_EXT.1.5

The evaluator shall examine the TSS to verify that IKEv1 and/or IKEv2 are implemented.

For IKEv1 implementations, the evaluator shall examine the TSS to ensure that, in the description of the IPsec protocol, it states that aggressive mode is not used for IKEv1 Phase 1 exchanges, and that only main mode is used. It may be that this is a configurable option.

Section 6.2 of [ST] indicates that IKEv2 (as defined in RFCs 5996 and 4868) protocols are implemented by the TOE. The TOE does not implement IKEv1 and therefore the second activity is not applicable.

FCS_IPSEC_EXT.1.6

The evaluator shall ensure the TSS identifies the algorithms used for encrypting the IKEv1 and/or IKEv2 payload, and that the algorithms chosen in the selection of the requirement are included in the TSS discussion.

Section 6.2 of [ST] states the TOE supports AES-CBC-128, AES-CBC-192, AES-CBC-256, AES-GCM-128, and AES-GCM-256 for IKEv2, consistent with the selections made in FCS_IPSEC_EXT.1.6.

FCS_IPSEC_EXT.1.7

The evaluator shall ensure the TSS identifies the lifetime configuration method used for limiting the IKEv1 Phase 1 SA lifetime and/or the IKEv2 SA lifetime. The evaluator shall verify that the selection made here corresponds to the selection in FCS_IPSEC_EXT.1.~~5~~7.

Section 6.2 of [ST] states the IKEv2 SA lifetimes are configurable from 1-8760 hours, and that byte ranges are also supported. As the SFR requires a maximum lifetime of 24 hours for SAs and 8 hours for child SAs, the specified configurable range supports both of these. The evaluator verified that the description in the TSS corresponds to the selection made in the SFR. The TOE does not support IKEv1.

FCS_IPSEC_EXT.1.8

The evaluator shall ensure the TSS identifies the lifetime configuration method used for limiting the IKEv1 Phase 2 SA lifetime and/or the IKEv2 Child SA lifetime. The evaluator shall verify that the selection made here corresponds to the selection in FCS_IPSEC_EXT.1.~~5~~8.

© 2024 Leidos. All rights reserved.

Section 6.2 of [ST] specifies length of time and number of bytes as configurable options for Phase 2/Child SA lifetimes. The evaluator verified that the description in the TSS corresponds to the selections made in the SFR.

FCS_IPSEC_EXT.1.9

The evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating "x". The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of "x" meets the stipulations in the requirement.

Section 6.2 of [ST] states that “x” in g^x mod p is generated using the TOE’s DRBG (as claimed in FCS_RBG_EXT.1). This section also identifies the length of x based on the supported DH groups.

FCS_IPSEC_EXT.1.10

If the first selection is chosen, the evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of the nonces meet the stipulations in the requirement.

If the second selection is chosen, the evaluator shall check to ensure that, for each PRF hash supported, the TSS describes the process for generating each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of the nonces meet the stipulations in the requirement.

Section 6.2 of [ST] states nonces are generated using the TOE’s Approved DRBG.

[ST] chooses the second selection. Section 6.2 states the nonce size is at least 128 bits and is at least half the output length of the PRF hash and indicates that the TOE generates 256 bits for each DH group.

FCS_IPSEC_EXT.1.11

The evaluator shall check to ensure that the DH groups specified in the requirement are listed as being supported in the TSS. If there is more than one DH group supported, the evaluator shall check to ensure the TSS describes how a particular DH group is specified/negotiated with a peer.

Section 6.2 of [ST] states the TOE supports DH groups 14, 15, 16, 19, 20, and 21, consistent with the selections made. This section also states the TSF will choose the first group that matches based on the peer configuration, and that the connection will fail if no matching groups are presented.

FCS_IPSEC_EXT.1.12

The evaluator shall check that the TSS describes the potential strengths (in terms of the number of bits in the symmetric key) of the algorithms that are allowed for the IKE and ESP exchanges. The TSS shall also describe the checks that are done when negotiating IKEv1 Phase 2 and/or IKEv2 CHILD_SA suites to ensure that the strength (in terms of the number of bits of key in the symmetric algorithm) of the negotiated algorithm is less than or equal to that of the IKE SA this is protecting the negotiation.

Section 6.2 of [ST] states the administrator is instructed to ensure that the size of the key used for ESP is less than or equal to the key size used to protect the IKE payload. Per earlier elements in this SFR, the TOE supports 128, 192, and 256 bits for both ESP and IKE.

FCS_IPSEC_EXT.1.13

© 2024 Leidos. All rights reserved.

The evaluator shall ensure that the TSS identifies RSA and/or ECDSA as being used to perform peer authentication. The description must be consistent with the algorithms as specified in FCS_COP.1/SigGen Cryptographic Operations (for cryptographic signature).

If pre-shared keys are chosen in the selection, the evaluator shall check to ensure that the TSS describes how pre-shared keys are established and used in authentication of IPsec connections. The description in the TSS shall also indicate how pre-shared key establishment is accomplished for TOEs that can generate a pre-shared key as well as TOEs that simply use a pre-shared key.

Section 6.2 of [ST] states the TOE uses RSA peer X.509v3 certificate authentication. This is consistent with the selections in FCS_COP.1.1/SigGen. The ST selects pre-shared keys in the selection in FCS_IPSEC_EXT.1.13 and the description states the TOE implements support for PPK (Post-Quantum Pre-shared Key) as specified in RFC 8784 for authentication with the regular IKEv2 key exchange. The TOE will generate PPK keys with sizes of 128 bits (default) and 256 bits upon request. The bit-based keys are generated from the TOE's approved DRBG in FIPS-CC mode, and keys must be securely transported via an out of band mechanism. Keys that are externally generated can be accepted and used by the TOE but the quality of those keys are out of scope.

FCS_IPSEC_EXT.1.14

The evaluator shall ensure that the TSS describes how the TOE compares the peer's presented identifier to the reference identifier. This description shall include which field(s) of the certificate are used as the presented identifier (DN, Common Name, or SAN). If the TOE simultaneously supports the same identifier type in the CN and SAN, the TSS shall describe how the TOE prioritizes the comparisons (e.g. the result of comparison if CN matches but SAN does not). If the location (e.g. CN or SAN) of non-DN identifier types must explicitly be configured as part of the reference identifier, the TSS shall state this. If the ST author assigned an additional identifier type, the TSS description shall also include a description of that type and the method by which that type is compared to the peer's presented certificate, including what field(s) are compared and which fields take precedence in the comparison.

Section 6.2 of [ST] states the TOE will only establish an IKE channel if the presented identifier in the peer X.509v3 certificate CN matches the configured identifier: Distinguished Name (DN); IP address; or Fully Qualified Domain Name (FQDN).

2.2.15.2 Guidance Activities

FCS_IPSEC_EXT.1.1

The evaluator shall examine the guidance documentation to verify it instructs the Administrator how to construct entries into the SPD that specify a rule for processing a packet. The description includes all three cases – a rule that ensures packets are encrypted/decrypted, dropped, and flow through the TOE without being encrypted. The evaluator shall determine that the description in the guidance documentation is consistent with the description in the TSS, and that the level of detail in the guidance documentation is sufficient to allow the administrator to set up the SPD in an unambiguous fashion. This includes a discussion of how ordering of rules impacts the processing of an IP packet.

Section 7.10.1 of [CCECG] ("Policy-Based Forwarding") describes how to configure Policy Based Forwarding (PBF) rules consistent with the definition of an IPsec Security Policy Database (SPD) as specified in RFC 4301 (i.e., rules that contain operations that DISCARD, BYPASS, and PROTECT network packets). The guidance gives procedures:

© 2024 Leidos. All rights reserved.

1) To configure the TOE to forward packets matching the security policy to a VPN peer without going through the IPsec tunnel (BYPASS)

2) To configure the TOE to forward packets matching the security policy to a VPN peer via the IPsec tunnel (PROTECT)

3) To configure the TOE to deny packets matching the security policy (DISCARD).

FCS_IPSEC_EXT.1.2
None.

FCS_IPSEC_EXT.1.3
The evaluator shall confirm that the guidance documentation contains instructions on how to configure the connection in each mode selected.

Section 7.10.1 of [CCECG] includes instructions for configuring tunnel mode for each configured connection.

FCS_IPSEC_EXT.1.4
The evaluator shall check the guidance documentation to ensure it provides instructions on how to configure the TOE to use the algorithms selected.

Section 7.10 of [CCECG] ("Configure IKE/IPsec VPN Gateway") describes how to configure the TOE to use the algorithms identified in [ST] and states that only the following encryption and authentication algorithms can be used for the IPsec protocol ESP: aes-cbc-128; aes-cbc-192; aes-cbc-256; aes-gcm-128; aes-gcm-256; hmac-sha1; hmac-sha256; hmac-sha384; and hmac-sha512.

FCS_IPSEC_EXT.1.5
The evaluator shall check the guidance documentation to ensure it instructs the administrator how to configure the TOE to use IKEv1 and/or IKEv2 (as selected), and how to configure the TOE to perform NAT traversal (if selected).

If the IKEv1 Phase 1 mode requires configuration of the TOE prior to its operation, the evaluator shall check the guidance documentation to ensure that instructions for this configuration are contained within that guidance.

Section 7.10 of [CCECG] describes how to configure the TOE to use IKEv2 and NAT traversal. The [CCECG] states that only IKEv2 should be used.

FCS_IPSEC_EXT.1.6
The evaluator shall ensure that the guidance documentation describes the configuration of all selected algorithms in the requirement.

Section 7.10 of [CCECG] describes how to configure an IKE cryptographic profile and states to use only aes-cbc-128, aes-cbc-192, aes-cbc-256, aes-gcm-128, or aes-gcm-256.

FCS_IPSEC_EXT.1.7
The evaluator shall verify that the values for SA lifetimes can be configured and that the instructions for doing so are located in the guidance documentation. If time-based limits are supported, configuring the limit may lead to a rekey no later than the specified limit. For some implementations, it may be

© 2024 Leidos. All rights reserved.

necessary, though, to configure the TOE with a lower time value to ensure a rekey is performed before the maximum SA lifetime of 24 hours is exceeded (e.g. configure a time value of 23h 45min to ensure the actual rekey is performed no later than 24h). The evaluator shall verify that the guidance documentation allows the Administrator to configure the Phase 1 SA value of 24 hours or provides sufficient instruction about the time value to configure to ensure the rekey is performed no later than the maximum SA lifetime of 24 hours. It is not permitted to configure a value of 24 hours if that leads to an actual rekey after more than 24hours. Currently there are no values mandated for the number of bytes, the evaluator just ensures that this can be configured if selected in the requirement.

Section 7.10 of [CCECG] shows the Phase 1 lifetime configured in the IKE profile and states that lifetime can be specified in seconds, minutes, hours, or days. The supported range is 3 minutes to 365 days, with a default of 8 hours, which allows the administrator to configure the Phase 1 SA value of 24 hours.

[CCECG] does not identify any configuration options for number of bytes in phase 1 SA lifetimes and this is consistent with 'number of bytes' not being selected in FCS_IPSEC_EXT.1.7.

FCS_IPSEC_EXT.1.8

The evaluator shall verify that the values for SA lifetimes can be configured and that the instructions for doing so are located in the guidance documentation. If time-based limits are supported, configuring the limit may lead to a rekey no later than the specified limit. For some implementations, it may be necessary, though, to configure the TOE with a lower time value to ensure a rekey is performed before the maximum SA lifetime of 8 hours is exceeded (e.g. configure a time value of 7h 45min to ensure the actual rekey is performed no later than 8h). The evaluator shall verify that the guidance documentation allows the Administrator to configure the Phase 2 SA value of 8 hours or provides sufficient instruction about the time value to configure to ensure the rekey is performed no later than the maximum SA lifetime of 8 hours. It is not permitted to configure a value of 8 hours if that leads to an actual rekey after more than 8 hours. Currently there are no values mandated for the number of bytes, the evaluator just ensures that this can be configured if selected in the requirement.

Section 7.10 of [CCECG] shows the Phase 2 lifetime configured in the IPsec Profile and states that lifetime can be specified in terms both of time (in seconds, minutes, hours, or days—the supported range is 3 minutes to 365 days), and volume of data. Configurable values for lifetime in terms of time include the required 8 hours.

FCS_IPSEC_EXT.1.9 and FCS_IPSEC_EXT.1.10

None.

FCS_IPSEC_EXT.1.11

The evaluator shall ensure that the guidance documentation describes the configuration of all algorithms selected in the requirement.

Section 7.10 of [CCECG] states when configuring an IPsec cryptographic profile to use only the following Diffie-Hellman (DH) groups: group14; group15; group16; group19; group20; and group 21. These are the same algorithms selected in the requirement. The administrator configures the algorithms in the IKE Profile under DH Group.

FCS_IPSEC_EXT.1.12

None.

© 2024 Leidos. All rights reserved.

FCS_IPSEC_EXT.1.13

The evaluator shall ensure the guidance documentation describes how to set up the TOE to use certificates with RSA and/or ECDSA signatures and public keys.

The evaluator shall check that the guidance documentation describes how pre-shared keys are to be generated and established. The description in the guidance documentation shall also indicate how pre-shared key establishment is accomplished for TOEs that can generate a pre-shared key as well as TOEs that simply use a pre-shared key.

The evaluator shall ensure that the guidance documentation describes how to configure the TOE to connect to a trusted CA and ensure a valid certificate for that CA is loaded into the TOE and marked "trusted".

Section 7.10 of [CCECG] provides instructions to configure the TOE to use certificates with RSA or ECDSA signatures and public keys, and Pre-shared Keys that conform to RFC 8784. The procedures indicate that Certificate Profiles are used to instruct the TOE to connect to a trusted CA. A trusted channel is established only if the presented identifier in the peer certificate matches the configured reference identifier and the peer certificate is signed by a trusted anchor CA specified in the Certificate Profile. Local identification defines the format and identification of the local gateway. The Local Certificate identifies the local gateway certificate (RSA-based or ECDSA-based) that will be presented to the IKE peer.

FCS_IPSEC_EXT.1.14

The evaluator shall ensure that the operational guidance describes all supported identifiers, explicitly states whether the TOE supports the SAN extension or not and includes detailed instructions on how to configure the reference identifier(s) used to check the identity of peer(s). If the identifier scheme implemented by the TOE does not guarantee unique identifiers, the evaluator shall ensure that the operational guidance provides a set of warnings and/or CA policy recommendations that would result in secure TOE use.

Section 7.10 of [CCECG] provides the instructions for configuring IKE/IPsec VPN Gateways. This section states the TOE establishes a trusted channel only if the presented identifier in the peer certificate matches the configured reference identifier. The guidance describes all supported identifiers (Distinguished Name (Subject), FQDN (hostname), IP address, and provides detailed instructions how to configure reference identifiers.

2.2.15.3 Test Activities

FCS_IPSEC_EXT.1.1

The evaluator shall use the guidance documentation to configure the TOE to perform the following tests:

Test 1: The evaluator shall configure the SPD such that there is a rule for dropping a packet, encrypting a packet, and allowing a packet to flow in plaintext. The selectors used in the construction of the rule shall be different such that the evaluator can generate a packet and send packets to the gateway with the appropriate fields (fields that are used by the rule - e.g., the IP addresses, TCP/UDP ports) in the packet header. The evaluator shall perform both positive and negative test cases for each type of rule (e.g. a packet that matches the rule and another that does not match the rule). The evaluator shall observe via the audit trail, and packet captures that the TOE exhibited the expected behaviour: appropriate packets were dropped, allowed to flow without modification, encrypted by the IPsec implementation.

© 2024 Leidos. All rights reserved.

The evaluator configured SPD rules to allow and deny traffic based on the following attributes: destination IP address; source IP address; and protocol. The evaluator configured a rule for encrypting a packet, allowing a packet to flow in plaintext, and denying a packet. The evaluator sent packets through the TOE to each of the destination addresses found in the forwarding rules and verified via packet capture that the TOE correctly implemented the packet forwarding rules.

FCS_IPSEC_EXT.1.1

Test 2: The evaluator shall devise several tests that cover a variety of scenarios for packet processing. As with Test 1, the evaluator shall ensure both positive and negative test cases are constructed. These scenarios must exercise the range of possibilities for SPD entries and processing modes as outlined in the TSS and guidance documentation. Potential areas to cover include rules with overlapping ranges and conflicting entries, inbound and outbound packets, and packets that establish SAs as well as packets that belong to established SAs. The evaluator shall verify, via the audit trail and packet captures, for each scenario that the expected behavior is exhibited, and is consistent with both the TSS and the guidance documentation.

The evaluator created SPD rules, one to bypass traffic, one to discard traffic and one to encrypt traffic each of which possessed difference conditions either a specific source IP address and protocol, a specific destination IP address and protocol or a specific destination subnet which overlaps the specific destination IP address. The evaluator observed that the first rule that matched the traffic was applied to the traffic and the TOE performed the configured action upon the traffic for the rule.

FCS_IPSEC_EXT.1.2

The assurance activity for this element is performed in conjunction with the activities for FCS_IPSEC_EXT.1.1.

The evaluator uses the guidance documentation to configure the TOE to perform the following tests:

Test 1: The evaluator shall configure the SPD such that there is a rule for dropping a packet, encrypting a packet, and allowing a packet to flow in plaintext. The evaluator may use the SPD that was created for verification of FCS_IPSEC_EXT.1.1. The evaluator shall construct a network packet that matches the rule to allow the packet to flow in plaintext and send that packet. The evaluator shall observe that the network packet is passed to the proper destination interface with no modification. The evaluator shall then modify a field in the packet header; such that it no longer matches the evaluator-created entries (there may be a “TOE created” final entry that discards packets that do not match any previous entries). The evaluator shall send the packet and observe that the packet was dropped.

Using the SPD configuration from FCS_IPSEC_EXT.1.1, the evaluator sent a packet that did not match any configured rule and observed that the packet did not pass through the TOE. This behavior allows the evaluator to conclude that the final entry exists to drop traffic that does not match any rule.

FCS_IPSEC_EXT.1.3

The evaluator shall perform the following test(s) based on the selections chosen:

Test 1: If tunnel mode is selected, the evaluator shall use the guidance documentation to configure the TOE to operate in tunnel mode and also configures a VPN peer to operate in tunnel mode. The evaluator shall configure the TOE and the VPN peer to use any of the allowable cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator shall then initiate a connection from the TOE to connect to the VPN peer. The evaluator shall observe (for example, in the audit trail and the captured packets) that a successful connection was established using the tunnel mode.

© 2024 Leidos. All rights reserved.

The ST selects only “tunnel mode” in FCS_IPSEC_EXT.1.3. All IPsec connections established by the TOE are in tunnel mode and no specific configuration for the connection mode is required. This test was covered by the testing for FCS_IPSEC_EXT.1.1.

FCS_IPSEC_EXT.1.3

Test 2: If transport mode is selected, the evaluator shall use the guidance documentation to configure the TOE to operate in transport mode and also configures a VPN peer to operate in transport mode. The evaluator shall configure the TOE and the VPN peer to use any of the allowed cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator shall then initiate a connection from the TOE to connect to the VPN peer. The evaluator shall observe (for example, in the audit trail and the captured packets) that a successful connection was established using the transport mode.

This test is not applicable because transport mode is not selected by the ST.

FCS_IPSEC_EXT.1.4

The evaluator shall configure the TOE as indicated in the guidance documentation configuring the TOE to use each of the supported algorithms, attempt to establish a connection using ESP, and verify that the attempt succeeds.

The evaluator configured the TOE to use each of the supported encryption algorithms (AES-CBC-128, AES-CBC-192, AES-CBC-256, AES-GCM-128, and AES-GCM-256) and each of the supported hash algorithms (HMAC-SHA256, HMAC-SHA384, HMAC-SHA512). The evaluator established separate IPsec connections using each algorithm for ESP between the TOE and an IPsec peer. The evaluator confirmed that each connection attempt was successful, and a Security Association was created using each algorithm.

FCS_IPSEC_EXT.1.5

Tests are performed in conjunction with the other IPsec evaluation activities.

Test 1: If IKEv1 is selected, the evaluator shall configure the TOE as indicated in the guidance documentation and attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator shall then show that main mode exchanges are supported.

This test is not applicable as the TOE does not select IKEv1.

FCS_IPSEC_EXT.1.5

Test 2: If NAT traversal is selected within the IKEv2 selection, the evaluator shall configure the TOE so that it will perform NAT traversal processing as described in the TSS and RFC 7296, Section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed.

The evaluator configured the TOE to perform NAT traversal and configured the IPsec peer to accept NAT traversal. The evaluator attempted a connection and confirmed that NAT traversal was performed as specified by the RFC.

FCS_IPSEC_EXT.1.6

The evaluator shall configure the TOE to use the ciphersuite under test to encrypt the IKEv1 and/or IKEv2 payload and establish a connection with a peer device, which is configured to only accept the payload encrypted using the indicated ciphersuite. The evaluator shall confirm the algorithm was that used in the negotiation.

© 2024 Leidos. All rights reserved.

The evaluator configured the TOE to use each of the encryption algorithms (AES128-CBC, AES192-CBC, AES256-CBC, AES128-GCM, AES256-GCM) to establish IKEv2 sessions with an IPsec peer.

FCS_IPSEC_EXT.1.7

When testing this functionality, the evaluator shall ensure that both sides are configured appropriately. From the RFC "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered."

Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection:

Test 1: If 'number of bytes' is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime in terms of the number of bytes allowed following the guidance documentation. The evaluator shall configure a test peer with a byte lifetime that exceeds the lifetime of the TOE. The evaluator shall establish a SA between the TOE and the test peer, and determine that once the allowed number of bytes through this SA is exceeded, a new SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 1 negotiation.

This test is not applicable because 'number of bytes' is not selected in FCS_IPSEC_EXT.1.7.

FCS_IPSEC_EXT.1.7

Test 2: If 'length of time' is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime no later than 24 hours for the Phase 1 SA following the guidance documentation. The evaluator shall configure a test peer with a Phase 1 SA lifetime that exceeds the Phase 1 SA lifetime on the TOE. The evaluator shall establish a SA between the TOE and the test peer, maintain the Phase 1 SA for 24 hours, and determine that a new Phase 1 SA is negotiated on or before 24 hours has elapsed. The evaluator shall verify that the TOE initiates a Phase 1 negotiation.

The evaluator configured the IKEv2 SA lifetimes to be a specified time value. The evaluator then established a connection between the TOE and an IPsec peer which uses IKEv2. The evaluator verified via a packet capture that the TOE initiated renegotiation the connection once the configured amount of time had elapsed.

FCS_IPSEC_EXT.1.8

When testing this functionality, the evaluator shall ensure that both sides are configured appropriately. From the RFC "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered."

Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection:

Test 1: If 'number of bytes' is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime in terms of the number of bytes allowed following the guidance documentation. The evaluator shall configure a test peer with a byte lifetime that exceeds the lifetime of the TOE. The

© 2024 Leidos. All rights reserved.

evaluator shall establish a SA between the TOE and the test peer, and determine that once the allowed number of bytes through this SA is exceeded, a new SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 2 negotiation.

The evaluator configured the TOE to rekey the Child SA negotiation after a specified amount of data had traversed the channel. The evaluator caused the TOE to connect to an IPsec peer and caused data to quickly traverse the channel. The evaluator observed that the TOE rekeyed the Phase 2 negotiation at or before the configured threshold for IKEv2.

FCS_IPSEC_EXT.1.8

Test 2: If 'length of time' is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime no later than 8 hours for the Phase 2 SA following the guidance documentation. The evaluator shall configure a test peer with a Phase 2 SA lifetime that exceeds the Phase 2 SA lifetime on the TOE. The evaluator shall establish a SA between the TOE and the test peer, maintain the Phase 1 SA for 8 hours, and determine that once a new Phase 2 SA is negotiated when or before 8 hours has lapsed. The evaluator shall verify that the TOE initiates a Phase 2 negotiation.

The evaluator configured the IKEv2 Child SA lifetimes to be a specified time value. The evaluator then established a connection between the TOE and an IPsec peer, using IKEv2. The evaluator verified via a packet capture that the TOE initiated renegotiation for IKEv2 connections at or before the configured amount of time had elapsed.

FCS_IPSEC_EXT.1.10

Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection:

Test 1: If the first selection is chosen, the evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of the nonces meet the stipulations in the requirement.

This test is not applicable because the first selection is not chosen in FCS_IPSEC_EXT.1.10.

FCS_IPSEC_EXT.1.10

Test 2: If the second selection is chosen, the evaluator shall check to ensure that, for each PRF hash supported, the TSS describes the process for generating each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of the nonces meet the stipulations in the requirement.

[ST] section 6.2 states "The nonces used in the IKE exchanges are generated at least 128 bits in size and at least half the output size of the PRF hash. The TSF generates the nonces of length 256 (for DH Groups 14, 15, and 16) and 256 (for DH Groups 19, 20, and 21) bits, and are generated with the Approved DRBG".

FCS_IPSEC_EXT.1.11

For each supported DH group, the evaluator shall test to ensure that all supported IKE protocols can be successfully completed using that particular DH group.

The evaluator configured the TOE to use each of the supported DH groups (14, 15, 16, 19, 20, and 21) separately and established connections with an IPsec peer using IKEv2. The evaluator confirmed each attempt was successful and the configured algorithm was used on the connection.

© 2024 Leidos. All rights reserved.

FCS_IPSEC_EXT.1.12

The evaluator shall follow the guidance to configure the TOE to perform the following tests.

Test 1: This test shall be performed for each version of IKE supported. The evaluator shall successfully negotiate an IPsec connection using each of the supported algorithms and hash functions identified in the requirements.

The evaluator configured the TOE for IKEv2 to use each of the supported algorithms claimed in the requirements in turn and established an IPsec connection between the TOE and an IPsec peer. The evaluator confirmed each attempt was successful.

FCS_IPSEC_EXT.1.12

Test 2: This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish a SA for ESP that selects an encryption algorithm with more strength than that being used for the IKE SA (i.e., symmetric algorithm with a key size larger than that being used for the IKE SA). Such attempts should fail.

The evaluator configured the TOE for IKEv2 to use AES-CBC-128 for IKE and AES-CBC-128 and AES-CBC-256 for ESP. The evaluator then attempted to establish an IKEv2 connection from an IPsec peer using AES-CBC-128 for IKE but AES-CBC-256 for ESP. The evaluator confirmed that the connection was denied.

FCS_IPSEC_EXT.1.12

Test 3: This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an IKE SA using an algorithm that is not one of the supported algorithms and hash functions identified in the requirements. Such an attempt should fail.

The evaluator attempted to establish an IKE SA for IKEv2 using a non-supported encryption algorithm. The evaluator verified that the TOE did not establish an IKE session.

FCS_IPSEC_EXT.1.12

Test 4: This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish a SA for ESP (assumes the proper parameters where used to establish the IKE SA) that selects an encryption algorithm that is not identified in FCS_IPSEC_EXT.1.4. Such an attempt should fail.

The evaluator attempted for IKEv2 to establish an SA for ESP that uses a non-supported encryption algorithm. The evaluator verified that the TOE rejected the ESP SA attempt and that the connection failed.

FCS_IPSEC_EXT.1.13

For efficiency sake, the testing is combined with the testing for FIA_X509_EXT.1, FIA_X509_EXT.2 (for IPsec connections), and FCS_IPSEC_EXT.1.1.

Per the evaluation activity, there is no separate testing for this SFR as it is addressed through other testing.

FCS_IPSEC_EXT.1.14

In the context of the tests below, a valid certificate is a certificate that passes FIA_X509_EXT.1 validation checks but does not necessarily contain an authorized subject.

The evaluator shall perform the following tests:

Test 1 [conditional]: For each CN/identifier type combination selected, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the field in the peer's

© 2024 Leidos. All rights reserved.

presented certificate and shall verify that the IKE authentication succeeds. If the TOE prioritizes CN checking over SAN (through explicit configuration of the field when specifying the reference identifier or prioritization rules), the evaluator shall also configure the SAN so it contains an incorrect identifier of the correct type (e.g. the reference identifier on the TOE is example.com, the CN=example.com, and the SAN:FQDN=otherdomain.com) and verify that IKE authentication succeeds.

This test is not applicable as the TOE does not utilize CN only reference identifiers and only uses Distinguished Name (DN) or SAN reference identifiers.

FCS_IPSEC_EXT.1.14

Test 2 [conditional]: For each SAN/identifier type combination selected, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the field in the peer's presented certificate and shall verify that the IKE authentication succeeds. If the TOE prioritizes SAN checking over CN (through explicit specification of the field when specifying the reference identifier or prioritization rules), the evaluator shall also configure the CN so it contains an incorrect identifier formatted to be the same type (e.g. the reference identifier on the TOE is DNS-ID; identify certificate has an identifier in SAN with correct DNS-ID, CN with incorrect DNS-ID (and not a different type of identifier)) and verify that IKE authentication succeeds.

The evaluator configured the TOE to associate a peer certificate with each supported SAN identifier (IP address, FQDN) and verified in all cases that IKE authentication is successful when the certificate associated with the valid identifier is presented.

The evaluator also configured the peer to present a certificate with a valid SAN but invalid CN and verified that the TOE prioritizes the SAN and allows the connection.

FCS_IPSEC_EXT.1.14

Test 3 [conditional]: For each CN/identifier type combination selected, the evaluator shall:

i. Create a valid certificate with the CN so it contains the valid identifier followed by '\0'. If the TOE prioritizes CN checking over SAN (through explicit specification of the field when specifying the reference identifier or prioritization rules) for the same identifier type, the evaluator shall configure the SAN so it matches the reference identifier.

ii. Configure the peer's reference identifier on the TOE (per the administrative guidance) to match the CN without the '\0' and verify that IKE authentication fails.

This test is not applicable as the TOE does not utilize CN only reference identifiers and only uses Distinguished Name (DN) or SAN reference identifiers.

FCS_IPSEC_EXT.1.14

Test 4 [conditional]: For each SAN/identifier type combination selected, the evaluator shall:

i. Create a valid certificate with an incorrect identifier in the SAN. The evaluator shall configure a string representation of the correct identifier in the DN. If the TOE prioritizes CN checking over SAN (through explicit specification of the field when specifying the reference identifier or prioritization rules) for the same identifier type, the addition/modification shall be to any non-CN field of the DN. Otherwise, the addition/modification shall be to the CN.

ii. Configure the peer's reference identifier on the TOE (per the administrative guidance) to match the correct identifier (expected in the SAN) and verify that IKE authentication fails.

© 2024 Leidos. All rights reserved.

The evaluator configured the IPsec peer to present a certificate that contained a SAN that did not match the value configured on the TOE (both for FQDN and IP address) and a valid CN. The evaluator verified that the TOE prioritizes the SAN and rejects the connection because of the SAN value to configured reference identifier mismatch.

FCS_IPSEC_EXT.1.14

Test 5 [conditional]: If the TOE supports DN identifier types, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the subject DN in the peer's presented certificate and shall verify that the IKE authentication succeeds.

The evaluator configured the TOE to utilize a Distinguished Name reference identifier for the IPSEC connection. The evaluator configured the IPsec peer to present a certificate with valid Distinguished Name (DN) information. The evaluator verified that the TOE was able to successfully establish connections when this certificate was presented.

FCS_IPSEC_EXT.1.14

Test 6 [conditional]: If the TOE supports DN identifier types, to demonstrate a bit-wise comparison of the DN, the evaluator shall create the following valid certificates and verify that the IKE authentication fails when each certificate is presented to the TOE:

i. Duplicate the CN field, so the otherwise authorized DN contains two identical CNs.

ii. Append '\0' to a non-CN field of an otherwise authorized DN.

The evaluator configured the IPsec peer to present a certificate that contained the CN field twice with identical values. The evaluator verified that the TOE rejected the connection attempt when this certificate was presented. The evaluator then configured the IPsec peer to present a certificate that contained '\0' appended to the Location field. The evaluator verified that the TOE rejected the connection attempt when this certificate was presented.

2.3 User Data Protection (FDP) (FW-SD)

2.3.1 FDP_RIP.2 Full Residual Information Protection

2.3.1.1 TSS Activities

"Resources" in the context of this requirement are network packets being sent through (as opposed to "to", as is the case when a security administrator connects to the TOE) the TOE. The concern is that once a network packet is sent, the buffer or memory area used by the packet still contains data from that packet, and that if that buffer is re-used, those data might remain and make their way into a new packet. The evaluator shall check to ensure that the TSS describes packet processing to the extent that they can determine that no data will be reused when processing network packets. The evaluator shall ensure that this description at a minimum describes how the previous data are zeroized/overwritten, and at what point in the buffer processing this occurs.

Section 6.3 of [ST] ("User Data Protection") states the TOE allocates and deallocates the memory resources used for network packet objects. When the TOE receives new data from the network and allocates new buffer resources to store and transmit data to the network, it ensures that the new buffers do not contain previously transmitted or otherwise residual information by overwriting unused parts of the buffer with 0s.

© 2024 Leidos. All rights reserved.

2.3.1.2 Guidance Activities

None defined.

2.3.1.3 Test Activities

None defined.

2.4 Identification and Authentication (FIA)

2.4.1 FIA_AFL.1 Authentication Failure Management

2.4.1.1 TSS Activities

The evaluator shall examine the TSS to determine that it contains a description, for each supported method for remote administrative actions, of how successive unsuccessful authentication attempts are detected and tracked. The TSS shall also describe the method by which the remote administrator is prevented from successfully logging on to the TOE, and the actions necessary to restore this ability.

Section 6.4 of [ST] ("Identification and Authentication") states the TSF enforces a lockout mechanism that will trigger if an administrator-configured number between 1 and 10 of consecutive failed attempts is reached. This section also states that this behavior applies to password-based authentication only because public key authentication cannot be brute forced in the same manner. The lock can be configured to last a specified amount of time (1 – 60 minutes) during which providing the correct credentials will still not allow access (i.e., locked out).

The evaluator shall examine the TSS to confirm that the TOE ensures that authentication failures by remote administrators cannot lead to a situation where no administrator access is available, either permanently or temporarily (e.g. by providing local logon which is not subject to blocking).

Section 6.4 of [ST] states the evaluated configuration requires at least one administrator (preferably the predefined 'admin' account) be configured with public key authentication for SSH. In the rare situation where all administrators (customer created) are locked out at the same time, this account can still be used to login.

2.4.1.2 Guidance Activities

The evaluator shall examine the guidance documentation to ensure that instructions for configuring the number of successive unsuccessful authentication attempts and time period (if implemented) are provided, and that the process of allowing the remote administrator to once again successfully log on is described for each "action" specified (if that option is chosen). If different actions or mechanisms are implemented depending on the secure protocol employed (e.g., TLS vs. SSH), all must be described.

Section 7.6 of [CCECG] ("Configure Idle Timeout and Lockout") states the number of failed authentication attempts allowed before the TOE locks a privileged account is configured by setting the number of failed attempts (range is 1 to 10) and the lockout time (in minutes, range is 1 to 60). If an administrator reaches the failed attempts threshold, the TOE locks the administrator out for the configured lockout time.

The guidance provides instructions to configure the TOE via the GUI, CLI, and API.

© 2024 Leidos. All rights reserved.

The evaluator shall examine the guidance documentation to confirm that it describes, and identifies the importance of, any actions that are required in order to ensure that administrator access will always be maintained, even if remote administration is made permanently or temporarily unavailable due to blocking of accounts as a result of FIA_AFL.1.

Section 7.6 of [CCECG] states it is required that an administrator be created or the default admin uses SSH public key-based authentication for additional security and prevention against permanent lockout.

2.4.1.3 Test Activities

The evaluator shall perform the following tests for each method by which remote administrators access the TOE (e.g. any passwords entered as part of establishing the connection protocol or the remote administrator application):

Test 1: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE (and, if the time period selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall also use the operational guidance to configure the time period after which access is re-enabled). The evaluator shall test that once the authentication attempts limit is reached, authentication attempts with valid credentials are no longer successful.

The evaluator configured the TOE's number of successive unsuccessful authentication attempts to 3 and the lockout period to 5 minutes. The evaluator then attempted to authenticate to the TOE with incorrect credentials 3 times and on the 4th time attempted to use the correct credentials. The evaluator confirmed that the TOE did not allow access. This test was performed for both the HTTPS web GUI and SSH CLI.

Test 2: After reaching the limit for unsuccessful authentication attempts as in Test 1 above, the evaluator shall proceed as follows.

If the administrator action selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall confirm by testing that following the operational guidance and performing each action specified in the ST to re-enable the remote administrator's access results in successful access (when using valid credentials for that administrator).

If the time period selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall wait for just less than the time period configured in Test 1 and show that an authorisation attempt using valid credentials does not result in successful access. The evaluator shall then wait until just after the time period configured in Test 1 and show that an authorisation attempt using valid credentials results in successful access.

The ST includes only the time period selection in FIA_AFL.1.2. After the TOE locked the user account in Test 1 above, the evaluator started a timer. The evaluator waited until just before the lockout expired and attempted to log in using correct credentials. The evaluator verified that the attempt failed. The evaluator waited until just after the lockout expired and attempted to login using correct credentials. The evaluator verified that this attempt was successful.

2.4.2 FIA_PMG_EXT.1 Password Management

2.4.2.1 TSS Activities

The evaluator shall check that the TSS:

a. lists the supported special character(s) for the composition of administrator passwords.

© 2024 Leidos. All rights reserved.

b. to ensure that the minimum_password_length parameter is configurable by a Security Administrator.
c. lists the range of values supported for the minimum_password_length parameter. The listed range shall include the value of 15.

Section 6.4 of [ST] ("Identification and Authentication") lists the supported special characters and states the minimum password length can be configured to a value from 8-15 characters, with the maximum password being a fixed 31 characters.

2.4.2.2 Guidance Activities

The evaluator shall examine the guidance documentation to determine that it:
a. identifies the characters that may be used in passwords and provides guidance to security administrators on the composition of strong passwords, and
b. provides instructions on setting the minimum password length and describes the valid minimum password lengths supported

Section 7.7 of [CCECG] ("Configure Minimum Password Length") identifies the characters that can be used in passwords and provides recommendations on how to configure strong passwords.

Section 7.7 of [CCECG] provides the instructions to set the minimum password length. It states the Minimum Length field can be set to a value between 8 and 15 characters.

2.4.2.3 Test Activities

Test 1: The evaluator shall compose passwords that meet the requirements in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, and a minimum length listed in the requirement are supported and justify the subset of those characters chosen for testing.

The evaluator configured the minimum password length on the TOE to be 8 and composed a set of passwords that were at least 8 characters long and together covered all the characters claimed to be supported by the TOE. The TOE accepted each password that met the specified minimum requirements.

Test 2: The evaluator shall compose passwords that do not meet the requirements in some way. For each password, the evaluator shall verify that the TOE does not support the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that the TOE enforces the allowed characters and the minimum length listed in the requirement and justify the subset of those characters chosen for testing.

The evaluator configured the minimum password length on the TOE to be 8 and composed a password that was less than 8 characters. The TOE rejected the password. The evaluator then configured the minimum password length to be 15 characters. The evaluator composed a password that was 14 characters long and verified that the password was not accepted.

© 2024 Leidos. All rights reserved.

2.4.3 FIA_PSK_EXT.1 (VPNGW-SD)

2.4.3.1 TSS Activities

The evaluator shall confirm that the TSS states which pre-shared key selections are supported for IKEv2 per FCS_IPSEC_EXT.1.13 and FPF_MFA_EXT.1.1.

Section 6.2 of [ST] ("Cryptographic Support") states the TOE supports PPK (Post-Quantum Pre-shared Key) as specified in RFC 8784 for authentication with the regular IKEv2 key exchange, consistent with the selection in FCS_IPSEC_EXT.1.13.

2.4.3.2 Guidance Activities

Modified in accordance with TD0838.

The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on how to configure all selected pre-shared key options if any configuration is required.

The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on how to configure the mandatory_or_not flag per RFC 8784.

Section 7.10 of [CCECG] ("Configure IKE/IPsec VPN Gateway") provides guidance to the administrator to configure Post-Quantum Pre-Shared Key (PPK) as specified in RFC 8784. The guidance states this configuration is optional, but if a PPK is configured it must also be configured on the peer. Configuring a PPK on the TOE configures the mandatory_or_not flag.

2.4.3.3 Test Activities

Modified in accordance with TD0838.

The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE).

Test FIA_PSK_EXT.1:1: For each mechanism selected in FIA_PSK_EXT.1.2 the evaluator shall attempt to establish a connection and confirm that the connection requires the selected factors in the PSK to establish the connection in alignment with table 1 from RFC 8784.

The evaluator verified that the TOE can utilize the PSK for an IPsec connection and that the PSK is required to complete the connection.

2.4.4 FIA_PSK_EXT.2 (VPNGW-SD)

2.4.4.1 TSS Activities

If "generate" is selected, the evaluator shall confirm that this process uses the RBG specified in FCS_RBG_EXT.1 and the output matches the size selected in FIA_PSK_EXT.2.1.

Section 6.2 of [ST] ("Cryptographic Support") states the TOE will generate Post-Quantum Pre-shared Keys (PPK) of 128 bits by default and 256 bits upon request. The bit-based keys are generated from the TOE's Approved DRBG. The output matches the sizes selected in FIA_PSK_EXT.2.1.

© 2024 Leidos. All rights reserved.

2.4.4.2 Guidance Activities

The evaluator shall confirm the operational guidance contains instructions for entering generated pre-shared keys for each protocol identified in the FIA_PSK_EXT.1.1.

Section 7.10 of [CCECG] (“Configure IKE/IPsec VPN Gateway”) provides guidance to the administrator for entering generated PPK keys for IKEv2.

2.4.4.3 Test Activities

Test FIA_PSK_EXT.2:1: [conditional] If generate was selected the evaluator shall generate a pre-shared key and confirm the output matches the size selected in FIA_PSK_EXT.2.1.

The evaluator verified that the TOE can successfully generate pre-shared keys with the output sizes specified in FIA_PSK_EXT.2. The evaluator verified that the TOE uses a RBG to generate the output pre-shared keys by making sure different values are returned on multiple calls to the generate function.

2.4.5 FIA_UIA_EXT.1 User Identification and Authentication

2.4.5.1 TSS Activities

The evaluator shall examine the TSS to determine that it describes the logon process for remote authentication mechanism (e.g. SSH public key, Web GUI password, etc.) and optional local authentication mechanisms supported by the TOE. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a “successful logon”.

Section 6.4 of [ST] (“Identification and Authentication”) states the only supported user authentication mechanisms to the TOE are username-password (defined internal to the TOE), SSH public key, and X.509 certificate (HTTPS only). A logon is successful when the username and password provided by the user matches a TOE-defined account, or when the TOE verifies the username and digital signature (i.e., verifies the possession of the private key that corresponds to the public key defined for the account).

The evaluator shall examine the TSS to determine that it describes which actions are allowed before administrator identification and authentication. The description shall cover authentication and identification for local and remote TOE administration.

The TOE supports remote administration only. Section 6.4 of [ST] identifies that the only functionality the TOE will perform without authentication is to display the warning banner or respond to an ICMP request.

For distributed TOEs the evaluator shall examine that the TSS details how Security Administrators are authenticated and identified by all TOE components. If not, all TOE components support authentication of Security Administrators according to FIA_UIA_EXT.1, the TSS shall describe how the overall TOE functionality is split between TOE components including how it is ensured that no unauthorized access to any TOE component can occur.

The TOE is not distributed so this evaluation activity is not applicable.

For distributed TOEs, the evaluator shall examine the TSS to determine that it describes for each TOE component which actions are allowed before administrator identification and authentication. The description shall cover authentication and identification for remote TOE administration and optionally

© 2024 Leidos. All rights reserved.

for local TOE administration if claimed by the ST author. For each TOE component that does not support authentication of Security Administrators according to FIA_UIA_EXT.1 the TSS shall describe any unauthenticated services/services that are supported by the component.

The TOE is not distributed so this evaluation activity is not applicable.

2.4.5.2 Guidance Activities

The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps (e.g., establishing credential material such as pre- shared keys, tunnels, certificates, etc.) to logging in are described. For each supported the login method, the evaluator shall ensure the guidance documentation provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the guidance documentation provides sufficient instruction on limiting the allowed services.

Section 5 of [CCECG] ("Identification and Authentication") states "Before any configuration can be performed on the TOE, the user must login. Other than viewing the login banner and pinging (i.e., ICMP echo request and reply) the TOE, no other action is provided to the users until they are successfully logged in. After that, the actions available will be based on the role and privileges assigned to that user".

Section 5.1 of [CCECG] ("Logging into the TOE") provides the instructions for a user to login to the Web Interface and the CLI.

Section 7.3.2 of [CCECG] ("Adding New Accounts") states "You can set the authentication method (password vs public-key)", and authentication profile (e.g., using authentication server). The guidance provides instructions to "Use only client certificate authentication (Web)" and "Use Public Key Authentication (SSH)". If public key authentication fails, the TOE will automatically failback to password authentication.

Section 6.6 of [CCECG] ("Configure SSH Public-Key Authentication (Recommended)") provides instructions to configure the TOE for remote public key authentication and the commands to enter on the remote workstation to access the TOE.

2.4.5.3 Test Activities

The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method:

Test 1: The evaluator shall use the guidance documentation to configure the appropriate credential supported for the login method. For all combinations of supported credentials and login methods, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.

In the evaluated configuration, the TOE supports only remote access with the following credentials: password at Web GUI and SSH CLI; SSH public key; Web GUI with X.509 certificate. For each credential and remote access mechanism, the evaluator followed guidance to configure credentials and confirmed that when providing valid credentials, access to the TOE was granted, and providing invalid credentials resulted in no access to the TOE. The evaluator tested incorrect and correct password credentials on the Web GUI and SSH CLI as part of testing for FIA_AFL.1.

© 2024 Leidos. All rights reserved.

The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method:

Test 2: The evaluator shall configure the services allowed (if any) according to the guidance documentation, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.

The evaluator confirmed that the TOE would respond to ICMP request messages before authentication. The evaluator also performed a port scan to verify that the only open ports were the ones used for protocols that required authentication. The presentation of the warning banner is confirmed with FTA_TAB.1 testing.

Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.

The TOE does not support local access in the evaluated configuration, so this test is not applicable.

Test 4: For distributed TOEs where not all TOE components support the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2, the evaluator shall test that the components authenticate Security Administrators as described in the TSS.

The TOE is not distributed so this evaluation activity is not applicable.

2.4.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

This SFR is modified to be mandatory when claiming MOD_VPNGW_v1.3, but there are no additional evaluation activities.

2.4.6.1 TSS Activities

The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied). It is expected that revocation checking is performed when a certificate is used in an authentication step and when performing trusted updates (if selected).

The TSS shall describe when revocation checking is performed and on what certificates. If the revocation checking during authentication is handled differently depending on whether a full certificate chain or only a leaf certificate is being presented, any differences must be summarized in the TSS section and explained in the Guidance.

Section 6.4 of [ST] (“Identification and Authentication”) states X.509 certificates are used for TLS, IKE/IPsec, and HTTPS connections. This section also describes the rules for validating certificates (including how certificate path validation of three or more links is enforced). The TOE checks the extendedKeyUsage field for Server Authentication purpose, Client Authentication purpose, and OCSP Signing purpose. It states that certificates are not used for trusted updates or executable code integrity, so the TOE does not support rules for the extendedKeyUsage field that has the Code Signing purpose, and this part of the requirement is trivially satisfied.

Section 6.4 of [ST] states the TOE supports revocation checking of IPsec peer certificates using OCSP and CRL. If both are configured, the TOE first tries the OCSP method; if the OCSP server is unavailable, the TOE uses the CRL method. For TLS, the TOE only supports OCSP.

© 2024 Leidos. All rights reserved.

The TOE downloads and caches OCSP status information for every CA listed in the trusted CA list of the TOE. The OCSP status is cached for the “next update time” that is configured on the OCSP responder. The TOE uses this received value as the cache time. Caching only applies to validated certificates; if the TOE has never validated a certificate, the TOE cache does not store the OCSP information for the issuing CA. The TOE downloads and caches the last-issued CRL for every CA listed in the trusted CA list of the TOE. The signature on the CRL is verified as defined in RFCs 5280 and 5759 (supporting strong Suite B ECDSA algorithms and curve sizes for signing CRL). Caching only applies to validated certificates; if TOE has never validated a certificate, the TOE cache does not store the CRL for the issuing CA. The TOE stores a CRL only until it expires.

Section 6.4 of [ST] states the TOE supports both OCSP and CRL for certificate revocation checking on IPsec peer certificates, with CRL existing as a fallback option if an OCSP responder is unavailable. It also states when OCSP is used and a certificate is checked for revocation status the first time it is used, and once validated, the status is cached for one hour.

2.4.6.2 Guidance Activities

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

Section 6.8.1 of [CCECG] (“Syslog Server Connection Settings (Required)”) indicates the check of validity of certificates takes place during the TLS handshake.

The TOE does not use X.509v3 certificates for trusted updates or executable code integrity verification, instead using the Palo Alto Networks public key to verify the digital signature on an update image (see Section 7.12 of [CCECG]) and using an HMAC-SHA-256 key and ECDSA public key to verify software integrity during power-up (see Section 7.15 of [CCECG]). As such, the TOE does not use or check for certificates with the Code Signing purpose specified in the extendedKeyUsage field.

Section 6.8.1 of [CCECG] states the TOE checks revocation status of the certificate presented by the external syslog server during the TLS connection attempt and rejects the certificate if it has been revoked. The TOE performs the revocation check based on revocation information in the certificate. The TOE supports both OCSP and CRL checking.

2.4.6.3 Test Activities

The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols:

Test 1a: The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function and shall use this chain to demonstrate that the function succeeds. Test 1a shall be designed in a way that the chain can be 'broken' in Test 1b by either being able to remove the trust anchor from the TOEs trust store, or by

© 2024 Leidos. All rights reserved.

setting up the trust store in a way that at least one intermediate CA certificate needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store).

(TLS) The evaluator configured a certificate chain consisting of a “root” CA, a “top” intermediate CA signed by the root CA, a “bottom” intermediate CA signed by the “top” CA, and a leaf certificate signed by the “bottom” CA representing a test TLS server. The evaluator imported only the trusted root CA to the TOE, such that intermediate CA certificates are required to be provided in order to complete the chain. The evaluator configured a test TLS server to present the leaf certificate and the required intermediate CAs and verified that the TOE accepted the chain.

(IPsec) The evaluator configured the peer IPsec endpoint to present a certificate signed by an intermediate CA certificate and the intermediate CA certificate. The evaluator caused the peer IPsec endpoint to attempt to establish the IPsec channel with the TOE and observed that the connection was successfully established.

Test 1b: The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE’s trust store used to terminate the chain, or by removing one of the intermediate CA certificates (provided together with the leaf certificate in Test 1a) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.

(TLS) The evaluator configured the test TLS server to present the same certificate chain as configured in Test 1a, but omit the “top” intermediate CA, breaking the chain between the leaf and root CAs. The evaluator verified that the TOE failed to validate the certificate and rejected the connection.

(IPsec) The evaluator configured the peer IPsec endpoint to present a certificate which is signed by an intermediate CA certificate, but not include the intermediate CA certificate. The evaluator caused the peer IPsec endpoint to attempt to establish a connection with the TOE and observed that the connection was rejected by the TOE.

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

The evaluator attempted a TLS connection with the TOE using an expired certificate and confirmed that the TOE did not accept this connection.

The evaluator attempted an IPsec connection with the TOE using an expired certificate and confirmed that the TOE did not accept this connection.

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

© 2024 Leidos. All rights reserved.

The evaluator attempted TLS connections using both revoked and non-revoked peer and peer intermediate CA certificates to be checked via OCSP. The evaluator confirmed that use of non-revoked certificates resulted in a successful connection and revoked certificates resulted in the TOE denying the connection.

The evaluator attempted IPSEC connections using both revoked and non-revoked peer and peer intermediate CA certificates to be checked via CRL. The evaluator confirmed that use of non-revoked certificates resulted in a successful connection and revoked certificates resulted in the TOE denying the connection. The evaluator repeated this test using certificates to be checked via OCSP. The evaluator again confirmed that use of non-revoked certificates resulted in a successful connection and revoked certificates resulted in the TOE denying the connection.

Test 4a: [conditional] If OCSP is selected, the evaluator shall configure an authorized responder or use a man-in-the-middle tool to use a delegated OCSP signing authority to respond to the TOE's OCSP request. The resulting positive OCSP response (certStatus: good (0)) shall be signed by an otherwise valid and trusted certificate with the extendedKeyUsage extension that does not contain the OCSPSigning (OID 1.3.6.1.5.5.7.3.9). The evaluator shall verify that the TSF does not successfully complete the revocation check.

Note: Per RFC 6960 Section 4.2.2.2, the OCSP signature authority is delegated when the CA who issued the certificate in question is NOT used to sign OCSP responses.

The evaluator attempted a TLS connection using a valid certificate with revocation checking via OCSP. The OCSP response was signed with a certificate missing the OCSP Signing key purpose. The evaluator confirmed that the TOE did not accept the connection.

The evaluator attempted an IPsec connection using a valid certificate with revocation checking via OCSP. The OCSP response was signed with a certificate missing the OCSP Signing key purpose. The evaluator confirmed that the TOE did not accept the connection.

Test 4b: [conditional] If CRL is selected, the evaluator shall present an otherwise valid CRL signed by a trusted certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.

As stated in section 6.4 of [ST], the TOE does not support CRLs for revocation checking on TLS connections.

The evaluator attempted an IPsec connection using a valid certificate with revocation checking via CRL. The CRL was signed by a CA that did not have the CRL signing extension in its certificate. The evaluator confirmed that the TOE did not accept the connection.

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

The evaluator attempted a TLS connection with a modified byte in the first eight bytes of the certificate presented to the TOE and verified that the TOE did not accept the connection.

The evaluator attempted an IPsec connection with a modified byte in the first eight bytes of the certificate presented to the TOE and verified that the TOE did not accept the connection.

Test 6: The evaluator shall modify any byte in the certificate signatureValue field (see RFC 5280 Sec. 4.1.1.3), which is normally the last field in the certificate, and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

© 2024 Leidos. All rights reserved.

The evaluator attempted a TLS connection with a modified last byte of the certificate presented to the TOE and verified that the TOE did not accept the connection.

The evaluator attempted an IPsec connection with a modified last byte of the certificate presented to the TOE and verified that the TOE did not accept the connection.

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

The evaluator attempted a TLS connection with a modified byte in the public key of the certificate presented to the TOE and verified that the TOE did not accept the connection.

The evaluator attempted an IPsec connection with a modified byte in the public key of the certificate presented to the TOE and verified that the TOE did not accept the connection.

The following tests are run when a minimum certificate path length of three certificates is implemented.

Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The evaluator shall conduct the following tests:

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

The evaluator configured a certificate chain consisting of an EC root certificate, an EC intermediate certificate formatted as a named curve, and an EC leaf certificate. The evaluator then imported only the EC root certificate into the TOE. The evaluator configured a test TLS server to present this certificate chain. The evaluator had the TOE attempt to connect to this server and verified that the connection succeeded.

Note, this test is not applicable to IPsec as the TOE does not claim to utilize ECDSA certificates for the IPsec connection.

Test 8b: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a chain of EC certificates (terminating in a trusted CA certificate), where the intermediate certificate in the certificate chain uses an explicit format version of the Elliptic Curve parameters in the public key information field, and is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

The evaluator configured a certificate chain consisting of an EC root certificate, an EC intermediate certificate formatted as an explicitly-formatted curve, and an EC leaf certificate. The evaluator then imported only the EC root certificate into the TOE. The evaluator configured a test TLS server to present the leaf and intermediate certificates. The evaluator had the TOE attempt to connect to this server and verified that the connection failed.

© 2024 Leidos. All rights reserved.

Note, this test is not applicable to IPsec as the TOE does not claim to utilize ECDSA certificates for the IPsec connection.

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

The evaluator generated two intermediate certificates, one formatted with a named curve, and the second with an explicit format curve. The evaluator was able to load the certificate with the named curve onto the TOE. However, when the evaluator attempted to load the certificate with the explicit format curve, the TOE rejected the certificate.

FIA_X509_EXT.1.2/Rev

The evaluator shall perform the following tests for FIA_X509_EXT.1.2/Rev. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/Rev. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted.

The goal of the following tests is to verify that the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using basicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation).

For each of the following tests the evaluator shall create a chain of at least three certificates: a self-signed root CA certificate, an intermediate CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain).

Test 1: The evaluator shall ensure that at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator shall confirm that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

(TLS) The evaluator configured a certificate chain that consisted of a trusted root CA, a valid "top" intermediate CA signed by the root CA, a "bottom" intermediate CA that lacked the Basic Constraints extension signed by the Top CA, and the leaf certificate signed by the invalid Bottom CA. The evaluator configured a test TLS server to present this certificate chain. The evaluator had the TOE attempt to connect to this server and verified that the connection failed.

(IPsec) The evaluator generated a certificate that lacked the basicConstraints extension. The evaluator attempted to import the certificate into the TOE and observed that the TOE did not allow a certificate that lacks the basicConstraints extension to be imported into the TOE.

Test 2: The evaluator shall ensure that at least one of the CA certificates in the chain has a basicConstraints extension in which the CA flag is set to FALSE. The evaluator shall confirm that the TOE

© 2024 Leidos. All rights reserved.

rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

(TLS) The evaluator configured a certificate chain that consisted of a trusted root CA, a valid "top" intermediate CA signed by the root CA, a "bottom" intermediate CA that had its CA flag set to FALSE signed by the Top CA, and the leaf certificate signed by the invalid Bottom CA. The evaluator configured a test TLS server to present this certificate chain. The evaluator had the TOE attempt to connect to this server and verified that the connection failed.

(IPsec) The evaluator generated a certificate with the basicConstraints extension but with the CA flag set to FALSE. The evaluator attempted to import the certificate into the TOE and observed that the TOE did not allow the certificate to be imported into the TOE.

The evaluator shall repeat these tests for each distinct use of certificates. Thus, for example, use of certificates for TLS connection is distinct from use of certificates for trusted updates so both of these uses would be tested. But there is no need to repeat the tests for each separate TLS channel in FTP_ITC.1 and FTP_TRP.1/Admin (unless the channels use separate implementations of TLS).

The testing for FIA_X509_EXT.1/Rev was performed for each distinct use of certificates (TLS, IPsec). No differences were observed between them.

2.4.7 FIA_X509_EXT.2 X.509 Certificate Authentication

There is no change to the Evaluation Activities specified for this SFR in the [VPNGW-SD]; it only modifies this SFR to support its use for IPsec at a minimum. The evaluator shall ensure that all evaluation of this SFR is performed against its use in IPsec communications as well as any other supported usage.

2.4.7.1 TSS Activities

The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates.

Section 6.4 of [ST] ("Identification and Authentication") describes the TOE's usage of X.509 certificates for TLS and IPsec authentication. It is implicit that the TOE has its own TLS client/TLS server/IPsec certificate that it presents to remote entities, and the certificate it uses to validate the remote entity is the certificate that is provided to it during establishment of the trusted channel. Similarly, any intermediate/root CAs used by the TOE are implicit in the signer of any certificate that is presented to it.

The evaluator shall examine the TSS to confirm that it describes the behaviour of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the guidance documentation contains instructions on how this configuration action is performed.

Section 6.4 of [ST] states that a TLS certificate is rejected when its revocation status cannot be determined for syslog but that in all other cases, the administrator has the ability to configure the default behavior. In the evaluated configuration, the required action is to block the TLS session and this must be configured, it is not done by default.

© 2024 Leidos. All rights reserved.

2.4.7.2 Guidance Activities

The evaluator shall also ensure that the guidance documentation describes the configuration required in the operating environment so the TOE can use the certificates. The guidance documentation shall also include any required configuration on the TOE to use the certificates. The guidance document shall also describe the steps for the Security Administrator to follow if the connection cannot be established during the validity check of a certificate used in establishing a trusted channel.

Section 6.8.1 of [CCECG] ("Syslog Server Connection Settings (Required)") provides guidance to the administrator for configuring secure connections with an external syslog server and lists the reasons the server certificate will fail the validity check when attempting to establish a TLS connection.

2.4.7.3 Test Activities

The evaluator shall perform the following test for each trusted channel:

a. **Test 1**: The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the guidance documentation to determine that all supported administrator-configurable options behave in their documented manner.

This test was performed for IPsec and TLS.

The evaluator attempted to establish an IPsec connection with a peer whose certificate referenced a CRL distribution point. The evaluator powered down the server hosting the CRL distribution point. The evaluator observed the TOE attempted to query the CRL distribution point and rejected the connection once the revocation status of the presented certificate could not be established.

The evaluator then attempted to establish an IPsec connection with a peer whose certificate referenced an OCSP responder. The evaluator manipulated the OCSP server such that the responder was not running. The evaluator verified that the TOE attempted to connect to the OCSP server and verified that the connection failed after the TOE could not contact the OCSP responder.

The evaluator configured a TLS test server with a certificate that referenced an OCSP responder. The evaluator manipulated the OCSP server such that the responder was not running. The evaluator verified that the TOE attempted to connect to the OCSP server and verified that the connection failed after the TOE could not contact the OCSP responder.

2.4.8 FIA_X509_EXT.3 X.509 Certificate Requests

This SFR is modified to be mandatory when claiming MOD_VPNGW_v1.3, but there are no additional evaluation activities.

2.4.8.1 TSS Activities

If the ST author selects "device-specific information", the evaluator shall verify that the TSS contains a description of the device-specific fields used in certificate requests.

Device-specific information is not selected in FIA_X509_EXT.3.1.

© 2024 Leidos. All rights reserved.

2.4.8.2 Guidance Activities

The evaluator shall check to ensure that the guidance documentation contains instructions on requesting certificates from a CA, including generation of a Certificate Request Message. If the ST author selects “Common Name”, “Organization”, “Organizational Unit”, or “Country”, the evaluator shall ensure that this guidance includes instructions for establishing these fields before creating the Certification Request.

Section 6.8.1 of [CCECG] (“Syslog Server Connection Settings (Required)”) provides the guidance to generate or import X.509v3 certificates. The guidance includes instructions for establishing the “Common Name”, “Organization”, “Organizational Unit”, or “Country”, fields before creating the Certification Request.

2.4.8.3 Test Activities

Test 1: The evaluator shall use the guidance documentation to cause the TOE to generate a Certification Request. The evaluator shall capture the generated request and ensure that it conforms to the format specified. The evaluator shall confirm that the Certification Request provides the public key and other required information, including any necessary user-input information.

The evaluator generated a certificate request on the TOE and verified it was in the correct format and contained all the information specified by the Security Target.

Test 2: The evaluator shall demonstrate that validating a response message to a Certification Request without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the response message, and demonstrate that the function succeeds.

The evaluator attempted to import the signed response to the certificate request onto the TOE without the trusted CA imported and confirmed that the import was denied. The evaluator then imported the correct trusted CA and attempted to import the signed response and confirmed that the certificate was successfully imported.

2.5 Security Management (FMT)

General requirements for distributed TOEs

TSS Activities

For distributed TOEs, the evaluator shall verify that the TSS describes how every function related to security management is realized for every TOE component and shared between different TOE components. The evaluator shall confirm that all relevant aspects of each TOE component are covered by the FMT SFRs.

The TOE is not distributed so these activities are not applicable.

General requirements for distributed TOEs

Guidance Activities

For distributed TOEs, the evaluator shall verify that the Guidance Documentation to describe management of each TOE component. The evaluator shall confirm that all relevant aspects of each TOE component are covered by the FMT SFRs.

© 2024 Leidos. All rights reserved.

The TOE is not distributed so these activities are not applicable.

General requirements for distributed TOEs

Test Activities

Tests defined to verify the correct implementation of security management functions shall be performed for every TOE component. For security management functions that are implemented centrally, sampling should be applied when defining the evaluator's tests (ensuring that all components are covered by the sample).

The TOE is not distributed so these activities are not applicable.

2.5.1 FMT_MOF.1/ManualUpdate Management of Functions Behavior

2.5.1.1 TSS Activities

For distributed TOEs see chapter 2.4.1.1. There are no specific requirements for non-distributed TOEs.

The TOE is not distributed so this evaluation activity is not applicable.

2.5.1.2 Guidance Activities

The evaluator shall examine the guidance documentation to determine that any necessary steps to perform manual update are described. The guidance documentation shall also provide warnings regarding functions that may cease to operate during the update (if applicable).

Section 7.12 of [CCECG] ("Verify and Update System Software") provides instructions to initiate manual updates and states that a restart must occur for the update to be applied.

For distributed TOEs the guidance documentation shall describe all steps how to update all TOE components. This shall contain description of the order in which components need to be updated if the order is relevant to the update process. The guidance documentation shall also provide warnings regarding functions of TOE components and the overall TOE that may cease to operate during the update (if applicable).

The TOE is not distributed so this evaluation activity is not applicable.

2.5.1.3 Test Activities

Test 1: The evaluator shall try to perform the update using a legitimate update image without prior authentication as Security Administrator (either by authentication as a user with no administrator privileges or without user authentication at all – depending on the configuration of the TOE). The attempt to update the TOE shall fail.

The evaluator logged into the TOE as a non-administrative user and confirmed that they did not have the ability to update the TOE.

Test 2: The evaluator shall try to perform the update with prior authentication as Security Administrator using a legitimate update image. This attempt should be successful. This test case should be covered by the tests for FPT_TUD_EXT.1 already.

This test is performed in conjunction with FPT_TUD_EXT.1.

© 2024 Leidos. All rights reserved.

2.5.2 FMT_MOF.1/Services Management of Security Functions Behaviour

2.5.2.1 TSS Activities

For distributed TOEs see chapter 2.4.1.1.

For non-distributed TOEs, the evaluator shall ensure the TSS lists the services the Security Administrator is able to start and stop and how that how that operation is performed.

The TOE is not distributed so the first part of the evaluation activity is not applicable.

Section 6.5 of [ST] ("Security Management") states the Administrator is able to start and stop DNS and SNMP services via the **Device > Setup > Services > DNS Servers** and **Device > Setup > Operations > SNMP Setup** GUI controls.

2.5.2.2 Guidance Activities

For distributed TOEs see chapter 2.4.1.2.

For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation describes how the TSS lists the services the Security Administrator is able to start and stop and how that operation is performed.

Section 7.8 of [CCECG] ("Configure Device DNS or SNMP Service") describes how the administrator starts and stops the DNS and SNMP services, which are the only services identified in the TSS.

2.5.2.3 Test Activities

Test 1: The evaluator shall try to enable and disable at least one of the services as defined in the Application Notes for FAU_GEN.1.1 (whichever is supported by the TOE) without prior authentication as Security Administrator (either by authenticating as a user with no administrator privileges, if possible, or without prior authentication at all). The attempt to enable/disable this service/these services should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to enable/disable this service/these services can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator

The evaluator logged into the TOE as a non-administrative user and confirmed that a change to services could not be made without administrative privilege.

Test 2: The evaluator shall try to enable and disable at least one of the services as defined in the Application Notes for FAU_GEN.1.1 (whichever is supported by the TOE) with prior authentication as Security Administrator. The attempt to enable/disable this service/these services should be successful.

The evaluator verified that the administrator could make changes to the TOE's services.

2.5.3 FMT_MTD.1/CryptoKeys Management of TSF Data

This SFR is modified by MOD_VPNGW_v1.3 to make it mandatory, but there are no modified or additional evaluation activities.

© 2024 Leidos. All rights reserved.

2.5.3.1 TSS Activities

For distributed TOEs see chapter 2.4.1.1.

For non-distributed TOEs, the evaluator shall ensure the TSS lists the keys the Security Administrator is able to manage to include the options available (e.g. generating keys, importing keys, modifying keys or deleting keys) and names the operations that are performed.

The TOE is not distributed so the first part of the evaluation activity is not applicable.

Section 6.5 of [ST] ("Security Management") identifies the mechanisms available for the Security Administrator to generate, manage, import, and delete keys, and the keys the Security Administrator can manage.

2.5.3.2 Guidance Activities

For distributed TOEs see chapter 2.4.1.2.

For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation describes how the operations are performed on the keys the Security Administrator is able to manage.

The TOE is not distributed so this part of the evaluation activity is not applicable.

Section 6.8.1 of [CCECG] ("Syslog Server Connection Settings (Required)") describes how the administrator can generate, import, and delete X.509 certificates and associated key pairs.

2.5.3.3 Test Activities

Test 1: The evaluator shall try to perform at least one of the related actions (modify, delete, generate/import) without prior authentication as Security Administrator (either by authentication as a non-administrative user, if supported, or without authentication at all). Attempts to perform related actions without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to manage cryptographic keys can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

The evaluator attempted to perform management actions on cryptographic key items as a user that lacked administrator privileges and observed that the attempts failed.

Test 2: The evaluator shall try to perform at least one of the related actions with prior authentication as Security Administrator. This attempt should be successful.

The evaluator performed this test as part of testing for FIA_X509_EXT.1, which showed the administrator was able to generate a CSR, including generation of a private key, and as part of testing for FIA_X509_EXT.1, Test 2, which shows the administrator is able to import a certificate into the trust store.

© 2024 Leidos. All rights reserved.

2.5.4 FMT_MTD.1/CoreData Management of TSF Data

2.5.4.1 TSS Activities

For each administrative function identified in the guidance documentation that is accessible through an interface prior to administrator log-in, the evaluator shall confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.

Section 6.4 of [ST] ("Identification and Authentication") states the only capabilities allowed prior to users authenticating are display of the login banner and responding to ICMP requests (e.g., ping or ICMP echo reply).

If the TOE supports handling of X.509v3 certificates and implements a trust store, the evaluator shall examine the TSS to determine that it contains sufficient information to describe how the ability to manage the TOE's trust store is restricted.

Section 6.5 of [ST] ("Security Management") states that role-based privileges on the CLI and web GUI/REST API are used to ensure that only authorized administrators can configure TSF behavior, which includes managing X.509v3 certificates in the trust store.

2.5.4.2 Guidance Activities

The evaluator shall review the guidance documentation to determine that each of the TSF-data-manipulating functions implemented in response to the requirements of the cPP is identified, and that configuration information is provided to ensure that only administrators have access to the functions.

The following lists the security management functions for the TOE as claimed by [ST] and where in the vendor's documentation the usage of these functions is described:

- Ability to administer the TOE remotely—[CCECG] sections 5.1.1 ("User Login to Web Interface") and 5.1.2 ("User Login to CLI Remotely"). The guidance notes that local console access is disabled when the TOE is in FIPS-CC mode and management is performed remotely.
- Resetting passwords—Section 7.3.4 of [CCECG] ("Change User Password")
- Ability to configure the access banner—Section 7.5 of [CCECG] ("Configure Login Banner")
- Ability to configure the session inactivity time before session termination or locking—Section 7.6 of [CCECG] ("Configure Idle Timeout and Lockout")
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates—Section 7.12 of [CCECG] ("Verify and Update System Software")
- Ability to configure the authentication failure parameters for FIA_AFL.1—Section 7.6 of [CCECG] ("Configure Idle Timeout and Lockout")
- Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1—[CCECG] sections 7.5 ("Configure Login Banner") and 7.8 ("Configure Device DNS or SNMP Service")
- Ability to configure the cryptographic functionality—[CCECG] sections 6.2 ("Enable FIPS-CC Mode (Required)"), 6.4 ("Configure SSH Encryption and Integrity Algorithms (Required)"), and 6.6 ("Configure SSH Public-Key Authentication (Recommended)")
- Ability to configure thresholds for SSH rekeying—Section 6.5 of [CCECG] ("Configure SSH Rekey Interval (Required)")
- Ability to set the time which is used for time-stamps—Section 7.4 of [CCECG] ("Configure System Time")

© 2024 Leidos. All rights reserved.

- Ability to import X.509v3 certificates to the TOE's trust store—[CCECG] sections 6.8.1 ("Syslog Server Connection Settings (Required)"), and 7.2 ("Configure Custom HTTPS or TLS Server Certificate")
- Ability to manage the TOE's trust store and designate X.509v3 certificates as trust anchor—[CCECG] sections 6.8.1 ("Syslog Server Connection Settings (Required)")
- Ability to configure firewall rules—Section 7.9 of [CCECG] ("Configure Stateful Inspection Filtering")
- Ability to configure the lifetime for IPsec SAs—Section 7.10 of [CCECG] ("Configure IKE/IPsec VPN Gateway")
- Ability to configure the reference identifiers for the peer—Section 7.10 of [CCECG] ("Configure IKE/IPsec VPN Gateway")
- Ability to configure the IPsec functionality—Section 7.10 of [CCECG] ("Configure IKE/IPsec VPN Gateway")
- Ability to configure audit behavior—[CCECG] sections 4 ("Required Auditable Events"), 6.7 ("Configure Auditing Settings"), and 6.8.1 ("Syslog Server Connection Settings (Required)")
- Ability to start and stop services—Section 7.8 of [CCECG] ("Configure Device DNS or SNMP Service")
- Ability to manage the trusted public keys database—Section 6.6 of [CCECG] ("Configure SSH Public-Key Authentication (Recommended)")
- Definition of packet filtering rules—Section 7.9 of [CCECG] ("Configure Stateful Inspection Filtering")
- Association of packet filtering rules to network interfaces—Section 7.9 of [CCECG] ("Configure Stateful Inspection Filtering")
- Ordering of packet filter rules by priority—Section 7.9 of [CCECG] ("Configure Stateful Inspection Filtering")
- Ability to configure the IPS functionality (i.e., all management functions specified in FMT_SMF.1/IPS)—Section 7.11 of [CCECG] ("Configure Threat Prevention").

Section 7.3.2 of [CCECG] ("Adding New Accounts") identifies the Admin Roles that can be assigned to a user. The Admin Roles consist of the following:

- Superuser – Full read-write access to Device.
- Superuser (Read Only) – Read-only access to Device.
- Device administrator – Full access to Device except for the following actions:
  - Create, modify, or delete user and roles.
  - Export, validate, revert, save, load, or import a configuration
  - Configure a **Scheduled Config Export** in the **Device** tab.

These administrator roles (except Read-Only) are considered Security Administrator as defined in the [NDcPP]. The role that is assigned to a user ensures that only administrators have access to the configuration information.

If the TOE supports handling of X.509v3 certificates and provides a trust store, the evaluator shall review the guidance documentation to determine that it provides sufficient information for the administrator to configure and maintain the trust store in a secure way. If the TOE supports loading of CA certificates, the evaluator shall review the guidance documentation to determine that it provides sufficient information for the administrator to securely load CA certificates into the trust store. The evaluator shall also review the guidance documentation to determine that it explains how to designate a CA certificate a trust anchor.

© 2024 Leidos. All rights reserved.

Sections 6.8.1 and 7.2 of [CCECG] provide information for the administrator to configure and maintain the trust store in a secure way. The information includes loading of CA certificates. When the administrator imports a CA certificate or generates a CA certificate internally, the TOE implicitly sets the CA certificate as a trust anchor.

2.5.4.3 Test Activities

No separate testing for FMT_MTD.1/CoreData is required unless one of the management functions has not already been exercised under any other SFR.

All management functions are covered by testing associated with other SFRs.

2.5.5 FMT_SMF.1, FMT_SMF.1/FFW, and FMT_SMF.1/VPN Specification of Management Functions

The security management functions for FMT_SMF.1 are distributed throughout the cPP and are included as part of the requirements in FTA_SSL_EXT.1, FTA_SSL.3, FTA_TAB.1, FMT_MOF.1/ManualUpdate, FMT_MOF.1/AutoUpdate (if included in the ST), FIA_AFL.1, FIA_X509_EXT.2.2 (if included in the ST), FPT_TUD_EXT.1.2 & FPT_TUD_EXT.2.2 (if included in the ST and if they include an administrator-configurable action), FMT_MOF.1/Services, and FMT_MOF.1/Functions (for all of these SFRs that are included in the ST), FMT_MTD, FPT_TST_EXT, and any cryptographic management functions specified in the reference standards. Compliance to these requirements satisfies compliance with FMT_SMF.1.

FMT_SMF.1/FFW (FW-SD)
The evaluation activities specified for FMT_SMF.1 in the Supporting Document for the Base-PP shall be applied in the same way to the newly added management functions defined in FMT_SMF.1/FFW in the FW Module.

See above – no explicit evaluation activities are identified so this behavior is addressed through the other management SFRs.

2.5.5.1 TSS Activities (also including activities for Guidance Documentation and Tests)

The evaluator shall examine the TSS, Guidance Documentation and the TOE as observed during all other testing and shall confirm that the management functions specified in FMT_SMF.1 are provided by the TOE. The evaluator shall confirm that the TSS details which security management functions are available through which interface(s) (local administration interface, remote administration interface).

Section 6.5 of [ST] (“Security Management”) lists the supported management functions specified in FMT_SMF.1, FMT_SMF.1/FFW, FMT_SMF.1/VPN, and FMT_SMF.1/IPS. The evaluation activities for the relevant SFRs demonstrate the proper implementation of these functions. This section also states that the TOE provides the ability to administer the TOE remotely; and that the GUI, CLI, and API (XML and REST) provide identical management functionality.

The evaluator shall examine the TSS and Guidance Documentation to verify they both describe the local administrative interface. The evaluator shall ensure the Guidance Documentation includes appropriate warnings for the administrator to ensure the interface is local.

© 2024 Leidos. All rights reserved.

The evaluator examined the TSS and Guidance Documentation and verified they describe the remote interfaces. This is consistent with the ST that does not select local authentication mechanisms in FIA_UIA_EXT.1.3; and the PP v3.0 that makes local administration optional in FMT_SMF.1 (note it is a selection and see also App Note 23).

For distributed TOEs with the option 'ability to configure the interaction between TOE components' the evaluator shall examine that the ways to configure the interaction between TOE components is detailed in the TSS and Guidance Documentation. The evaluator shall check that the TOE behaviour observed during testing of the configured SFRs is as described in the TSS and Guidance Documentation.

The TOE is not distributed, so this activity is N/A.

VPNGW-SD

The evaluator shall examine the TSS to confirm that all management functions specified in FMT_SMF.1/VPN are provided by the TOE. As with FMT_SMF.1 in the Base-PP, the evaluator shall ensure that the TSS identifies what logical interfaces are used to perform these functions and that this includes a description of the local administrative interface.

Section 6.5 of [ST] identifies the TOE's supported management functions, which includes the relevant VPN functionality claimed in FMT_SMF.1/VPN. This section states that the GUI, CLI, and API (XML and REST) provide identical management functionality.

2.5.5.2 Guidance Activities

VPNGW-SD

The evaluator shall examine the operational guidance to confirm that all management functions specified in FMT_SMF.1/VPN are provided by the TOE. As with FMT_SMF.1 in the Base-PP, the evaluator shall ensure that the operational guidance identifies what logical interfaces are used to perform these functions and that this includes a description of the local administrative interface.

Section 2.5.4.2 above lists the supported management functions and where in the vendor documentation their use is described. The operational guidance identifies the logical interfaces used to perform each specified function. The ST does not claim the ability to administer the TOE locally, so the operational guidance does not describe any local administrative interface.

2.5.5.3 Test Activities

The evaluator tests management functions as part of testing the SFRs identified in section 2.4.4. No separate testing for FMT_SMF.1 is required unless one of the management functions in FMT_SMF.1.1 has not already been exercised under any other SFR.

The evaluator covered the testing of management functions as follows:

- Ability to administer the TOE remotely.
  - This function is exercised testing FIA_AFL.1. In that test, the evaluator utilizes each method of remotely accessing the TOE.
- Ability to configure the access banner.
  - This function is exercised testing FTA_TAB.1. In that test, the evaluator configured the access banner and verified that it was displayed.
- Ability to configure the session inactivity time before session termination.

© 2024 Leidos. All rights reserved.

- This function is exercised testing FTA_SSL.3. In that test, the evaluator verified that the session timeout value could be configured.

- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates.
  - This function is exercised testing FPT_TUD_EXT.1. In the tests for this SFR, the evaluator verified that the TOE could apply updates and that those updates were digitally signed with a trusted signature.
- Ability to manage the cryptographic keys.
  - This function is exercised testing FIA_X509_EXT.3. In that test, the evaluator generates a new cryptographic keypair by generating a CSR.
- Ability to configure the cryptographic functionality.
  - This function is exercised when placing the device into FIPS-CC mode. This mode restricts the TOE to use approved algorithms and DRBGs along with other FIPS-related configuration.
- Ability to configure the list of supported (D)TLS ciphers.
  - This function is exercised testing FCS_TLSS_EXT.1.5 Test 1. In that test, the evaluator configured the list of advertised ciphersuites and verified that the TOE responded appropriately.
- Ability to configure the lifetime for IPsec SAs.
  - This function is exercised testing FCS_IPSEC_EXT.1.7. In that test, the evaluator configures a lifetime for IPsec SAs and verifies that the TOE enforces the lifetimes.
- Ability to generate Certificate Signing Request (CSR) and process CA certificate response.
  - This function is exercised testing FIA_X509_EXT.3. In the tests for that SFR, the evaluator generates a CSR and imports the resulting CA response into the TOE.
- Ability to start and stop services.
  - This function is exercised testing FMT_MOF.1/Services. In that test, the evaluator verifies that the TOE allows an administrator to configure the services.
- Ability to configure audit behavior (e.g. changes to storage locations for audit; changes to behavior when local audit storage space is full).
  - This function is exercised in FAU_STG_EXT.1 Test 4. In that test, the evaluator configured the logging behavior and allocation of log space on the TOE.
- Ability to modify the behavior of the transmission of audit data to an external IT entity.
  - This function is exercised in FAU_STG_EXT.1 Test 1. In that test, the evaluator configured the TOE to export audit records to an external audit server over TLS.
- Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1.
  - This function is exercised in FIA_UIA_EXT.1 Test 2 and testing FTA_TAB.1. In those tests, the evaluator configures the login banner and verifies that it and the Ping functionality are the only allowed actions to a non-authenticated entity.
- Ability to configure thresholds for SSH rekeying.
  - This function is exercised testing FCS_SSH_EXT.1.8. In that test, the administrator configured the rekey thresholds and verified that the TOE enforced the thresholds.
- Ability to set the time which is used for time-stamps.
  - This function is exercised in FPT_STM_EXT.1 Test 1. In that test, the evaluator verified that the administrator could set the time and that it was reflected on the TOE.
- Ability to configure the reference identifier for the peer.

© 2024 Leidos. All rights reserved.

- This function is exercised testing FCS_TLSC_EXT.1.2. In the testing for that SFR, the evaluator configured the reference identifier and verified that the TOE verified the peer using the reference identifier.

- Ability to manage the TOE's trust store and designate X509v3 certificates as trust anchors.
  - This function is exercised in FIA_X509_EXT.3 Test 2. In that test, the evaluator added certificates to the TOE's trust store in order to create a trusted certificate path for the response from the CSR.
- Ability to manage the trusted public keys database.
  - This function is exercised in FIA_UIA_EXT.1 Test 1. In that test, the evaluator configures the TOE to accept a public key for a user.
- Ability to configure the authentication failure parameters for FIA_AFL.1.
  - This function is exercised in FIA_AFL.1 Test 1.

VPNGW-SD

The evaluator tests management functions as part of performing other test EAs. No separate testing for FMT_SMF.1/VPN is required unless one of the management functions in FMT_SMF.1.1/VPN has not already been exercised under any other SFR.

This test is performed in conjunction with FPF_RUL_EXT.1.6.

2.5.6 FMT_SMF.1/IPS

2.5.6.1 TSS Activities

The evaluator shall verify that the TSS describes how the IPS data analysis and reactions can be configured. This may be performed in conjunction with the evaluation of IPS_ABD_EXT.1, IPS_IPB_EXT.1, and IPS_SBD_EXT.1.

This activity is performed in conjunction with IPS_ABD_EXT.1 (see section 2.11.1.1); IPS_IPB_EXT.1 (see section 2.11.2.1, and IPS_SBD_EXT.1 (see section 2.11.4.1).

2.5.6.2 Guidance Activities

The evaluator shall verify that the operational guidance describes the instructions for each function defined in the SFR, describes how to configure the IPS data analysis and reactions, including how to set any configurable defaults and how to configure each of the applicable analysis pattern matching methods and reaction modes.

The operational guidance provides the instructions for each function defined in the SFR, describes how to configure the IPS data analysis and reactions, including how to set any configurable defaults and how to configure each of the applicable analysis pattern matching methods and reaction modes. The guidance for each specified management function is provided in [CCECG] as follows:

- Enable, disable signatures applied to sensor interfaces, and determine the behavior of IPS functionality—section 7.11.2 ("Configure Anomaly-Based Detection")
- Modify the parameters (source/dest IP address, source/dest port, protocol, ICMP type/code) that define the network traffic to be collected and analyzed—section 7.11.4 ("Configure L3 & L4 Header Rules") and section 7.9.1 ("Zone Protection Profile")
- Update (import) signatures—section 7.13 ("Dynamic Updates")
- Create custom signatures—section 7.11.6 ("Configure Vulnerability/Threat Signature Rules")
- Configure anomaly detection—section 7.11.2 ("Configure Anomaly-Based Detection")

© 2024 Leidos. All rights reserved.

- Enable and disable actions to be taken when signature or anomaly matches are detected—section 7.11.2 (“Configure Anomaly-Based Detection”) and section 7.11.6 (“Configure Vulnerability/Threat Signature Rules”)
- Modify thresholds that trigger IPS reactions—section 7.11.2 (“Configure Anomaly-Based Detection”) and section 7.11.4 (“Configure L3 & L4 Header Rules”)
- Modify the duration of traffic blocking actions—section 7.11.2 (“Configure Anomaly-Based Detection”)
- Modify the known-good and known-bad lists (of IP addresses or address ranges)—section 7.11.3 (“Configure External Dynamic List”)
- Configure the known-good and known-bad lists to override signature-based IPS policies—section 7.11.3 (“Configure External Dynamic List”).

2.5.6.3 Test Activities

Test 1: The evaluator shall use the operational guidance to create a signature and enable it on an interface. The evaluator shall then generate traffic that would be successfully triggered by the signature. The evaluator should observe the TOE applying the corresponding reaction in the signature.

The evaluator created a signature and enabled it on the TOE. The evaluator verified that the TOE enforced the configured signature.

Test 2: The evaluator shall then disable the signature and attempt to regenerate the same traffic and ensure that the TOE allows the traffic to pass with no reaction.

The evaluator verified that the TOE only reacts to configured and enabled signatures.

Test 3: The evaluator shall use the operational guidance to import signatures and repeat the test conducted in Test 1.

The evaluator verified that signatures could be imported into the TOE.

Other testing for this SFR is performed in conjunction with the EAs for IPS_ABD_EXT.1 and IPS_SBD_EXT.1.

The evaluator verified the signatures effectiveness was tested as part of IPS_ABD_EXT.1 and IPS_SBD_EXT.1

2.5.7 FMT_SMR.2 Restrictions on Security Roles

2.5.7.1 TSS Activities

The evaluator shall examine the TSS to determine that it details the TOE supported roles and any restrictions of the roles involving administration of the TOE (e.g. if local administrators and remote administrators have different privileges or if several types of administrators with different privileges are supported by the TOE).

Section 6.5 of [ST] (“Security Management”) identifies the pre-defined Superuser, Superuser (Read-Only), and Device Administrator roles supported by the TOE and states that an account can only be assigned one role at a time.

© 2024 Leidos. All rights reserved.

2.5.7.2 Guidance Activities

The evaluator shall review the guidance documentation to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration.

Section 5 of [CCECG] ("Identification and Authentication") states the TOE has a web interface that users can use to perform administrative, management, and analysis tasks. Users can access the web interface by logging into the TOE using a web browser. The table below identifies the web browser compatibility and required options and settings.

Browser	Required Enabled Options and Settings
Chrome (version 119 or later)	JavaScript, cookies, Transport Layer Security (TLS) v1.2 and v1.3

In addition, the TOE provides a CLI and API that can be used to manage the TOE remotely. These interfaces provide the equivalent operations provided by the web interface. When accessing the CLI, the client computer requires the use of an SSHv2 client installed. The GUI and API are accessed over HTTPS or tunneled using IPsec.

The TOE in the evaluated configuration only supports SSH, HTTPS, and IPsec security protocols for management. Telnet and HTTP are not enabled for management and must not be enabled.

The [CCECG] provides instructions for configuring each of the management functions using GUI, CLI, and API.

The API is accessible remotely over HTTPS. References throughout [CCECG] labeled "API HINT" provide instructions for the API equivalent for how to perform a management function being discussed at that point in the guidance.

2.5.7.3 Test Activities

In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this cPP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH, if the TSF shall be validated against the Functional Package for Secure Shell referenced in Section 2.2 of the cPP; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.

The evaluator performed this testing throughout overall testing where administrative actions were performed. While testing was generally done using the CLI or GUI, a subset of management functions was repeated using the API to demonstrate its proper function.

© 2024 Leidos. All rights reserved.

2.6 Protection of the TSF (FPT)

2.6.1 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

2.6.1.1 TSS Activities

The evaluator shall examine the TSS to determine that it details how any preshared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through any interface designed specifically for that purpose, by any enabled role, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

Section 6.6 of [ST] ("Protection of the TSF") states that all secret and private key data is stored encrypted with a Master Key using 256-bit AES. The TOE does not provide an interface to read the Master Key.

2.6.1.2 Guidance Activities

None

2.6.1.3 Test Activities

None

2.6.2 FPT_APW_EXT.1 Protection of Administrator Passwords

2.6.2.1 TSS Activities

The evaluator shall examine the TSS to determine that it details all authentication data that are subject to this requirement, and the method used to obscure the plaintext password data when stored. The TSS shall also detail passwords are stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note.

Section 6.6 of [ST] ("Protection of the TSF") states the TOE protects the confidentiality of user passwords by hashing the passwords using SHA-256. It also states that certificates (used for authentication) and their associated key data are stored in a PKCS#12 file which stores the X.509 certificate and encrypted private key. The TOE does not offer any functions that will disclose to any users a stored cryptographic key or password.

2.6.2.2 Guidance Activities

None

2.6.2.3 Test Activities

None

© 2024 Leidos. All rights reserved.

2.6.3 FPT_FLS.1/SelfTest Failure with Preservation of Secure State (Self-test Failures) (VPNGW-SD)

2.6.3.1 TSS Activities

The evaluator shall ensure the TSS describes how the TOE ensures a shutdown upon a self-test failure, a failed integrity check of the TSF executable image, or a failed health test of the noise source. If there are instances when a shutdown does not occur, (e.g., a failure is deemed non- security relevant), the evaluator shall ensure that those cases are identified and a rationale is provided that supports the classification and justifies why the TOE's ability to enforce its security policies is not affected in any such instance.

Section 6.6 of [ST] ("Protection of the TSF") describes how all power-up self-test failures will cause the module to reboot and enter maintenance mode (i.e., error state) in which the reason for the failure can be determined. In the maintenance mode, all operational and network functions will be unavailable with one exception, which is the Reboot operation. No cryptographic operations or functions are performed and all data output from the TOE is inhibited. [ST] does not identify any self-test failures that are not security relevant.

2.6.3.2 Guidance Activities

The evaluator shall verify that the operational guidance provides information on the self-test failures that can cause the TOE to shut down and how to diagnose the specific failure that has occurred, including possible remediation steps if available.

Section 7.15 of [CCECG] ("Self-Tests") identifies the self-test failures that can cause the TSF to shut down: Firmware Integrity failure; Known Answer Test (KAT) failures; and Entropy Health Test. The TSF enters maintenance mode where the TOE is no longer in the evaluated configuration. In this state (an error state) the TOE will output an error indicator. At this point, the TSF has "shutdown"; no cryptographic operations are performed; and the appliance can only be manually re-booted.

Section 7.15 of [CCECG] describes the possible failures and shows a sample log that can be viewed in the system logs. The sample log shows the output of a successfully executed self-test. Each self-test that is executed has an indication of what the test is and whether or not it has passed or failed. The guidance states that if a self-test fails, to re-boot the appliance and if the self-tests continue to fail, contact Palo Alto Networks Support—the guidance provides email and phone contact information.

2.6.3.3 Test Activities

There are no test EAs for this component.

2.6.4 FPT_STM_EXT.1 Reliable Time Stamps

2.6.4.1 TSS Activities

The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time, and that it provides a description of how the time is maintained and considered reliable in the context of each of the time related functions.

© 2024 Leidos. All rights reserved.

Section 6.6 of [ST] (“Protection of the TSF”) states the TOE uses time data for audit record time stamps, measuring session activity for termination, administrator lockout, SSH/IPsec session rekeying, and X.509 certificate validation. Time data is reliable through use of an internal hardware-based real-time clock. Section 6.6 includes a description of this clock and states how it is used by the TOE for functionality that requires timestamps.

If “obtain time from the underlying virtualization system” is selected, the evaluator shall examine the TSS to ensure that it identifies the VS interface the TOE uses to obtain time. If there is a delay between updates to the time on the VS and updating the time on the TOE, the TSS shall identify the maximum possible delay.

This assurance activity is not applicable because the TOE does not obtain time from an underlying VS.

2.6.4.2 Guidance Activities

The evaluator shall examine the guidance documentation to ensure it instructs the administrator how to set the time. If the TOE supports the use of an NTP server, the guidance documentation instructs how a communication path is established between the TOE and the NTP server, and any configuration of the NTP client on the TOE to support this communication.

Section 7.4 of [CCECG] (“Configure System Time”) provides the instructions for how an administrator can manually set the time using the web UI, CLI, or XML API. The TOE does not support NTP.

If the TOE supports obtaining time from the underlying VS, the evaluator shall verify the Guidance Documentation specifies any configuration steps necessary. If no configuration is necessary, no statement is necessary in the Guidance Documentation. If there is a delay between updates to the time on the VS and updating the time on the TOE, the evaluator shall ensure the Guidance Documentation informs the administrator of the maximum possible delay.

Section 6.6 of [ST] states for VM-Series virtual appliances, the hardware hosting the VM-Series provides the time clock. Section 7.4 of [CCECG] directs the administrator, for PAN-OS VM on Hyper-V, to disable the “Time Synchronization” setting in Hyper-V to allow time change on the VM.

2.6.4.3 Test Activities

Test 1: If the TOE supports direct setting of the time by the Security Administrator then the evaluator shall use the guidance documentation to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.

The evaluator queried the time on the TOE then attempted to change the time. The evaluator queried the time again and verified that the time successfully changed.

Test 2: If the TOE supports the use of an NTP server; the evaluator shall use the guidance documentation to configure the NTP client on the TOE and set up a communication path with the NTP server. The evaluator shall observe that the NTP server has set the time to what is expected. If the TOE supports multiple protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the guidance documentation.

This test is not applicable because the TOE does not support use of an NTP server.

© 2024 Leidos. All rights reserved.

Test 3 [conditional]: If the TOE obtains time from the underlying VS, the evaluator shall record the time on the TOE, modify the time on the underlying VS, and verify the modified time is reflected by the TOE. If there is a delay between the setting the time on the VS and when the time is reflected on the TOE, the evaluator shall ensure this delay is consistent with the TSS and Guidance.

This test is not applicable because the TOE does not obtain time from an underlying VS.

If the audit component of the TOE consists of several parts with independent time information, then the evaluator shall verify that the time information between the different parts are either synchronized or that it is possible for all audit information to relate the time information of the different part to one base information unambiguously.

The TOE writes audit records to one of five different log files, depending on the type of audit record. However, the TOE is a single component it uses its system clock to provide the time stamp for the audit records, so the time information in all audit records is unambiguously related to the TOE's single time source.

2.6.5 FPT_TST_EXT.1 TSF Testing

This SFR is modified by MOD_VPNGW_v1.3, but there are no modified or additional evaluation activities.

2.6.5.1 TSS Activities

Modified in accordance with TD0836.

The evaluator shall examine the TSS to ensure that it details each of the self-tests that are ~~run~~ identified by the ~~TSF~~ SFR; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly. If more than one failure response is listed in FPT_TST_EXT.1.2, the evaluator shall examine the TSS to ensure it clarifies which response is associated with which type of failure.

Section 6.6 of [ST] ("Protection of the TSF") details each of the self-tests listed in the requirement, comprising: integrity verification of TSF executable code and stored TSF data; cryptographic known answer tests; and entropy health tests. The description outlines what the tests do. The cryptographic known-answer tests operate the cryptographic algorithm on data for which the correct output is already known and compare the calculated output with this known answer. If the calculated result does not equal the known answer, the test fails. The firmware integrity test verifies the signature of the firmware image to ensure that it has not been modified or tampered with. If the signature verification fails, the integrity test will fail. The entropy health tests consist of two tests, RCT (Repetition Count Test) and APT (Adaptive Proportion Test). RCT is designed to detect if the noise source becomes stuck outputting the same value too many times in a row. APT is designed to detect if the frequency of a sample value occurs too many times in a particular window size. The entropy health test will fail if either the RCT or the APT fails.

The TSS argues that this is sufficient to ensure correct functionality of the TSF because the self-tests encompass the cryptographic functionality and the integrity of the entire TOE software executable code.

The statement of FPT_TST_EXT.1.2 specifies the TOE responds to all failures by entering an error state called "maintenance mode".

© 2024 Leidos. All rights reserved.

For distributed TOEs the evaluator shall examine the TSS to ensure that it details which TOE component performs which self-tests and when these selftests are run. The evaluator shall also examine the TSS to ensure it describes how the TOE reacts if one or more TOE components fail self-testing (e.g. halting and displaying an error message; failover behaviour).

The TOE is not distributed so this evaluation activity is not applicable.

2.6.5.2 Guidance Activities

The evaluator shall also ensure that the guidance documentation describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors shall correspond to those described in the TSS.

Section 7.15 of [CCECG] ("Self-Tests") states the TOE performs a suite of FIPS self-tests during power-up and when rebooted. If any self-test fails, the TOE will enter maintenance mode (i.e., no longer in the evaluated configuration). The TOE enters an error state and outputs an error indicator. The TOE does not perform cryptographic operations and inhibits all data output while in the error state. The guidance instructs the administrator to re-boot the appliance in the event a self-test fails and to contact Palo Alto Networks Support if the self-tests continue to fail. The guidance provides contact information for Palo Alto Networks Support, in terms of phone number and email address.

For distributed TOEs the evaluator shall ensure that the guidance documentation describes how to determine from an error message returned which TOE component has failed the self-test.

The TOE is not distributed so this evaluation activity is not applicable.

2.6.5.3 Test Activities

Modified in accordance with TD0836.

It is expected that at least the following tests are performed:

a. Verification of the integrity of the firmware and executable software of the TOE
b. Verification of the correct operation of the cryptographic functions necessary to fulfil any of the SFRs.

Although formal compliance is not mandated, the self-tests performed should aim for a level of confidence comparable to:

a. [FIPS 140-2], Section 4.9.1, Software/firmware integrity test for the verification of the integrity of the firmware and executable software. Note that the testing is not restricted to the cryptographic functions of the TOE.
b. [FIPS 140-2], Section 4.9.1, Cryptographic algorithm test for the verification of the correct operation of cryptographic functions. Alternatively, national requirements of any CCRA member state for the security evaluation of cryptographic functions should be considered as appropriate.

The evaluator shall ~~either~~ verify that the self-tests described above are carried out ~~during initial start-up or that the developer has justified any deviation from this~~ according to the SFR and in agreement with the descriptions in the TSS.

The evaluator verified through audit records that the TOE's claimed self-tests ran successfully at boot time.

© 2024 Leidos. All rights reserved.

For distributed TOEs the evaluator shall perform testing of self-tests on all TOE components according to the description in the TSS about which self-test are performed by which component.

The TOE is not distributed and therefore this evaluation activity is not applicable.

2.6.6 FPT_TST_EXT.3 Self-Test with Defined Methods (VPNGW-SD)

2.6.6.1 TSS Activities

The evaluator shall verify that the TSS describes the method used to perform self-testing on the TSF executable code, and that this method is consistent with what is described in the SFR.

Section 6.6 of [ST] ("Protection of the TSF") states the TOE's firmware integrity test uses ECDSA as defined in FCS_COP.1/SigGen. This matches the description in the SFR.

2.6.6.2 Guidance Activities

There are no guidance EAs for this component.

2.6.6.3 Test Activities

There are no test EAs for this component.

2.6.7 FPT_TUD_EXT.1 Trusted Update

This SFR is modified by MOD_VPNGW_v1.3, but there are no modified or additional evaluation activities.

2.6.7.1 TSS Activities

The evaluator shall verify that the TSS describe how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the TSS shall describe how and when the inactive version becomes active. The evaluator shall verify this description.

Section 6.6 of [ST] ("Protection of the TSF") identifies the UI and CLI/API commands that are used to show the current software version.

The TSF does not contain a delayed activation mechanism for downloaded updates, so this is not discussed in the TSS.

The evaluator shall verify that the TSS describes all TSF software update mechanisms for updating the system firmware and software (for simplicity the term 'software' will be used in the following although the requirements apply to firmware and software). The evaluator shall verify that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails. The evaluator shall verify that the TSS describes the method by which the digital signature is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature of the update, and the actions that take place for both successful and unsuccessful signature verification.

Section 6.6 of [ST] describes how to check for and obtain (download/upload) updates. As part of the download activity, the update's 2048-bit RSA digital signature is checked. The update is only installed if the signature verification is successful.

© 2024 Leidos. All rights reserved.

If the options 'support automatic checking for updates' or 'support automatic updates' are chosen from the selection in FPT_TUD_EXT.1.2, the evaluator shall verify that the TSS explains what actions are involved in automatic checking or automatic updating by the TOE, respectively.

[ST] does not claim automatic checking or application of updates, so this activity is not applicable.

For distributed TOEs, the evaluator shall examine the TSS to ensure that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component. Alternatively, this description can be provided in the guidance documentation. In that case the evaluator shall examine the guidance documentation instead.

The TOE is not distributed so this evaluation activity is not applicable.

2.6.7.2 Guidance Activities

The evaluator shall verify that the guidance documentation describes how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the guidance documentation shall describe how to query the loaded but inactive version.

Section 7.12 of [CCECG] ("Verify and Update System Software") provides instructions to query the currently active version of the TOE. The TOE does not provide a capability to install a trusted update with a delayed activation.

The evaluator shall verify that the guidance documentation describes how the verification of the authenticity of the update is performed (digital signature verification). The description shall include the procedures for successful and unsuccessful verification. The description shall correspond to the description in the TSS.

Section 7.12 of [CCECG] provides information on how the TOE verifies updates. It describes how the updates are digitally signed and verified prior to installation. It also states that if the verification fails, the TOE will not install the system updates. The guidance advises the administrator to ensure system updates are authentic by downloading the images from updates.paloaltonetworks.com only. The description corresponds to the description in the TSS.

For distributed TOEs the evaluator shall verify that the guidance documentation describes how the versions of individual TOE components are determined for FPT_TUD_EXT.1, how all TOE components are updated, and the error conditions that may arise from checking or applying the update (e.g. failure of signature verification, or exceeding available storage space) along with appropriate recovery actions. The guidance documentation only has to describe the procedures relevant for the Security Administrator; it does not need to give information about the internal communication that takes place when applying updates.

The TOE is not distributed so this evaluation activity is not applicable.

If this was information was not provided in the TSS: For distributed TOEs, the evaluator shall examine the Guidance Documentation to ensure that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when

© 2024 Leidos. All rights reserved.

applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component.

The TOE is not distributed so this evaluation activity is not applicable.

If this was information was not provided in the TSS: If the ST author indicates that a certificate-based mechanism is used for software update digital signature verification, the evaluator shall verify that the Guidance Documentation contains a description of how the certificates are contained on the device. The evaluator shall also ensure that the Guidance Documentation describes how the certificates are installed/updated/selected, if necessary.

The TOE update function does not use certificates in its digital signature verification process. This evaluation activity is not applicable.

2.6.7.3 Test Activities

Test 1: The evaluator shall perform the version verification activity to determine the current version of the product. If a trusted update can be installed on the TOE with a delayed activation, the evaluator shall also query the most recently installed version (for this test the TOE shall be in a state where these two versions match). The evaluator obtains a legitimate update using procedures described in the guidance documentation and verifies that it is successfully installed on the TOE. For some TOEs loading the update onto the TOE and activation of the update are separate steps ('activation' could be performed e.g. by a distinct activation step or by rebooting the device). In that case, the evaluator shall verify after loading the update onto the TOE but before activation of the update that the current version of the product did not change but the most recently installed version has changed to the new product version. After the update, the evaluator shall perform the version verification activity again to verify the version correctly corresponds to that of the update and that current version of the product and most recently installed version match again.

The evaluator checked the TOE's version, ran an update and then checked the version again. The TOE was verified to have been updated to the new version.

Test 2 [conditional]: If the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE the following test shall be performed (otherwise the test shall be omitted). The evaluator shall first confirm that no updates are pending and then perform the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator shall obtain or produce illegitimate updates as defined below and attempt to install them on the TOE. The evaluator shall verify that the TOE rejects all of the illegitimate updates. The evaluator shall perform this test using all of the following forms of illegitimate updates:

i. A modified version (e.g. using a hex editor) of a legitimately signed update
ii. An image that has not been signed
iii. An image signed with an invalid signature (e.g. by using a different key as expected for creating the signature or by manual modification of a legitimate signature)
iv. The handling of version information of the most recently installed version might differ between different TOEs depending on the point in time when an attempted update is rejected. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update

© 2024 Leidos. All rights reserved.

the evaluator shall verify that both the current version and most recently installed version, reflect the same version information as prior to the update attempt.

The evaluator first confirmed the current version of the TOE. The evaluator modified a byte in the signature section of a candidate update image and attempted to upload it onto the TOE. The TOE rejected the update image. The evaluator confirmed the TOE version was unchanged. The evaluator then attempted to upload an update image that had a missing signature. The TOE again rejected the update image. Lastly, the evaluator then attempted to upload an otherwise valid update image, with valid signature, that had one byte modified. The TOE rejected this image also.

The evaluator shall perform Test 1 and Test 2 for all methods supported (manual updates, automatic checking for updates, automatic updates).

The only update mechanism is manual update, so all testing above was performed using that mechanism.

For distributed TOEs the evaluator shall perform Test 1 and Test 2 for all TOE components.

The TOE is not distributed so multiple iterations of testing for separate TOE components is not applicable.

2.7 TOE Access (FTA)

2.7.1 FTA_SSL.3 TSF-initiated Termination

2.7.1.1 TSS Activities

The evaluator shall examine the TSS to determine that it details the administrative remote session termination and the related inactivity time period.

Section 6.7 of [ST] (“TOE Access”) describes the session termination behavior for remote sessions. The inactivity time period is a configurable value between 1 and 1,440 minutes.

2.7.1.2 Guidance Activities

The evaluator shall confirm that the guidance documentation includes instructions for configuring the inactivity time period for remote administrative session termination.

Section 7.6 of [CCECG] (“Configure Idle Timeout and Lockout”) states the administrator can configure an idle session timeout for both web UI and CLI users that access the TOE remotely. By default, the idle timeout value is 60 minutes, and the value can be configured from 1 to 1,440 minutes.

2.7.1.3 Test Activities

For each method of remote administration, the evaluator shall perform the following test:

Test 1: The evaluator shall follow the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator shall establish a remote interactive session with the TOE. The evaluator shall then observe that the session is terminated after the configured time period.

The evaluator configured the TOE to have different inactivity timeout values. The evaluator then authenticated to the TOE and verified when the inactivity value was hit the evaluator was logged out and had to re-authenticate to the TOE. This was performed for the HTTPS web GUI and SSH CLI.

© 2024 Leidos. All rights reserved.

2.7.2 FTA_SSL.4 User-initiated Termination

2.7.2.1 TSS Activities

The evaluator shall examine the TSS to determine that it details how the remote administrative session (and if applicable the local administrative session) are terminated.

Section 6.7 of [ST] ("TOE Access") states the TOE provides remote administrators the ability to logout (or terminate) their sessions. When connected via the Web interface, they can click on the "Logout" link, whilst when connected via the CLI, they can enter the "exit" command.

2.7.2.2 Guidance Activities

The evaluator shall confirm that the guidance documentation states how to terminate a remote interactive session (and if applicable the local administrative session).

Section 5.1.4 of [CCECG] ("User Logout") provides the guidance to terminate remote interactive sessions at the web GUI and the CLI and to terminate an API session.

2.7.2.3 Test Activities

Test 1 [conditional]: If the TOE supports local administration, the evaluator shall initiate an interactive local session with the TOE. The evaluator shall then follow the guidance documentation to exit or log off the session and observes that the session has been terminated.

The TOE in its evaluated configuration does not support local administration, so this test is not applicable.

Test 2: For each method of remote administration, the evaluator shall initiate an interactive remote session with the TOE. The evaluator shall then follow the guidance documentation to exit or log off the session and observes that the session has been terminated.

The evaluator performed a logout on the TOE via the HTTPS web GUI and SSH CLI and confirmed that the user was logged out and needed to re-authenticate to gain access to the TOE.

2.7.3 FTA_TAB.1 Default TOE Access Banners

2.7.3.1 TSS Activities

The evaluator shall check the TSS to ensure that it details each administrative method of access (local and/or remote) available to the Security Administrator (e.g. serial port, SSH, HTTPS). The evaluator shall check the TSS to ensure that all administrative methods of access available to the Security Administrator are listed and that the TSS states that the TOE is displaying an advisory notice and a consent warning message for each administrative method of access. The advisory notice and the consent warning message might be different for different administrative methods of access and might be configured during initial configuration (e.g. via configuration file).

Section 6.7 of [ST] ("TOE Access") states the TOE can be configured to display an informative banner that will appear prior to authentication when accessing the TOE via remote connection to the management port in order to access the Web Interface (HTTPS) or CLI (SSH). Since the banner is configurable by an administrator it can be used to display an advisory notice and consent warning.

© 2024 Leidos. All rights reserved.

2.7.3.2 Guidance Activities

The evaluator shall check the guidance documentation to ensure that it describes how to configure the banner message.

Section 7.5 of [CCECG] (“Configure Login Banner”) describes how to configure the banner message using the web GUI, the CLI, and the XML API.

2.7.3.3 Test Activities

Test 1: The evaluator shall follow the guidance documentation to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.

The evaluator configured the TOE access banner and confirmed it was displayed before authentication via the HTTPS web GUI and SSH CLI.

2.8 Trusted Path/Channels (FTP)

2.8.1 FTP_ITC.1 Inter-TSF Trusted Channel

2.8.1.1 TSS Activities

The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each secure communication mechanism is identified in terms of the allowed protocols for that IT entity, whether the TOE acts as a server or a client, and the method of assured identification of the non-TSF endpoint. The evaluator shall also confirm that all secure communication mechanisms are described in sufficient detail to allow the evaluator to match them to the cryptographic protocol Security Functional Requirements listed in the ST.

Section 6.8 of [ST] (“Trusted Path/Channels”) identifies the TOE’s communications with authorized IT entities as follows:

- With remote VPN gateways/peers using IPsec, as specified in FTP_ITC.1/VPN. The TOE is an IPsec peer.
- With GlobalProtect (VPN peer) using TLS, as specified in FTP_ITC.1. The TOE acts as the TLS server.
- With an external audit server using IPsec or TLS, as specified in FTP_ITC.1. The TOE acts as the TLS client.

Section 6.2 of [ST] (“Cryptographic Support”) describes how the TOE as TLS client identifies the external audit server using the X.509 certificate presented by the audit server during the TLS handshake. It also describes how the TOE can be configured as a TLS server for mutual certificate-based authentication for secure connections. Section 6.2 of [ST[also describes how the TOE identifies an IPsec peer using the presented identifier in the peer X.509 certificate.

The evaluator confirmed the TSS describes the secure communication mechanisms in sufficient detail to match them to the cryptographic protocol SFRs listed in Section 5.2.2 of [ST], specifically: FCS_TLSC_EXT.1; FCS_TLSS_EXT.1; and FCS_IPSEC_EXT.1.

© 2024 Leidos. All rights reserved.

2.8.1.2 Guidance Activities

The evaluator shall confirm that the guidance documentation contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken.

The TOE permits the following communications:

- Connecting with remote VPN gateways/peers using IPsec
- Connecting with Global Protect (VPN peer) using TLS
- Transmitting audit records to an audit server using IPsec or TLS.

Section 7.10 of [CCECG] ("Configure IKE/IPsec VPN Gateway") provides the instructions to configure the TOE as an IKE/IPsec VPN gateway to negotiate IKEv2 connections with its peers.

Section 7.2 of [CCECG] ("Configure Custom HTTPS or TLS Server Certificate") provides the instructions to configure TLS between the TOE and GlobalProtect.

Section 6.8.1 of [CCECG] ("Syslog Server Connection Settings (Required)") provides the instructions to configure a secure TLS channel or IPsec channel between the TOE and the external syslog server. The guidance provides Instructions for maintaining the physical connection between the TOE and the external syslog server should the connection be unintentionally broken.

2.8.1.3 Test Activities

The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report.

Test 1: The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.

The evaluation team set up each of the trusted channels as specified in the SFR, i.e.:

- IPsec connection with remote VPN peer
- TLS connection to external audit server
- TLS connection from GlobalProtect app.

The evaluation team confirmed each such connection was established successfully.

Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the guidance documentation to ensure that in fact the communication channel can be initiated from the TOE.

The requirement specifies the TOE initiates communication via the trusted channel for transmitting audit records to an external audit server. The TOE's ability to do this is demonstrated in testing of FAU_STG_EXT.1.

Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.

© 2024 Leidos. All rights reserved.

The evaluation team confirmed each such connection was established successfully and that all communicated data was protected.

Test 4: Objective: The objective of this test is to ensure that the TOE reacts appropriately to any connection outage or interruption of the route to the external IT entities.

The evaluator shall, for each instance where the TOE acts as a client utilizing a secure communication mechanism with a distinct IT entity, interrupt the connection of that IT entity for the following durations: i) a duration that exceeds the TOE's application layer timeout setting, ii) a duration shorter than the application layer timeout but of sufficient length to interrupt the network link layer.

The evaluator shall ensure that, when the connectivity is restored, communications are appropriately protected and no TSF data is sent in plaintext.

In the case where the TOE is able to detect TOE external interruption (such as a cable being physically removed or a virtual connection being disabled), another network device shall be used to interrupt the connection between the TOE and the distinct IT entity. The interruption shall be external to the TOE (i.e., by manipulating the test environment and not by TOE configuration change). In the case where the TOE is able to detect when the cable is removed from the device, another physical network device (e.g. a core switch) shall be used to interrupt the connection between the TOE and the distinct IT entity. The interruption shall not be performed at the virtual node (e.g. virtual switch) and must be physical in nature.

The evaluator had the TOE establish a connection to an external audit server. The evaluator then interrupted the network communication at an intermediate switch level for a short period, approximately 30-60 seconds. The evaluator observed that the TOE did not send data in plaintext and maintained protected communications when the connection was restored.

Next, the evaluator had the TOE establish a connection with an external audit server. The evaluator then interrupted the network communication at an intermediate switch level for a long period, approximately 30 minutes. The evaluator observed that the TOE did not send data in plaintext and initiated a new TLS session when connectivity was restored.

Further assurance activities are associated with the specific protocols.

For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of external secure channels to TOE components in the Security Target.

The TOE is not distributed so this evaluation activity is not applicable.

2.8.2 FTP_ITC.1/VPN Inter-TSF Trusted Channel (VPN Communications) (VPNGW-SD)

2.8.2.1 TSS Activities

The EAs specified for FTP_ITC.1 in the Supporting Document for the Base-PP shall be applied for IPsec VPN communications.

This evaluation activity is performed in combination with FTP_ITC.1 as suggested - see Section 2.8.1.1**Error! Reference source not found.**.

2.8.2.2 Guidance Activities

The EAs specified for FTP_ITC.1 in the Supporting Document for the Base-PP shall be applied for IPsec VPN communications.

© 2024 Leidos. All rights reserved.

This evaluation activity is performed in combination with FTP_ITC.1 as suggested - see Section 2.8.1.2.

2.8.2.3 Test Activities

The EAs specified for FTP_ITC.1 in the Supporting Document for the Base-PP shall be applied for IPsec VPN communications. Additional evaluation testing for IPsec is covered in FCS_IPSEC_EXT.1.

This evaluation activity is performed in combination with FTP_ITC.1 as suggested - see Section 2.8.1.3.

2.8.3 FTP_TRP.1/Admin Trusted Path

2.8.3.1 TSS Activities

The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.

Section 6.5 of [ST] ("Security Management) states the TOE provides a GUI, API management interface, and CLI to support security management of the TOE. An administrator accesses the GUI and API via HTTPS or HTTPS tunneled over IPsec and accesses the CLI via SSHv2. Section 6.8 of [ST] ("Trusted Path/Channels") states the TOE provides SSH and HTTPS to support secure remote authentication. In addition, HTTPS can be tunneled through IPsec. All remote security management functions require the use of a secure channel. The TOE's evaluated configuration permanently disables telnet and HTTP.

The protocols specified in the requirement (FTP_TRP.1.1/Admin) are SSH, HTTPS, and IPsec, which is consistent with the claims made in the TSS. The TOE also claims FCS_SSH_EXT.1, FCS_SSHS_EXT.1, FCS_HTTPS_EXT.1 (along with its dependency FCS_TLSS_EXT.1), and FCS_IPSEC_EXT.1. Therefore, the methods of TOE administration are consistent between the SFRs and TSS.

2.8.3.2 Guidance Activities

The evaluator shall confirm that the guidance documentation contains instructions for establishing the remote administrative sessions for each supported method.

The TOE uses HTTPS for the web UI and API and SSH for the CLI to secure the remote administrative sessions. In addition, the TOE supports tunneling of HTTPS management sessions over IPsec. Telnet and HTTP are disabled by default and must not be enabled in the evaluated configuration.

Section 5.1.1 of [CCECG] ("User Login to Web Interface") provides the instructions for a remote administrator to login to the TOE web UI via an HTTPS connection with username and password.

Section 6.8.2 of [CCECG] ("Certificate-Based Authentication for Web UI (Optional)") provides instructions to use certificate-based authentication for remote Web UI administrator accounts that are local to the TOE. Certificate-based authentication involves the exchange and verification of a digital signature instead of a password. Configuring certificate-based authentication for any administrator disables the username/password logins for all administrators on the TOE and all administrators thereafter require the certificate to log in. The [CCECG] describes login using a certificate following the configuration instructions (steps 11 – 15).

Section 5.1.2 of [CCECG] ("User Login to CLI Remotely") provides the instructions to remotely log into the TOE CLI via an SSHv2 connection, while section 6.6 of [CCECG] ("Configure SSH Public-Key Authentication

© 2024 Leidos. All rights reserved.

(Recommended)") provides the instructions for a remote user to login to the CLI via SSH public key authentication.

Section 7.14 of [CCECG] ("XML and REST API") provides the instructions for a remote administrator to use the API for management of the TOE. The administrator uses their username and password to generate an API key used to authenticate API calls. The instructions include a description of the parameters and example calls for both types of API: XML and REST. Additionally, throughout the [CCECG], there are examples of how to perform each management function required by the PP and specified in the ST using the management interfaces, including the API.

2.8.3.3 Test Activities

Test 1: The evaluators shall ensure that communications using each specified (in the guidance documentation) remote administration method is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.

The TOE supports remote administration via:

- SSH—Evidence for the use of this method can be seen in the testing for FMT_SMR.2 in the CLI test case. In that test, the evaluator connected to the TOE via SSH and was successful in utilizing the interface.
- HTTPS—Evidence for the use of this method can be seen in the testing for FPT_STM_EXT.1. In that testing, the evaluator connected to the TOE over the HTTPS interface in order to set the system time and was successful in connecting to it.
- IPsec—Evidence for the use of this method can be seen in the evidence for FCS_IPSEC_EXT.1.1, where the TOE is shown to be able to enforce filtering rules for IPsec traffic. As the purpose of the IPsec claim is to allow HTTPS traffic to be tunneled over IPsec, this has been deemed to be sufficient by the evaluator.

Test 2: The evaluator shall ensure, for each communication channel, the channel data is not sent in plaintext.

Testing for FCS_IPSEC_EXT.1 provides evidence the TOE correctly implements the IPsec protocol. As the IPsec protocol when using ESP tunneling encrypts the traffic this is sufficient to show that data sent over an IPsec ESP tunnel is not sent in plaintext.

For SSH, the evaluator established an SSH connection with the TOE while capturing packets. The evaluator verified that the TOE sent the packets in an encrypted manner as required by the SSH protocol.

For HTTPS, the evaluator initiated an HTTPS session with the TOE while capturing packets. The evaluator successfully logged on to the TOE over HTTPS. The evaluator reviewed the packet capture and observed that no data was sent in plaintext.

Further assurance activities are associated with the specific protocols.

For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of trusted paths to TOE components in the Security Target.

This test is not applicable because the TOE is not distributed.

© 2024 Leidos. All rights reserved.

2.9 Firewall (FFW) (FW-SD)

2.9.1 FFW_RUL_EXT.1 Stateful Traffic Filtering

2.9.1.1 TSS Activities

FFW_RUL_EXT.1.1
The evaluator shall verify that the TSS provides a description of the TOE's initialization/startup process, which clearly indicates where processing of network packets begins to take place, and provides a discussion that supports the assertion that packets cannot flow during this process.

Section 6.9 of [ST] ("Stateful Traffic Filtering") states the TOE runs a series of system checks and the FIPS power up self- tests during start-up and initialization, to ensure the system is functioning correctly. If these tests run successfully, the TOE will bring up the control plane and data plane system modules. The Policy Enforcement Module (running on the data plane) uses the policy configuration information created from the Management Server Module (running on the control plane). The configuration information includes all the policies required by the Policy Enforcement Module. Policies are used to control information flow on the network. The TOE can pass traffic only after the Policy Enforcement Module is executing on the data plane and the TOE's system configuration is applied to enforce all security policies.

The evaluator shall verify that the TSS also include a narrative that identifies the components (e.g., active entity such as a process or task) involved in processing the network packets and describe the safeguards that would prevent packets flowing through the TOE without applying the ruleset in the event of a component failure. This could include the failure of a component, such as a process being terminated, or a failure within a component, such as memory buffers full and cannot process packets. The description shall also include a description how the TOE behaves in the situation where the traffic exceeds the amount of traffic the TOE can handle and how it is ensured that also in this condition stateful traffic filtering rules are still applied so that traffic does not pass that shouldn't pass according to the specified rules.

Section 6.9 of [ST] states the TOE implements the following safeguards that prevent packets from flowing through the TOE without applying the ruleset in the event of a component failure. The traffic can go through the TOE only if the Policy Enforcement Module is fully functional and enforcing all policies. The Policy Enforcement Module can be configured to stop traffic when the traffic or system logs are full. During start-up and initialization, the TOE runs a series of system checks and the FIPS power up self- tests to ensure the system is functioning correctly. If these tests run successfully, the TOE will bring up the control plane and data-plane system modules. The Policy Enforcement Module (running on Data Plane) uses the policy configuration information created from the Management Server Module (running on the control plane). The configuration information includes all policies required by the Policy Enforcement Module. Policies are used to control information flow on the network. Only once the Policy Enforcement Module running on the data-plane is up and running and the TOE's system configuration is applied to enforce all security policies, can the TOE pass the traffic. Whenever a failure occurs within the TOE that results in the TOE ceasing operation, the TOE securely disables its interfaces to prevent the unintentional flow of any information to or from the TOE and reloads.

© 2024 Leidos. All rights reserved.

FFW_RUL_EXT.1.2/FFW_RUL_EXT.1.3/FFW_RUL_EXT.1.4

The evaluator shall verify that the TSS describes a stateful packet filtering policy and the following attributes are identified as being configurable within stateful traffic filtering rules for the associated protocols:

- ICMPv4
  - Type
  - Code
- ICMPv6
  - Type
  - Code
- IPv4
  - Source address
  - Destination Address
  - Transport Layer Protocol
- IPv6
  - Source address
  - Destination Address
  - Transport Layer Protocol and where defined by the ST author, Extension Header Type, Extension Header Fields
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

Section 6.9 of [ST] describes the required attributes as being configurable within stateful traffic filtering rules for the associated protocols.

FFW_RUL_EXT.1.2/FFW_RUL_EXT.1.3/FFW_RUL_EXT.1.4

The evaluator shall verify that each rule can identify the following actions: permit or drop with the option to log the operation. The evaluator shall verify that the TSS identifies all interface types subject to the stateful packet filtering policy and explains how rules are associated with distinct network interfaces.

Section 6.9 of [ST] describes how rules can be configured to permit or drop the traffic. The rule also includes a Log Setting option to determine whether a log should be generated when the rule is triggered. Rules can be configured for distinct network interfaces and zones. Separate zones must be created for each type of interface (Layer 2, Layer 3, or virtual wire). All TOE interfaces assigned to a zone are subject to the policy.

FFW_RUL_EXT.1.5

The evaluator shall verify that the TSS identifies the protocols that support stateful session handling. The TSS shall identify TCP, UDP, and, if selected by the ST author, also ICMP.

© 2024 Leidos. All rights reserved.

Section 6.9 of [ST] states that TCP, UDP, and ICMP protocols are supported for stateful session handling.

The evaluator shall verify that the TSS describes how stateful sessions are established (including handshake processing) and maintained.

Section 6.9 of [ST] states the TOE enforces the stateful traffic filtering rules based on subject and information security attributes:

- Source security zone to which the physical network interface is assigned
- Destination security zone to which the network interface is assigned
- Information specifiable in security policies, which provide the information flow rule sets:
  - presumed identity of source subject—source address information within the packet
  - identity of destination subject—destination address information within the packet
  - transport layer protocol (e.g., TCP, UDP)
  - Internet layer protocol (e.g., ICMP type, code)
  - source subject service identifier (e.g., source port number)
  - destination subject service identifier (e.g., destination port number)
- Information security attributes for stateful packet inspection—for connection-oriented protocols (e.g., TCP), the sequence number, acknowledgement number, and flags (SYN, ACK, RST, FIN); and for connectionless protocols (e.g., UDP), the source and destination network identifiers; and source and destination service identifiers. Note that the TOE uses an IP-based network stack.

Section 6.9 of [ST] describes the traffic handling policy. When the TOE receives a packet, it first determines if it represents a new connection or if it is part of an existing session. If it is part of an existing session, the traffic is processed based on the parameters of the existing session. If it is a new connection, the TOE retrieves the source and destination zones and performs an initial policy lookup. If a policy is defined for the zone pair (i.e., source and destination zones) a session is created and packet processing proceeds. By default, traffic between each pair of security zones is blocked until at least one rule is added to allow traffic between the two zones. Sessions are not created for a new connection if there is no policy defined for the zone pair; or if half-open is an initial deny rule for the application service (i.e., service-HTTP, service-https) matching the traffic with no applications defined.

The TOE performs the following steps when processing traffic:

- The traffic is passed through the Application Identification and Application Decoders to determine what type of application is creating the session.
- Once the application is known, the TOE performs a policy lookup with the following information:
  - The source/destination IP address
  - The source/destination security zone
  - The application and service (port and protocol, Next Header)
  - The source user, when available (the source user in policies is not within the scope of the evaluation).
- If a security policy is found, the policy rules are compared against the incoming traffic in sequence and the first rule that matches the traffic is applied. If a policy rule matching all of the traffic attributes listed above is not found, or if it is found and it specifies a drop action, then the packet is dropped (or DISCARDed) and the session is deleted.
- If the application flow is allowed and no further security profiles are applied then it is forwarded (it is allowed to BYPASS the tunnel).

© 2024 Leidos. All rights reserved.

- If the application is allowed and there are additional security profiles set, it will be sent to the stream signature processor. The traffic matching the IPsec crypto Security profile would then flow through the IPsec tunnel and be classified as “PROTECTED”.
    - If there is no SA that the IPsec can use to protect this traffic to the peer, IPsec uses IKE to negotiate with the remote peer to set up the necessary IPsec SAs on behalf of the data flow. The negotiation uses information specified in the IKE Network Profiles.

Security policies can also specify security profiles that may be used to protect against viruses, spyware, and other threats after the connection is established.

Section 6.9 of [ST] goes on to say that the first rule that matches the observed traffic is applied, and that traffic for which there is no matching rule is dropped by default. If the rule allows the traffic with no additional security policies, it is forwarded. If the rule allows the traffic but additional security profiles are set, it is forwarded to the stream signature processor.

The evaluator shall verify that for TCP, the TSS identifies and describes the use of the following attributes in session determination: source and destination addresses, source and destination ports, sequence number, and individual flags.

Section 6.9 of [ST] identifies for TCP the following security attributes against which policy is enforced:

1. presumed identity of source subject (address)
2. identity of destination subject (address)
3. source service identifier (port)
4. destination service identifier (port)
5. transport layer protocol (TCP in this case)
6. internet layer protocol (N/A in this case)
7. information security attributes for stateful inspection – [ST] gives examples of this for TCP as the sequence number, acknowledgement number, and flags (e.g. SYN, ACK, RST, FIN)

The evaluator shall verify that for UDP, the TSS identifies and describes the following attributes in session determination: source and destination addresses, source and destination ports.

Section 6.9 f [ST] identifies for UDP the following security attributes against which policy is enforced:

1. presumed identity of source subject (address)
2. identity of destination subject (address)
3. source service identifier (port)
4. destination service identifier (port)
5. transport layer protocol (UDP in this case)
6. internet layer protocol (N/A in this case)
7. information security attributes for stateful inspection – [ST] gives examples of this for UDP as the source and destination service identifiers (i.e. ports)

The evaluator shall verify that for ICMP (if selected), the TSS identifies and describes the following attributes in session determination: source and destination addresses, other attributes chosen in FFW_RUL_EXT.1.5.

© 2024 Leidos. All rights reserved.

Section 6.9 of [ST] identifies for ICMP the following security attributes against which policy is enforced:

1. presumed identity of source subject (address)
2. identity of destination subject (address)
3. source service identifier (port)
4. destination service identifier (port)
5. transport layer protocol (N/A in this case)
6. internet layer protocol (ICMP in this case, including type and code)
7. information security attributes for stateful inspection (N/A in this case)

The evaluator shall verify that the TSS describes how established stateful sessions are removed. The TSS shall describe how connections are removed for each protocol based on normal completion and/or timeout conditions. The TSS shall also indicate when session removal becomes effective (e.g., before the next packet that might match the session is processed).

Section 6.9 of [ST] states traffic flows are removed from the set of existing flows based on session inactivity timeout or completion of the expected information flow. Session timeout is based on an administrator-configurable value from 1-6,044,800 seconds. The TSS also states this behavior applies to all supported protocols.

FFW_RUL_EXT.1.6

The evaluator shall verify that the TSS identifies the following as packets that will be automatically dropped and are counted or logged:

a) Packets which are invalid fragments, including a description of what constitutes an invalid fragment
b) Fragments that cannot be completely re-assembled
c) Packets where the source address is defined as being on a broadcast network
d) Packets where the source address is defined as being on a multicast network
e) Packets where the source address is defined as being a loopback address
f) The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address “reserved for future use” (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;
g) The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as an “unspecified address” or an address “reserved for future definition and use” (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;
h) Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified
i) Other packets defined in FFW_RUL_EXT.1.6 (if any)

Section 6.9 of [ST] identifies the required packets and additional types that are automatically dropped and are counted or logged (a-h).

The additional types defined in the SFR (i) and listed in the TSS are for the TOE’s capability to block certain types of IPv6 traffic:

- block both inbound and outbound IPv6 Site Local Unicast addresses (FEC0::/10)
- block IPv6 Jumbo Payload datagrams (Option Type 194).

© 2024 Leidos. All rights reserved.

- drop all inbound and outbound IPv6 packets containing a Hop-by-Hop header with option type values intended for Destination Options
- drop all inbound IPv6 packets for which the layer 4 protocol and ports (undetermined transport) cannot be located.
- drop all inbound IPv6 packets with a Type 0 Routing header.
- drop all inbound IPv6 packets with a Type 1 or Types 3 through 255 Routing Header.
- drop all inbound IPv6 packets containing undefined header extensions/protocol values.
- drop fragmented IPv6 packets when any fragment overlaps another.
- drop all inbound IPv6 packets containing more than one Fragmentation Header within an IP header chain.
- drop all inbound and outbound IPv6 packets containing a Hop-by-Hop header with option type values intended for Destination Options.
- block IPv6 multicast addresses (FF00::/8) as a source address.
- blocks RFC 6598 “Carrier Grade NAT” IP address block of 100.64.0.0/10.

FFW_RUL_EXT.1.7

The evaluator shall verify that the TSS explains how the following traffic can be dropped and counted or logged:

a) Packets where the source address is equal to the address of the network interface where the network packet was received

b) Packets where the source or destination address of the network packet is a link-local address

c) Packets where the source address does not belong to the networks associated with the network interface where the network packet was received, including a description of how the TOE determines whether a source address belongs to a network associated with a given network interface

Section 6.9 of [ST] states the administrator can configure the TOE to reject and log network packets where the source or destination address of the network packet is defined as a link-local address. Specifically, the TOE rejects packets where the source address is equal to the address of the network interface where the network packet was received.

The TOE also rejects requests when received on an interface that is not associated with the source address from which the information flow is sourced.

FFW_RUL_EXT.1.8

The evaluator shall verify that the TSS describes the algorithm applied to incoming packets, including the processing of default rules, determination of whether a packet is part of an established session, and application of administrator defined and ordered ruleset.

Section 6.9 of [ST] states the TOE uses Security Zones and Security Policies to determine whether to block or allow packets based on attributes such as the source and destination security zone, the source and destination IP address, the application, user, and the service. The default rules are defined and are part of the security policies applied.

When the TOE receives a packet, it first determines if it represents a new connection or if it is part of an existing session. If it is part of an existing session, the traffic is processed based on the parameters of the existing session. If it is a new connection, the TOE retrieves the source and destination zones and performs an initial policy lookup. If a policy is defined for the zone pair (i.e., source and destination zones) a session is created and packet processing proceeds. By default, traffic between each pair of security zones is

© 2024 Leidos. All rights reserved.

blocked until at least one rule is added to allow traffic between the two zones. Sessions are not created for a new connection if there is no policy defined for the zone pair; or if there is an initial deny rule for the application service (i.e. service-HTTP, service-https) matching the traffic with no applications defined.

Administrator-defined rulesets become part of the Security Policy. Security policies are evaluated left to right and from top to bottom in a packet filtering table format. A packet is matched against the first rule that meets the defined criteria; after a match is triggered the subsequent rules are not evaluated. Therefore, the more specific rules must precede more generic ones in order to enforce the best match criteria. Traffic that matches a rule generates a log entry at the end of the session in the traffic log, if logging is enabled for that rule.

Added per TD0545.

FFW_RUL_EXT.1.8
If the TOE implements a mechanism that ensures that no conflicting rules can be configured, the TSS shall describe the underlying mechanism.

Section 6.9 of [ST] states that rules are processed in hierarchical order so the first match is always processed. This prevents potential conflicting rules since the hierarchy always enforces an order of precedence.

FFW_RUL_EXT.1.9
The evaluator shall verify that the TSS describes the process for applying stateful traffic filtering rules and also that the behavior (either by default, or as configured by the administrator) is to deny packets when there is no rule match unless another required conditions allows the network traffic (i.e., FFW_RUL_EXT.1.5 or FFW_RUL_EXT.2.1).

Section 6.9 of [ST] states that all traffic is matched against a session and that each session is matched against a security policy. Security policies are evaluated left to right and top to bottom, with the first applicable match being enforced (so more specific rules should precede more generic ones). This section also states that the TOE can be configured to deny all traffic for which there is no rule match.

FFW_RUL_EXT.1.10
The evaluator shall verify that the TSS describes how the TOE tracks and maintains information relating to the number of half-open TCP connections. The TSS should identify how the TOE behaves when the administratively defined limit is reached and should describe under what circumstances stale half-open connections are removed (e.g. after a timer expires).

Section 6.9 of [ST] states the number of half-open TCP connections can be limited administratively, as can the thresholds that constitute flooding. This section also states the Security Administrator can specify the thresholds at which the firewall generates a DoS alarm. The TOE takes action such as Random Early Drop, drop additional incoming connections, and the dropped connections are logged, if configured to do so.

2.9.1.2 Guidance Activities

FFW_RUL_EXT.1.1
The guidance documentation associated with this requirement is assessed in the subsequent test evaluation activities.

© 2024 Leidos. All rights reserved.

FFW_RUL_EXT.1.2/FFW_RUL_EXT.1.3/FFW_RUL_EXT.1.4

The evaluators shall verify that the guidance documentation identifies the following attributes as being configurable within stateful traffic filtering rules for the associated protocols:

- ICMPv4
  - Type
  - Code
- ICMPv6
  - Type
  - Code
- IPv4
  - Source address
  - Destination Address
  - Transport Layer Protocol
- IPv6
  - Source address
  - Destination Address
  - Transport Layer Protocol and where defined by the ST author, Extension Header Type, Extension Header Fields
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

Section 7.9 of [CCECG] (“Configure Stateful Inspection Filtering”) identifies the protocols and attributes above as being configurable within stateful traffic filtering rules.

FFW_RUL_EXT.1.2/FFW_RUL_EXT.1.3/FFW_RUL_EXT.1.4

The evaluator shall verify that the guidance documentation indicates that each rule can identify the following actions: permit, drop, and log.

Section 7.9 of [CCECG] states that each rule can be configured with an “allow” (i.e., permit) or “deny” (i.e., drop) action and a logging option.

FFW_RUL_EXT.1.2/FFW_RUL_EXT.1.3/FFW_RUL_EXT.1.4

The evaluator shall verify that the guidance documentation explains how rules are associated with distinct network interfaces.

Section 7.9 of [CCECG] describes how rules are associated with distinct network interfaces by assigning interfaces to zones and defining rules for zones.

© 2024 Leidos. All rights reserved.

FFW_RUL_EXT.1.5
The evaluator shall verify that the guidance documentation describes stateful session behaviours. For example, a TOE might not log packets that are permitted as part of an existing session.

Section 7.9 of [CCECG] states security policy rules are used to determine whether to block or allow a session based on traffic attributes such as the source and destination security zone, the source and destination IP address, the application, user, and the service. When an administrator creates a security policy rule, the administrator can specify if the TOE will log traffic matching the rule.

Stateful session behaviors described in the ST are included along with any configuration options. For example: descriptions and procedures are provided for removing existing traffic flows from the set of established traffic flows based on session inactivity timeout, completion of the expected information flow (as described in [ST] FFW_RUL_EXT.1.5 b)).

FFW_RUL_EXT.1.6
The evaluator shall verify that the guidance documentation describes packets that are discarded and potentially logged by default. If applicable protocols are identified, their descriptions need to be consistent with the TSS. If logging is configurable, the evaluator shall verify that applicable instructions are provided to configure auditing of automatically rejected packets.

Section 7.9 of [CCECG] describes the default Stateful Traffic Filtering rules defined in FFW_RUL_EXT.1.6, including packets that are discarded and logged by default and rules applicable to IPv6 packets. The guidance also provides instructions to configure auditing of automatically rejected packets that are not logged by default.

FFW_RUL_EXT.1.7
The evaluator shall verify that the guidance documentation describes how the TOE can be configured to implement the required rules. If logging is configurable, the evaluator shall verify that applicable instructions are provided to configure auditing of automatically rejected packets.

Section 7.9 of [CCECG] provides instructions to configure the security policy rules to determine whether to block or allow a session based on traffic attributes such as the source and destination security zone, the source and destination IP address, the application, user and service, and to specify if the TOE will log traffic matching the rule.

FFW_RUL_EXT.1.8
The evaluator shall verify that the guidance documentation describes how the order of stateful traffic filtering rules is determined and provides the necessary instructions so that an administrator can configure the order of rule processing.

Section 7.9 of [CCECG] states that security policies are evaluated left to right and from top to bottom. A packet is matched against the first rule that meets the defined criteria; after a match is triggered the subsequent rules are not evaluated. Therefore, the more specific rules must precede more generic ones in order to enforce the best match criteria. This section also provides an example of configuring security policies that define how stateful traffic filtering is applied to network packets received by the firewall based on traffic attributes: source and destination security zone; the source and destination IP address; the application; user; and the service.

© 2024 Leidos. All rights reserved.

FFW_RUL_EXT.1.9

The evaluator shall verify that the guidance documentation describes the behavior if no rules or special conditions apply to the network traffic. If the behavior is configurable, the evaluator shall verify that the guidance documentation provides the appropriate instructions to configure the behavior to deny packets with no matching rules.

Section 7.9 of [CCECG] states that for traffic that doesn't match any defined rules, the default rules apply. The default rules are predefined to allow all intrazone (within the zone) traffic and deny all interzone (between zones) traffic.

FFW_RUL_EXT.1.10

The evaluator shall verify that the guidance documentation describes the behaviour of imposing TCP half-open connection limits and its default state if unconfigured. The evaluator shall verify that the guidance clearly indicates the conditions under which new connections will be dropped e.g. per-destination or per-client.

Sections 7.9.1 (per zone) and 7.9.2 (per destination) of [CCECG] describe the behavior of imposing TCP half-open connection limits and procedures to configure. The zone protection profile is applied to the zone (e.g., trust or untrust zones). The DoS protection profile is tied to a security rule and will be based on the destination address defined in that DoS rule.

2.9.1.3 Test Activities

The following table provides an overview about execution of test cases regarding IPv4 and IPv6.

SFR Element/Test Case	Test execution
FFW_RUL_EXT.1, Tests 1-2	Both, IPv4 and IPv6.
FFW_RUL_EXT.1.2/1.3/1.4, Tests 1-2	As defined in the test description.
FFW_RUL_EXT.1.5, Tests 1-8	Both, IPv4 and IPv6.
FFW_RUL_EXT.1.6, Tests 1-2	Both IPv4 and IPv6 shall be tested for items a), b), c), d), and e) of the SFR element FFW_RUL_EXT.1.6. Both IPv4 and IPv6 shall be tested for item i) unless the rule definition is specific to IPv4 or IPv6. Note: f), g), and h) are specific to IPv4 or IPv6 and shall be tested accordingly.
FFW_RUL_EXT.1.7, Tests 1-2	Both, IPv4 and IPv6.
FFW_RUL_EXT.1.8, Tests 1-2	Both, IPv4 and IPv6.
FFW_RUL_EXT.1.9, Test 1	As defined in the test description.
FFW_RUL_EXT.1.10, Tests 1	Both, IPv4 and IPv6.

FFW_RUL_EXT.1.1

Test 1: The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would otherwise be denied by the ruleset should be

© 2024 Leidos. All rights reserved.

sourced and be directed at a host. The evaluator shall verify using a packet sniffer that none of the generated network traffic is permitted through the firewall during initialization.

The evaluator caused the TOE to reboot and directed traffic at the TOE that should be denied by its ruleset. The evaluator verified that the TOE did not permit the traffic to pass while being initialized.

Test 2: The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would be permitted by the ruleset should be sourced and be directed at a host. The evaluator shall verify using a packet sniffer that none of the generated network traffic is permitted through the firewall during initialization and is only permitted once initialization is complete.

The evaluator caused the TOE to reboot and directed traffic at the TOE that should be permitted by its ruleset. The evaluator verified that the TOE did not permit the traffic to pass while being initialized, but once initialization was complete the traffic was passed.

Note: The remaining testing associated with application of the ruleset is addressed in the subsequent test evaluation activities.

FFW_RUL_EXT.1.2/FFW_RUL_EXT.1.3/FFW_RUL_EXT.1.4

Test 1: The evaluator shall use the instructions in the guidance documentation to test that stateful packet filter firewall rules can be created that permit, drop, and log packets for each of the following attributes:

- ICMPv4
  - Type
  - Code
- ICMPv6
  - Type
  - Code
- IPv4
  - Source address
  - Destination Address
  - Transport Layer Protocol
- IPv6
  - Source address
  - Destination Address
  - Transport Layer Protocol and where defined by the ST author, Extension Header Type, Extension Header Fields
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port

© 2024 Leidos. All rights reserved.

- Destination Port

The evaluator used the guidance documentation to configure stateful packet filter rules for each of the attributes specified in the requirement. A 'permit' rule and a 'deny' rule were created for different values of each attribute. The evaluator sent packets with values of each tested attribute that matched either the 'permit' rule or the 'deny' rule. The evaluator confirmed via TOE logs and packet captures that traffic was correctly permitted or denied based on the applicable rules.

Test 2: Repeat the test evaluation activity above to ensure that stateful traffic filtering rules can be defined for each distinct network interface type supported by the TOE.

The evaluator observed that regardless of the physical interface type used by the TOE, the logical interface is always configured as Ethernet. There are no other types of logical interfaces for which stateful traffic filtering rules can be defined.

Note that these Test Activities should be performed in conjunction with those of FFW RUL_EXT.1.9 where the effectiveness of the rules is tested. The Test Activities for FFW_RUL_EXT.1.9 define the protocol/attribute combinations required to be tested. If those combinations are configured manually, that will fulfil the objective of these Test Activities, but if those combinations are configured otherwise (e.g., using automation), these Test Activities may be necessary in order to ensure the guidance is correct and the full range of configurations can be achieved by a TOE administrator.

The evaluator configured the rules manually, so this testing was addressed through the testing for FFW_RUL_EXT.1.9.

FFW_RUL_EXT.1.5

The following tests shall be run using IPv4 and IPv6.

Test 1: The evaluator shall configure the TOE to permit and log TCP traffic. The evaluator shall initiate a TCP session. While the TCP session is being established, the evaluator shall introduce session establishment packets with incorrect flags to determine that the altered traffic is not accepted as part of the session (i.e., a log event is generated to show the ruleset was applied). After a TCP session is successfully established, the evaluator shall alter each of the session determining attributes (source and destination addresses, source and destination ports, sequence number, flags) one at a time in order to verify that the altered packets are not accepted as part of the established session.

The evaluator established a TCP session across the TOE. The evaluator then directed TCP packets modified in the prescribed manner at the TOE. The evaluator verified that the TOE did not accept the modified TCP packets as part of the established session, but instead either dropped the packets or accepted them as a new session, depending on the validity of the packets to start a session.

Test 2: The evaluator shall terminate the TCP session established per Test 1 as described in the TSS. The evaluator shall then immediately send a packet matching the former session definition in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

The evaluator terminated the TCP session. The evaluator then sent packets that matched the previous session across the TOE and observed that they were subject to the ruleset.

© 2024 Leidos. All rights reserved.

Test 3: The evaluator shall expire (i.e., reach timeout) the TCP session established per Test 1 as described in the TSS. The evaluator shall then send a packet matching the former session in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

The evaluator reached the timeout of the TCP session. The evaluator then sent packets that matched the timed-out session across the TOE and observed that they were subject to the ruleset.

Test 4: The evaluator shall configure the TOE to permit and log UDP traffic. The evaluator shall establish a UDP session. Once a UDP session is established, the evaluator shall alter each of the session determining attributes (source and destination addresses, source and destination ports) one at a time in order to verify that the altered packets are not accepted as part of the established session.

The evaluator established a UDP session across the TOE. The evaluator then directed UDP packets modified in the prescribed manner at the TOE. The evaluator verified that the TOE did not accept the modified UDP packets as part of the established session, but instead either dropped the packets, or accepted them as a new session.

Test 5: The evaluator shall expire (i.e., reach timeout) the UDP session established per Test 4 as described in the TSS. The evaluator shall then send a packet matching the former session in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

The evaluator reached the timeout of the UDP session. The evaluator then sent packets that matched the timed-out session across the TOE and observed that they were subject to the ruleset.

Test 6: If ICMP is selected, the evaluator shall configure the TOE to permit and log ICMP traffic. The evaluator shall establish a session for ICMP as defined in the TSS. Once an ICMP session is established, the evaluator shall alter each of the session determining attributes (source and destination addresses, other attributes chosen in FFW_RUL_EXT.1.5) one at a time in order to verify that the altered packets are not accepted as part of the established session.

The evaluator established an ICMP session across the TOE. The evaluator then directed ICMP packets modified in the prescribed manner at the TOE. The evaluator verified that the TOE did not accept the modified ICMP packets as part of the established session, but instead either dropped the packets, or accepted them as a new session.

Test 7: If applicable, the evaluator shall terminate the ICMP session established per Test 6 as described in the TSS. The evaluator shall then immediately send a packet matching the former session definition in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

The TOE does not identify any method to terminate an ICMP session.

Test 8: The evaluator shall expire (i.e., reach timeout) the ICMP session established per Test 6 as described in the TSS. The evaluator shall then send a packet matching the former session in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

The evaluator reached the timeout of the ICMP session. The evaluator then sent packets that matched the timed-out session across the TOE and observed that they were subject to the ruleset.

© 2024 Leidos. All rights reserved.

FFW_RUL_EXT.1.6

Both IPv4 and IPv6 shall be tested for items a), b), c), d), and e) of the SFR element. Both IPv4 and IPv6 shall be tested for item i) unless the rule definition is specific to IPv4 or IPv6. Note: f), g), and h) are specific to IPv4 or IPv6 and shall be tested accordingly.

Test 1: The evaluator shall test each of the conditions for automatic packet rejection in turn. In each case, the TOE should be configured to allow all network traffic and the evaluator shall generate a packet or packet fragment that is to be rejected. The evaluator shall use packet captures to ensure that the unallowable packet or packet fragment is not passed through the TOE.

The evaluator directed packets at the TOE that met each of the possible automatic packet rejection rules one at a time. The evaluator verified that the TOE rejected the packets and that they were not transmitted across the TOE.

Test 2: For each of the cases above, the evaluator shall use any applicable guidance to enable dropped packet logging or counting. In each case above, the evaluator shall ensure that the rejected packet or packet fragment was recorded (either logged or an appropriate counter incremented).

The evaluator followed guidance to enable dropped packet logging. The evaluator examined the logs and verified that the dropped packets were logged.

FFW_RUL_EXT.1.7

The following tests shall be run using IPv4 and IPv6.

Test 1: The evaluator shall configure the TOE to drop and log network traffic where the source address of the packet matches that of the TOE network interface upon which the traffic was received. The evaluator shall generate suitable network traffic to match the configured rule and verify that the traffic is dropped and a log message generated.

The evaluator directed packets at the TOE that matched the source address of the TOE interface the packets were received on and verified that the TOE dropped the packets and logged the packets.

Test 2: The evaluator shall configure the TOE to drop and log network traffic where the source IP address of the packet fails to match the network reachability information of the interface to which it is targeted, e.g. if the TOE believes that network 192.168.1.0/24 is reachable through interface 2, network traffic with a source address from the 192.168.1.0/24 network should be generated and sent to an interface other than interface 2. The evaluator shall verify that the network traffic is dropped and a log message generated.

The evaluator directed packets at the TOE that did not match the subnet address served by TOE interface the packets were received on and verified that the TOE dropped the packets and logged the packets.

Modified per TD0545.

FFW_RUL_EXT.1.8

Test 1: If the TOE implements a mechanism that ensures that no conflicting rules can be configured, the evaluator shall try to configure two conflicting rules and verify that the TOE rejects the conflicting rule(s). It is important to verify that the mechanism is implemented in the TOE but not in the non-TOE environment. If the TOE does not implement a mechanism that ensures that no conflicting rules can be configured, the evaluator shall devise two equal stateful traffic filtering rules with alternate operations – permit and drop. The rules should then be deployed in two distinct orders and in each case the

© 2024 Leidos. All rights reserved.

evaluator shall ensure that the first rule is enforced in both cases by generating applicable packets and using packet capture and logs for confirmation.

The evaluator created two equal filtering rules with alternate operations (permit and deny) and deployed them with the 'deny' rule ahead of the 'permit' rule. The evaluator attempted to send traffic through the TOE and verified via logs and packet captures that the 'deny' rule was enforced. The evaluator then reversed the order of the rules and again attempted to send traffic through the TOE. The evaluator verified via logs and packet captures that the 'permit' rule was enforced on the traffic. Thus, the evaluator concluded that whichever rule the packet matches first is enforced on the packet.

Test 2: The evaluator shall repeat the procedure above, except that the two rules should be devised where one is a subset of the other (e.g., a specific address vs. a network segment). Again, the evaluator should test both orders to ensure that the first is enforced regardless of the specificity of the rule.

The evaluator created two filtering rules where the first (higher priority) rule denied all traffic from a range of network addresses and the second (lower priority) rule permitted all traffic from a specific network address within that range. The evaluator attempted to send traffic through the TOE and verified via logs and packet captures that the 'deny' rule was enforced. The evaluator then reversed the order of the rules and again attempted to send traffic through the TOE. The evaluator verified via logs and packet captures that the 'permit' rule was enforced.

FFW_RUL_EXT.1.9

For each attribute in FFW_RUL_EXT.1.2, the evaluator shall construct a test to demonstrate that the TOE can correctly compare the attribute from the packet header to the ruleset, and shall demonstrate both the permit and deny for each case. It shall also be verified that a packet is dropped if no matching rule can be identified for the packet. The evaluator shall check the log in each case to confirm that the relevant rule was applied. The evaluator shall record a packet capture for each test to demonstrate the correct TOE behaviour.

The evaluator configured stateful packet filter rules for each of the attributes specified in the requirement. A 'permit' rule and a 'drop' rule was created for different values of each attribute. The evaluator sent packets with values of each tested attribute that matched either the 'permit' rule or the 'deny' rule. The evaluator confirmed via TOE logs and packet captures that the traffic expected to be allowed through the firewall based on the ruleset was allowed through, and traffic expected to be denied based on the ruleset was denied.

FFW_RUL_EXT.1.10

The following tests shall be run using IPv4 and IPv6.

Test 1: The evaluator shall define a TCP half-open connection limit on the TOE. The evaluator shall generate TCP SYN requests to pass through the TOE to the target system using a randomised source IP address and common destination IP address. The number of SYN requests should exceed the TCP half-open threshold defined on the TOE. TCP SYN-ACK messages should not be acknowledged. The evaluator shall verify through packet capture that once the defined TCP half-open threshold has been reached, subsequent TCP SYN packets are not transmitted to the target system. The evaluator shall verify that when the configured threshold is reached that, depending upon the selection, either a log entry is generated or a counter is incremented.

The evaluator configured the TOE to accept a specific connection limit. The evaluator then generated TCP SYN packets to pass through the TOE to exceed the set connection limit. The evaluator verified that once

© 2024 Leidos. All rights reserved.

the connection limit had been met, a log entry was generated and no other SYN packets were sent through the TOE to the common destination address.

2.9.2 FFW_RUL_EXT.2 Stateful Filtering for Dynamic Protocols

These Evaluation Activities are for the Optional Requirements defined in the FW-Module.

2.9.2.1 TSS Activities

The evaluator shall verify that the TSS identifies the protocols that can cause the automatic creation of dynamic packet filtering rules. In some cases rather than creating dynamic rules, the TOE might establish stateful sessions to support some identified protocol behaviors.

Section 6.9 of [ST] ("Stateful Traffic Filtering") states the TOE creates dynamic rules for FTP based on App-ID. The App-ID identifies the application based on its unique properties and transaction characteristics used to allow transmission of data.

The evaluator shall verify that the TSS explains the dynamic nature of session establishment and removal. The TSS also shall explain any logging ramifications.

Dynamic session establishment is described in the above evaluation activity.

Section 6.9 of [ST] states that dynamic sessions are removed when FTP sessions are terminated or when the TCP timeout expires. This section also states that logging can be enabled in the security policy rule for FTP session starts, FTP session ends, or both; individual FTP commands or data packets are not logged.

The evaluator shall verify that for each of the protocols selected, the TSS explains the dynamic nature of session establishment and removal specific to the protocol.

Section 6.9 of [ST] states the TOE creates dynamic rules for FTP in accordance with RFC 959 using the FTP App-ID. This identifies the application based on its unique properties and transaction characteristics to dynamically establish the connection, determine the parameters for the session and negotiate the ports that are to be used for data transfer.

2.9.2.2 Guidance Activities

The evaluator shall verify that the guidance documentation describes dynamic session establishment capabilities.

Section 7.9 of [CCECG] ("Configure Stateful Inspection Filtering") describes the TOE's dynamic session establishment capabilities. The TOE creates dynamic rules, maintaining the session states to support processing the FTP network protocol traffic for TCP data sessions in accordance with the FTP protocol as specified in RFC 959 using the FTP App-ID. The TOE uses App-ID, the traffic classification technology, to identify traffic on the network. Logging can be enabled in the security policy rule configured to control the FTP traffic.

The evaluator shall verify that the guidance documentation describes the logging of dynamic sessions consistent with the TSS.

Section 7.9 of [CCECG] describes the TOE's dynamic session establishment logging capabilities consistent with the TSS.

© 2024 Leidos. All rights reserved.

2.9.2.3 Test Activities

Test 1: The evaluator shall define stateful traffic filtering rules to permit and log traffic for each of the supported protocols and drop and log TCP and UDP ports above 1024. Subsequently, the evaluator shall establish a connection for each of the selected protocols in order to ensure that it succeeds. The evaluator shall examine the generated logs to verify they are consistent with the guidance documentation.

The evaluator defined a rule to permit traffic over FTP and created a rule to drop all other traffic. The evaluator verified that the FTP traffic was able to traverse the TOE. FTP is the only supported protocol for this function per the [ST].

Test 2: Continuing from Test 1, the evaluator shall determine (e.g., using a packet sniffer) which port above 1024 opened by the control protocol, terminate the connection session, and then verify that TCP or UDP (depending on the protocol selection) packets cannot be sent through the TOE using the same source and destination addresses and ports.

The evaluator determined which port was opened for the communication and attempted to cause packets to traverse the TOE over the port. The evaluator observed that the packets did not traverse the TOE.

Test 3: For each additionally supported protocol, the evaluator shall repeat the procedure above for the protocol. In each case the evaluator must use the applicable RFC or standard in order to determine what range of ports to block in order to ensure the dynamic rules are created and effective.

This test is not applicable because the TOE only claims one supported protocol (FTP) for this function.

2.10 Packet Filtering (FPF) (VPNGW-SD)

2.10.1 FPF_RUL_EXT.1 Rules for Packet Filtering

2.10.1.1 TSS Activities

FPF_RUL_EXT.1.1

The evaluator shall verify that the TSS provide a description of the TOE's initialization and startup process, which clearly indicates where processing of network packets begins to take place, and provides a discussion that supports the assertion that packets cannot flow during this process. The evaluator shall verify that the TSS also includes a narrative that identifies the components (e.g., active entity such as a process or task) involved in processing the network packets and describes the safeguards that would prevent packets flowing through the TOE without applying the ruleset in the event of a component failure. This could include the failure of a component, such as a process being terminated, or a failure within a component, such as memory buffers full and cannot process packets.

Section 6.10 of [ST] ("Packet Filtering") describes the packet filtering function as a subset of the TOE's stateful traffic filtering function, which is described in section 6.9 ("Stateful Traffic Filtering"). Section 6.9 of [ST] states network traffic can only be processed by the TOE if the Policy Enforcement Module is fully functional. The Policy Enforcement Module is not started until the power-on self-tests pass and the control plane and data plane modules are booted. The control plane loads configuration information into the Policy Enforcement Module, which runs on the data plane. Once this has occurred, the Policy Enforcement Module is able to begin processing packets.

© 2024 Leidos. All rights reserved.

Section 6.9 of [ST] identifies several safeguards that prevent traffic flow without analysis. Specifically, traffic cannot flow while the TOE is being booted, while traffic or system logs are full (if configured to behave this way), or when a self-test failure occurs. This section also references the existence of DoS Protection profile policy rules that can be used to ensure that an excessive volume of traffic is dropped rather than potentially overwhelming the TOE's policy enforcement engine.

FPF_RUL_EXT.1.2 and FPF_RUL_EXT.1.3

There are no EAs specified for this element. Definition of packet filtering policy, association of operations with packet filtering rules, and association of these rules to network interfaces is described collectively under FPF_RUL_EXT.1.4.

FPF_RUL_EXT.1.4

The evaluator shall verify that the TSS describes a packet filtering policy that can use the following fields for each identified protocol, and that the RFCs identified for each protocol are supported:

- IPv4 (RFC 791)

Source address

Destination Address

Protocol

- IPv6 (RFC 2460)
  - source address
  - destination address
  - next header (protocol)
- TCP (RFC 793)
  - source port
  - destination port
- UDP (RFC 768)
  - source port
  - destination port

Section 6.9 of [ST] identifies the TOE's support for IPv4, IPv6, TCP, and UDP traffic. This section identifies that source/destination subject (address), source/destination subject service identifier (port), and protocol are used for rule processing. IPv6 traffic can also be filtered on IPv6 extension headers, which includes the Next Header field as required by the SFR.

FPF_RUL_EXT.1.4

The evaluator shall verify that the TSS describes how conformance with the identified RFCs has been determined by the TOE developer (e.g., third party interoperability testing, protocol compliance testing).

Section 6.9 of [ST] states that conformance with the RFC 792 (ICMPv4), RFC 4443 (ICMPv6), RFC 791(IPv4), RFC 2460 (IPv6), RFC 793 (TCP), RFC 768 (UDP) protocols is verified by Palo Alto through regular quality assurance, regression, and interoperability testing.

FPF_RUL_EXT.1.4

The evaluator shall verify that each rule can identify the following actions: permit, discard, and log.

© 2024 Leidos. All rights reserved.

Section 6.9 of [ST] states an administrator may configure the TOE to apply packet filtering rules with permit, drop, and log actions.

FPF_RUL_EXT.1.4
The evaluator shall verify that the TSS identifies all interface types subject to the packet filtering policy and explains how rules are associated with distinct network interfaces. Where interfaces can be grouped into a common interface type (e.g., where the same internal logical path is used, perhaps where a common device driver is used), they can be treated collectively as a distinct network interface.

Section 6.9 of [ST] states the TOE groups interfaces into security zones. Separate zones must be created for each type of interface (Layer 2, Layer 3, virtual wire) and each interface must be assigned to a zone before it can process traffic. Security policies provide the rule sets that specify whether to block or allow network connections. Security policies can be defined only between zones of the same type and be assigned to a distinct network interface.

FPF_RUL_EXT.1.5
The evaluator shall verify that the TSS describes the algorithm applied to incoming packets, including the processing of default rules, determination of whether a packet is part of an established session, and application of administrator defined and ordered ruleset.

Section 6.10 of [ST] summarizes the algorithm for applying rules in a packet filtering table, where the ordering is from left to right and from top to bottom. Traffic is assessed against all rules, and the first rule that meets the defined criteria determines how the traffic is processed. This section states it is important to define rules in decreasing order of specificity since the TOE's rule processing engine assumes the first match is the best match.

Section 6.9 of [ST] summarizes the packet processing behavior with respect to stateful traffic filtering. Specifically, fragmented packets are reassembled and the TSF determines whether or not the traffic is associated with an existing session. If it is, then the traffic is allowed.

Section 6.10 of [ST] identifies default rules as denying all traffic that is not permitted by another rule, allowing all intrazone traffic, and denying all traffic between zones.

FPF_RUL_EXT.1.6
The evaluator shall verify that the TSS describes the process for applying packet filtering rules and also that the behavior (either by default, or as configured by the administrator) is to discard packets when there is no rule match. The evaluator shall verify the TSS describes when the IPv4 and IPv6 protocols supported by the TOE differ from the full list provided in the RFC Values for IPv4 and IPv6 table.

Section 6.10 of [ST] describes the rule processing as a packet filtering table, where security policies are evaluated left to right and top to bottom. The first rule match for the traffic determines how it is processed, and when there is no rule match, the default behavior is to deny the information flow.

This section states that protocols are processed as specified in the relevant RFCs, so it is understood that there is no difference in the TOE's implementation of this.

2.10.1.2 Guidance Activities

FPF_RUL_EXT.1.1
The operational guidance associated with this requirement is assessed in the subsequent test EAs.

© 2024 Leidos. All rights reserved.

Section 7.9 of [CCECG] (“Configure Stateful Inspection Filtering”) provides instructions to configure the traffic rules. Any configurable defaults and how to configure each of the applicable rule attributes, actions, and associated interfaces are covered in the activities below. FPF_RUL_EXT.1.6 covers instruction that would allow an administrator to ensure that configured rules are properly ordered.

FPF_RUL_EXT.1.2 and FPF_RUL_EXT.1.3

There are no EAs specified for this element. Definition of packet filtering policy, association of operations with packet filtering rules, and association of these rules to network interfaces is described collectively under FPF_RUL_EXT.1.4.

FPF_RUL_EXT.1.4

The evaluator shall verify that the operational guidance identifies the following protocols as being supported and the following attributes as being configurable within packet filtering rules for the associated protocols:

- IPv4 (RFC 791)
  - destination address
  - protocol
- IPv6 (RFC 2460)
  - source address
  - destination address
  - next header (protocol)
- TCP (RFC 793)
  - source port
  - destination port
- UDP (RFC 768)
  - source port
  - destination port

Section 7.9 of [CCECG] describes each of the above protocols; how policies are defined; and provides instructions to configure the attributes identified above within the Packet filtering rules.

FPF_RUL_EXT.1.4

The evaluator shall verify that the operational guidance indicates that each rule can identify the following actions: permit, discard, and log.

Section 7.9 of [CCECG] provides instructions to configure the security policy rules including permit traffic (allow), discard traffic (block), and log. Example configuration steps are shown to choose a ‘Deny’ action with ‘Log at Session End’. The alternative action is described as ‘allow’ and logging can also be selected to occur at session start rather than at session end.

FPF_RUL_EXT.1.4

The evaluator shall verify that the operational guidance explains how rules are associated with distinct network interfaces.

© 2024 Leidos. All rights reserved.

The guidance may describe the other protocols contained within the ST (e.g., IPsec, IKE, potentially HTTPS, SSH, and TLS) that are processed by the TOE. The evaluator shall ensure that it is made clear what protocols were not considered as part of the TOE evaluation.

Section 7.9 of [CCECG] provides instructions to associate rules with distinct interfaces by configuring the source and destination zones and provides an example.

Section 1.1 of [CCECG] ("Common Criteria (CC) Evaluated Configuration") makes clear that only the protocols specified in [ST] (i.e., TLS, HTTPS, SSH, and IKE/IPsec) are within the scope of evaluation, and only to the extent they have been specified by the SFRs.

FPF_RUL_EXT.1.5

The evaluator shall verify that the operational guidance describes how the order of packet filtering rules is determined and provides the necessary instructions so that an administrator can configure the order of rule processing.

Section 7.9 of [CCECG] provides instructions to configure the ordering of rule processing. Security policies are evaluated left to right and from top to bottom. A packet is matched against the first rule that meets the defined criteria; after a match is triggered the subsequent rules are not evaluated. Therefore, the more specific rules must precede more generic ones in order to enforce the best match criteria.

FPF_RUL_EXT.1.6

The evaluator shall verify that the operational guidance describes the behavior if no rules or special conditions apply to the network traffic. If the behavior is configurable, the evaluator shall verify that the operational guidance provides the appropriate instructions to configure the behavior to discard packets with no matching rules. The evaluator shall verify that the operational guidance describes the range of IPv4 and IPv6 protocols supported by the TOE.

Section 7.9 of [CCECG] states that for all traffic that doesn't match any defined rules, the default rules apply. The default rules allow all intrazone (within the same zone) traffic and deny all interzone (between different zones, e.g., 'trust' and 'untrust') traffic. Typically, intrazone traffic is considered to be trusted however both intrazone and interzone traffic can be configured to deny all traffic if there is no rule match by clicking on the security policy and clicking on the Override button on the bottom on the Policy ->Security screen. In the evaluated configuration, the default deny all rule for interzone traffic must not be modified.

Section 7.9 of [CCECG] describes the range of IPv4 and IPv6 protocols supported. When configuring stateful traffic filtering on IPv4 and IPv6, the administrator can enter any number in the range 0-255 for the transport layer protocol, with the exception of 6 (TCP) and 17 (UDP), for which the administrator configures separate rules and policies.

2.10.1.3 Test Activities

FPF_RUL_EXT.1.1

Test 1: The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would otherwise be denied by the ruleset should be sourced and directed to a host. The evaluator shall use a packet sniffer to verify none of the generated network traffic is permitted through the TOE during initialization.

© 2024 Leidos. All rights reserved.

The evaluator caused the TOE to reboot and directed traffic at the TOE that should be denied by its ruleset. The evaluator verified that the TOE did not permit the traffic to pass while being initialized or after the TOE was initialized.

FPF_RUL_EXT.1.1

Test 2: The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would be permitted by the ruleset should be sourced and directed to a host. The evaluator shall use a packet sniffer to verify none of the generated network traffic is permitted through the TOE during initialization and is only permitted once initialization is complete.

Note: The remaining testing associated with application of the ruleset is addressed in the subsequent test EAs.

The evaluator caused the TOE to reboot and directed traffic at the TOE that should be permitted by its ruleset. The evaluator verified that the TOE did not permit the traffic to pass while being initialized, and observed that once initialization was complete the traffic was permitted to traverse the TOE.

FPF_RUL_EXT.1.2 and FPF_RUL_EXT.1.3

There are no EAs specified for this element. Definition of packet filtering policy, association of operations with packet filtering rules, and association of these rules to network interfaces is described collectively under FPF_RUL_EXT.1.4.

FPF_RUL_EXT.1.4

The evaluator shall perform the following tests:

Test 1: The evaluator shall use the instructions in the operational guidance to test that packet filter rules can be created that permit, discard, and log packets for each of the following attributes:

- IPv4
  - destination address
  - protocol
- IPv6
  - source address
  - destination address
  - next header (protocol)
- TCP
  - source port
  - destination port
- UDP
  - source port
  - destination port

The evaluator performed this in conjunction with the FPF_RUL_EXT.1.6 testing.

© 2024 Leidos. All rights reserved.

FPF_RUL_EXT.1.4

Test 2: The evaluator shall repeat Test 1 above for each distinct network interface type supported by the TOE to ensure that packet filtering rules can be defined for all supported types.

Note that these test activities should be performed in conjunction with those of FPF_RUL_EXT.1.6 where the effectiveness of the rules is tested; here the evaluator is just ensuring the guidance is sufficient and the TOE supports the administrator creating a ruleset based on the above attributes. The test activities for FPF_RUL_EXT.1.6 define the combinations of protocols and attributes required to be tested. If those combinations are configured manually, that will fulfill the objective of these test activities, but if those combinations are configured otherwise (e.g., using automation), these test activities may be necessary in order to ensure the guidance is correct and the full range of configurations can be achieved by a TOE administrator.

The evaluator performed this in conjunction with the FPF_RUL_EXT.1.6 testing.

FPF_RUL_EXT.1.5

The evaluator shall perform the following tests:

Test 1: The evaluator shall devise two equal packet filtering rules with alternate operations – permit and discard. The rules should then be deployed in two distinct orders and in each case the evaluator shall ensure that the first rule is enforced in both cases by generating applicable packets and using packet capture and logs for confirmation.

Test 2: The evaluator shall repeat the procedure above, except that the two rules should be devised where one is a subset of the other (e.g. a specific address vs. a network segment). Again, the evaluator should test both orders to ensure that the first is enforced regardless of the specificity of the rule.

The evaluator created two equal filtering rules with alternate operations (permit and deny) and deployed them with the 'deny' rule ahead of the 'permit' rule. The evaluator attempted to send traffic through the TOE and verified via logs and packet captures that the 'deny' rule was enforced. The evaluator then reversed the order of the rules and again attempted to send traffic through the TOE. The evaluator verified via logs and packet captures that the 'permit' rule was enforced.

The evaluator created two filtering rules where the first (higher priority) rule denied all traffic from a range of network addresses and the second (lower priority) rule permitted all traffic from a specific network address within that range. The evaluator attempted to send traffic through the TOE and verified via logs and packet captures that the 'deny' rule was enforced. The evaluator then reversed the order of the rules and again attempted to send traffic through the TOE. The evaluator verified via logs and packet captures that the 'permit' rule was enforced.

FPF_RUL_EXT.1.6

The evaluator shall perform the following tests:

Test 1: The evaluator shall configure the TOE to permit and log each supported IPv4 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each supported IPv4 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that the supported protocols are permitted (i.e., by capturing the packets after passing through the TOE) and logged. Any protocols not supported by the TOE must be denied.

© 2024 Leidos. All rights reserved.

The evaluator configured the TOE with a rule to permit and log each IPv4 protocol. The evaluator verified that the TOE enforced the configured rule.

Test 2: The evaluator shall configure the TOE to permit all traffic except to discard and log each supported IPv4 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv4 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that the supported protocols are denied (i.e., by capturing no applicable packets passing through the TOE) and logged. Any protocols not supported by the TOE must also be denied but are not required to be logged.

The evaluator configured the TOE with a rule to discard and log each IPv4 protocol. The evaluator verified that the TOE enforced the configured rule.

Test 3: The evaluator shall configure the TOE to permit and log each supported IPv4 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. Additionally, the evaluator shall configure the TOE to discard and log each supported IPv4 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with different (than those permitted above) combinations of a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each supported IPv4 Transport Layer Protocol and outside the scope of all source and destination addresses configured above in order to ensure that the supported protocols are denied (i.e., by capturing no applicable packets passing through the TOE) and logged. Any protocols not supported by the TOE must be denied.

The evaluator configured the TOE with a rule to permit some IPv4 protocols and deny all other IPv4 protocols and to log each result. The evaluator verified that the TOE enforced the configured rule.

Test 4: The evaluator shall configure the TOE to permit and log each supported IPv6 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that the supported protocols are permitted (i.e., by capturing the packets after passing through the TOE) and logged. Any protocols not supported by the TOE must be denied.

The evaluator configured the TOE with a rule to permit and log each IPv6 protocol. The evaluator verified that the TOE enforced the configured rule.

Test 5: The evaluator shall configure the TOE to permit all traffic except to discard and log each supported IPv6 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard

© 2024 Leidos. All rights reserved.

source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that the supported protocols are denied (i.e., by capturing no applicable packets passing through the TOE) and logged. Any protocols not supported by the TOE must also be denied but are not required to be logged.

The evaluator configured the TOE with a rule to discard and log each IPv6 protocol. The evaluator verified that the TOE enforced the configured rule.

Test 6: The evaluator shall configure the TOE to permit and log each supported IPv6 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. Additionally, the evaluator shall configure the TOE to discard and log each supported IPv6 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with different (than those permitted above) combinations of a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and outside the scope of all source and destination addresses configured above in order to ensure that the supported protocols are dropped (i.e., by capturing no applicable packets passing through the TOE) and logged. Any protocols not supported by the TOE must be denied.

The evaluator configured the TOE with a rule to permit some IPv6 protocols and deny all other IPv6 protocols and to log each result. The evaluator verified that the TOE enforced the configured rule.

Test 7: The evaluator shall configure the TOE to permit and log protocol 6 (TCP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination TCP ports in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged.

The evaluator configured the TOE with a rule to permit and log TCP packets. The evaluator verified that the TOE enforced the configured rule.

Test 8: The evaluator shall configure the TOE to discard and log protocol 6 (TCP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination TCP ports in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.

The evaluator configured the TOE with a rule to discard and log TCP packets. The evaluator verified that the TOE enforced the configured rule.

Test 9: The evaluator shall configure the TOE to permit and log protocol 17 (UDP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination UDP ports in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged. Here the evaluator ensures that the UDP port 500 (IKE) is included in the set of tests.

© 2024 Leidos. All rights reserved.

The evaluator configured the TOE with a rule to permit and log UDP packets. The evaluator verified that the TOE enforced the configured rule.

Test 10: The evaluator shall configure the TOE to discard and log protocol 17 (UDP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination UDP ports in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged. Again, the evaluator ensures that UDP port 500 is included in the set of tests.

Refer to the RFC Values for IPv4 and IPv6 table in [VPNGW-SD] for the RFC defined values for the protocol fields for IPv4 and IPv6 to be used in configuring and otherwise testing Packet Filtering rule definition and enforcement.

The evaluator configured the TOE with a rule to discard and log UDP packets. The evaluator verified that the TOE enforced the configured rule.

2.11 Intrusion Prevention (IPS) (IPS-SD)

2.11.1 IPS_ABD_EXT.1 Anomaly-Based IPS Functionality

2.11.1.1 TSS Activities

The evaluator shall verify that the TSS describes the composition, construction, and application of baselines or anomaly-based attributes specified in IPS_ABD_EXT.1.1.

Section 6.11 of [ST] (“Intrusion Prevention”) describes the supported anomaly-based traffic detection via configurable correlation baseline, using the following packet attributes: IP addresses; ports; protocols; data payload. For example, the Security Administrator can configure an FTP login brute-force attempts detection and if the attempt threshold exceeds 10 attempts in 60 seconds (configurable) from the same source and/or destination address pair, then it will trigger the configured action. Frequency detection settings can be applied to any unique event (e.g., brute-force, Denial of Service, flooding attack). Frequency detection is part of the Vulnerability Protection profile security policy. The Vulnerability Protection profile is tied to a security policy rule which can be associated with source and destination zones. The security zones are tied to any network data interface (not management interface).

The evaluator shall verify that the TSS provides a description of how baselines are defined and implemented by the TOE, or a description of how anomaly-based rules are defined and configured by the administrator.

If ‘frequency’ is selected in IPS_ABD_EXT.1.1, the TSS shall include an explanation of how frequencies can be defined on the TOE.

If ‘thresholds’ is selected in IPS_ABD_EXT.1.1, the TSS shall include an explanation of how the thresholds can be defined on the TOE.

Section 6.11 of [ST] describes the supported anomaly-based traffic detection via configurable, correlation baseline. This is also described above. The description includes how frequencies are defined.

The evaluator shall verify that each baseline or anomaly-based rule can be associated with a reaction specified in IPS_ABD_EXT.1.3.

Section 6.11 of [ST] states the Security Administrator can configure the following actions that can be triggered when a configured frequency threshold is exceeded:

© 2024 Leidos. All rights reserved.

- Allow (i.e., allow the traffic flow)
- Alert (i.e., allow the traffic flow and also record a threat log)
- Reset Client (i.e., send a TCP reset to the source address of the offending traffic)
- Reset Server (i.e., send a TCP reset to the destination address of the offending traffic)
- Reset Both
- Drop (i.e., drop the traffic flow).

The evaluator shall verify that the TSS identifies all interface types capable of applying baseline or anomaly-based rules and explains how they are associated with distinct network interfaces. Where interfaces can be grouped into a common interface type (e.g., where the same internal logical path is used, perhaps where a common device driver is used) they can be treated collectively as a distinct network interface.

Section 6.11 of [ST] states the anomaly-based rules can be applied to any network data interface (not management interface) and describes how they are associated with distinct network interfaces.

2.11.1.2 Guidance Activities

The evaluator shall verify that the operational guidance provides instructions to manually create baselines or anomaly-based rules according to the selections made in IPS_ABD_EXT.1.1. Note that dynamic “profiling” of a network to establish a baseline is outside the scope of the PP-Module.

Section 7.11.2 of [CCECG] (“Configure Anomaly-Based Detection”) describes how to configure frequency detection on security events that can trigger an action from the Vulnerability/Threat signature rule. A profile is created and attached to a security policy rule. The frequency attempt threshold is configurable via “Duration (sec)” and the number of hits per second. Security policy rules can be associated with source and destination zones, while the security zones are tied to any network data interface (not management interface). Because the profile is tied to a security policy rule, it can be applied to all packet header and data elements defined in IPS_SBD_EXT.1. Sectio 7.11.2 identifies the actions as Allow, Alert (allow and record a threat log), Block IP, Reset Server, Reset Client, Reset Both, or Drop.

The evaluator shall verify that the operational guidance provides instructions to associate reactions specified in IPS_ABD_EXT.1.3 with baselines or anomaly-based rules.

Section 7.11.2 of [CCECG] describes how to configure actions associated with IPS_ABD_EXT.1.3 baselines.

The evaluator shall verify that the operational guidance provides instructions to associate the different policies with distinct network interfaces.

Section 7.11.2 of [CCECG] describes how to attach the profile to security policy rules. Sections 7.11 of [CCECG] (“Configure Threat Prevention”) describes how a security policy rule can be associated with source and destination zones, and how the security zones are tied to any network data interface (not management interface).

2.11.1.3 Test Activities

Test 1: The evaluator shall use the instructions in the operational guidance to configure baselines or anomaly-based rules for each attributes specified in IPS_ABD_EXT.1.1. The evaluator shall send traffic that does not match the baseline or matches the anomaly- based rule and verify the TOE applies the configured reaction. This shall be performed for each attribute in IPS_ABD_EXT.1.1.

© 2024 Leidos. All rights reserved.

The evaluator created a signature to define the anomaly rules. The evaluator directed traffic across the TOE that matches the anomalous traffic. The evaluator verified that the TOE directed the anomaly and performed the configured reaction.

Test 2: The evaluator shall repeat the test above to ensure that baselines or anomaly- based rules can be defined for each distinct network interface type supported by the TOE.

The evaluator observed that regardless of the physical interface type used by the TOE, the logical interface is always configured as Ethernet. There are no other types of logical interfaces for which stateful traffic filtering rules can be defined.

2.11.2 IPS_IPB_EXT.1 IP Blocking

2.11.2.1 TSS Activities

The evaluator shall verify how good/bad lists affect the way in which traffic is analyzed with respect to processing packets. The evaluator shall also very that the TSS provides details for the attributes that create a known good list, a known bad list, and their associated rules, including how to define the source or destination IP address (e.g. a single IP address or a range of IP addresses).

If the TSF uses address types other than a single IP or a range of IP addresses (e.g. MAC addresses), the evaluator shall check that the TSS explains what configurations would cause non-IP lists of known-good and known-bad addresses to take precedence over IP-based address lists.

Section 6.11 of [ST] (“Intrusion Prevention”) describes the default packet processing flow and attributes as follows. The TOE supports a security rules-based known-good or known-bad list rules mechanism, referred to as the External Dynamic List (EDL) mechanism. The EDL must be referenced in the security policy rule or profile. EDL supports a predefined or custom IP address list, which is an address list of IP addresses (single or range), URLs, or domains. The scope of the evaluation focuses on IP addresses only, as URLs and domains are just IP addresses translated via DNS. The TOE supports multiple EDLs in a security policy rule. When multiple lists are referenced, the Security Administrator can prioritize the order of evaluation to ensure the most important EDLs are checked first. The TOE includes built-in EDLs and EDLs can be defined externally and downloaded into the TOE. The Security Administrator cannot modify the contents of the built-in lists but can create custom EDLs. When viewing the entries of an EDL, the Security Administrator can exclude up to 100 entries from the list (i.e., known-good IP addresses). The TOE does not support filtering based on MAC addresses.

The TOE does not use address types other than a single IP address or range of IP addresses.

The evaluator shall also verify that the TSS identifies all the roles and level of access for each of those roles that have been specified in the requirement.

The requirement identifies the Security Administrators as able to configure the following IPS policy elements: known-good list rules; known-bad list rules; and IP addresses. The TSS includes this information in Section 6.11.

2.11.2.2 Guidance Activities

The evaluator shall verify that the administrative guidance provides instructions with how each role specified in the requirement can create, modify and delete the attributes of a known good and known bad lists.

© 2024 Leidos. All rights reserved.

If the TSF uses address types other than a single IP or a range of IP addresses (e.g. MAC addresses), the evaluator shall check that the operational guidance includes instructions for any configurations that would cause non-IP lists of known-good and known-bad addresses to take precedence over IP-based address lists.

Section 7.11.3 of [CCECG] ("Configure External Dynamic List") provides instructions to the Security Administrator for configuring the known good and known bad lists using an External Dynamic List (EDL). The EDL is enforced via a security policy rule, and the Security Administrator can block (i.e., blacklist) or allow (i.e., whitelist) traffic based on IP addresses.

2.11.2.3 Test Activities

Test 1: The evaluator shall use the instructions in the operational guidance to create a known-bad address list. Using a single IP address, a list of addresses or a range of addresses from that list, the evaluator shall attempt to send traffic through the TOE that would otherwise be allowed by the TOE and observe the TOE automatically drops that traffic.

The evaluator configured the TOE to utilize a known-bad address list. The evaluator configured the TOE to block traffic which is from sources on the known-bad address list. The evaluator verified that the TOE blocks the traffic.

Test 2: The evaluator shall use the instructions in the operational guidance to create a known-good address list. Using a single IP address, a list of addresses or a range of addresses from that list, the evaluator shall attempt to send traffic that would otherwise be denied by the TOE and observe the TOE automatically allowing traffic.

The evaluator configured the TOE to utilize a known-good address list. The evaluator configured the TOE to allow traffic which is from sources on the known-good address list. The evaluator verified that the TOE permits the traffic.

Test 3: The evaluator shall add conflicting IP addresses to each list and ensure that the TOE handles conflicting traffic in a manner consistent with the precedence in IPS_NTA_EXT.1.1.

The TOE enforces the known-good or known-bad address list with policies. The policies are evaluated in a top-down manner where the first policy hit is always enforced. This functionality has already been tested as part of FFW_RUL_EXT.1 and FPF_RUL_EXT.1 thus the evaluator is certain that the TOE handles conflicting traffic in the top-down, first policy hit manner.

2.11.3 IPS_NTA_EXT.1 Network Traffic Analysis

2.11.3.1 TSS Activities

IPS_NTA_EXT.1.1

The evaluator shall verify that the TSS explains the TOE's capability of analyzing IP traffic in terms of the TOE's policy hierarchy (precedence). The TSS should identify if the TOE's policy hierarchy order is configurable by the administrator for IPS policy elements (known-good lists, known-bad lists, signature-based rules, and anomaly-based rules).

Regardless of whether the precedence is configurable, the evaluator shall verify that the TSS describes the default precedence as well as the IP analyzing functions supported by the TOE.

© 2024 Leidos. All rights reserved.

Section 6.11 of [ST] (“Intrusion Prevention”) describes the TOE’s default packet processing flow. A packet is received from the ingress network interface. The TOE parses the packet and determines whether the packet is subject to further inspection (the TOE must be deployed inline to meet both firewall and IPS requirements). If so, the TOE will continue with a session lookup after the packet enters the security processing stage. During the processing stage, the TOE may discard the packet due to protocol or security violation. In certain cases, due to the TOE attack prevention features (part of the Zone Protection profile), it will discard packets with or without any configurable options. Next, in the session lookup stage, the TOE will determine if the packet is part of an existing or new session. If it is part of an existing session, the TOE will determine if content ('payload’) inspection is required (per security policy rule). If it is not part of an existing session, then the firewall (i.e., security policy) rules will be applied (in order) to determine if the packet will be dropped or a new session will be set up. If the packet is allowed by the security policy rule, the session will be added to the allowed session table and the TOE will determine if content inspection will be applied based on the IPS policy configured per security policy rule with logging options. If content inspection is configured and the TOE is in inline deployment, any match of the signature rules with action set to drop will cause the packet to be discarded. If content inspection is not configured or no signature rule matches, then the packet will be sent to the Forward/Route stage where it will be sent out the appropriate egress network interface.

Section 6.11 of [ST] describes the following IP analyzing functions supported by the TOE:

- Anomaly-based traffic detection via configurable correlation baseline
- Security rules-based known-good and known-bad lists of source IP addresses using External Dynamic Lists (EDLs)
- Signature rules based on packet header and payload fields.

Section 6.11 of [ST] states the Security Administrator can configure the IPS policies order.

IPS_NTA_EXT.1.2

The evaluator shall verify that the TSS indicates that the following protocols are supported:

- IPv4
- IPv6
- ICMPv4
- ICMPv6
- TCP
- UDP

Section 6.11 of [ST] states the TOE supports all protocols (IPv4, IPv6, ICMPv4, ICMPv6, TCP, and UDP).

IPS_NTA_EXT.1.2

The evaluator shall verify that the TSS describes how conformance with the identified protocols has been determined by the TOE developer (e.g., third party interoperability testing, protocol compliance testing).

Section 6.11 of [ST] states the vendor has taken the TOE through extensive third-party interoperability testing (e.g., DoDIN-APL, ICSA, NSS Labs) and protocol compliance testing (e.g., USGv6).

© 2024 Leidos. All rights reserved.

IPS_NTA_EXT.1.3

The evaluator shall verify that the TSS identifies all interface types capable of being deployed in the modes of promiscuous, and or inline mode as well as the interfaces necessary to facilitate each deployment mode (at a minimum, the interfaces need to support inline mode). The evaluator shall also check that the TSS provides a description for how the management interface is logically distinct from any sensor interfaces.

Section 6.11 of [ST] states the TOE supports multiple network interface types from Tap (Promiscuous), Virtual Wire, Layer 2, and Layer 3 (inline) interfaces and describes the interfaces necessary to facilitate each. The TOE also supports a dedicated management (MGMT) port to be connected to an isolated management network as well.

2.11.3.2 Guidance Activities

IPS_NTA_EXT.1.1

The evaluator shall verify that the guidance describes the default precedence.

If the precedence is configurable, the evaluator shall verify that the guidance explains how to configure the precedence.

Section 7.11 of [CCECG] (“Configure Threat Prevention”) describes the TOE’s default precedence for applying IPS policies. The TOE first applies L3 & L4 header rules, which are ordered from top to bottom with the first match triggering the configured action. The L3 & L4 header rules, which focus on inspecting the header content fields of IPv4, IPv6, ICMPv4, ICMPv6, TCP, and UDP packets, are applied to all packets assigned to security zone in the Zone Protection profile. The TOE then applies security policy rules (i.e., firewall rules), which again are ordered from top to bottom, on traffic traversing from the configured source security zone to the configured destination security zone. Finally, the TOE applies Vulnerability/Threat signature rules, which are for inspecting packet payloads and are applied as part of the firewall rules.

The precedence for applying the three types of rules is not configurable, but the order in which rules of a particular type are applied is configurable and [CCECG] explains how to configure the precedence.

IPS_NTA_EXT.1.2

There are no guidance EAs for this element.

IPS_NTA_EXT.1.3

The evaluator shall verify that the operational guidance provides instructions on how to deploy each of the deployment methods outlined in the TSS. The evaluator shall also verify that the operational guidance provides instructions of applying IPS policies to interfaces for each deployment mode. If the management interface is configurable, the evaluator shall verify that the operational guidance explains how to configure the interface as a management interface.

The deployment methods outlined in section 6.11 of [ST] state the TOE supports multiple network interface deployment types from Tap (Promiscuous), Virtual Wire, Layer 2, and Layer 3 (inline) interfaces. The TOE also supports a dedicated management (MGMT) port to be connected to an isolated management network as well.

Section 7.11 of [CCECG] (“Configure Threat Prevention”) states the network data ports can be configured as a Tap (“Promiscuous”) or Virtual Wire port or can be configured as a 'Layer 2/3’ inline pair ports. In the

© 2024 Leidos. All rights reserved.

evaluated configuration, the firewall should be configured for inline deployment mode to support blocking action. Section 7.11.1 of [CCECG] (“Configure Inline Deployment Interfaces”) provides instructions on how to deploy Inline Deployment Interfaces. These same instructions show selections for configuring Tap, Virtual Wire, Layer 2 or Layer 3 on the Ethernet Interface tab as well as an option for virtual system and setting the security zone. This section also provides instructions for applying IPS policies to interfaces and states that the management interface is a dedicated interface and thus not configurable.

IPS_NTA_EXT.1.3
The evaluator shall verify that the operational guidance explains how the TOE sends commands to remote traffic filtering devices if this functionality is supported.

The TOE does not support sending commands to remote traffic filtering devices and therefore this activity is not applicable.

2.11.3.3 Test Activities

IPS_NTA_EXT.1.1 and IPS_NTA_EXT.1.2
There are no test EAs for this element.

Per the evaluation activity there is no activity to be completed.

IPS_NTA_EXT.1.3
Testing for this element is performed in conjunction with testing where promiscuous and inline interfaces are tested.

This testing has been performed in conjunction with testing of IPS_ABD_EXT.1 and IPS_SBD_EXT.1, where rules for anomaly-based detection and signature-based detection are exercised.

2.11.4 IPS_SBD_EXT.1 Signature-Based IPS Functionality

2.11.4.1 TSS Activities

IPS_SBD_EXT.1.1
The evaluator shall verify that the TSS describes what is comprised within a signature rule.

Section 6.11 of [ST] (“Intrusion Prevention”) describes layer 3 and layer 4 threat signature rules based on header and payload fields. Signature rules include the following elements: rule name; unique threat identifier; packet capture; exempt IP; log severity; log interval; action; signature; and pattern match.

IPS_SBD_EXT.1.1
The evaluator shall verify that each signature can be associated with a reaction specified in IPS_SBD_EXT.1.5.

Section 6.11 of [ST] (“Intrusion Prevention”) states each signature rule must be associated with an action, which will be triggered if the signature matches traffic otherwise allowed by the associated security policy. Possible actions are:

- Allow (i.e., allow the traffic flow)
- Alert (i.e., allow the traffic flow and also record a threat log)
- Reset Client (i.e., send a TCP reset to the source address of the offending traffic)

© 2024 Leidos. All rights reserved.

- Reset Server (i.e., send a TCP reset to the destination address of the offending traffic)
- Reset Both
- Drop (i.e., drop the traffic flow).

IPS_SBD_EXT.1.1
The evaluator shall verify that the TSS identifies all interface types that are capable of applying signatures and explains how rules are associated with distinct network interfaces. Where interfaces can be grouped into a common interface type (e.g., where the same internal logical path is used, perhaps where a common device driver is used) they can be treated collectively as a distinct network interface.

Section 6.11 of [ST] states that signature rules are assigned to a Zone Protection profile which is then associated with a Security Zone. A Security Zone is then applied to a network data interface (e.g., ethernet1/1). Section 6.11 also states the TOE supports all protocols (IPv4, IPv6, ICMPv4, ICMPv6, TCP, and UDP) on all the network data interfaces.

IPS_SBD_EXT.1.2
The evaluator shall verify that the TSS describes what is comprised within a string-based detection signature.

Section 6.11 of [ST] describes the string-based detection rule as including a rule name, unique threat identifier, packet capture, exempt IP, log severity, log interval, action, signature, and pattern match. The signature element includes its name, an optional description, its scope (full session or single transaction), and signature matching conditions. If pattern matching is specified, it also includes a pattern matching context (FTP, HTTP, SMTP) and signature pattern (defined using a regular expression).

IPS_SBD_EXT.1.2
The evaluator shall verify that each packet payload string-based detection signature can be associated with a reaction specified in IPS_SBD_EXT.1.5.

Section 6.11 of [ST] states each signature rule must be associated with an action, which will be triggered if the signature matches traffic otherwise allowed by the associated security policy. This applies to both packet header contents and packet payload string-based detection signatures. Possible actions are:

- Allow (i.e., allow the traffic flow)
- Alert (i.e., allow the traffic flow and also record a threat log)
- Reset Client (i.e., send a TCP reset to the source address of the offending traffic)
- Reset Server (i.e., send a TCP reset to the destination address of the offending traffic)
- Reset Both
- Drop (i.e., drop the traffic flow).

IPS_SBD_EXT.1.3
The evaluator shall verify that the TSS describes how the attacks defined in IPS_SBD_EXT.1.3 are processed by the TOE and what reaction is triggered when these attacks are identified.

Section 6.11 of [ST] states the TOE processes IPS_SBD_EXT.1.3 attacks as part of the Zone Protection profile prior to the security policy rule. However, the Security Administrator can configure the action in the Zone Protection profile in the same manner as the security policy rule (i.e., specify the reaction as Allow, Alert, Reset Client, Reset Server, Reset Both, or Drop). The TSS notes some attacks are always

© 2024 Leidos. All rights reserved.

dropped, such as land, ping of death, teardrop, IP spoof, MAC spoof, and ICMP fragment attacks. These are tracked with counters per network interface.

IPS_SBD_EXT.1.4
The evaluator shall verify that the TSS describes how the attacks defined in IPS_SBD_EXT.1.4 are processed by the TOE and what reaction is triggered when these attacks are identified.

Section 6.11 of [ST] states the TOE processes IPS_SBD_EXT.1.4 attacks as part of the Zone Protection profile prior to the security policy rule. However, the Security Administrator can configure the action in the Zone Protection profile in the same manner as the security policy rule (i.e., specify the reaction as Allow, Alert, Reset Client, Reset Server, Reset Both, or Drop).

IPS_SBD_EXT.1.5 and IPS_SBD_EXT.1.6
There are no TSS EAs for these elements.

2.11.4.2 Guidance Activities

Modified in accordance with TD0722.

IPS_SBD_EXT.1.1
The evaluator shall verify that the operational guidance provides instructions with how to create and/or configure rules using the following protocols and header inspection fields:

- IPv4: version; header length; packet length; ID; IP flags; fragment offset; time to live (TTL); protocol; header checksum; source address; destination address; IP options; and, if selected, type of service (ToS).
- IPv6: Version; payload length; next header; hop limit; source address; destination address; routing header; and, if selected, traffic class and/or flow label.
- ICMP: type; code; header checksum; and, if selected, other header fields (varies based on the ICMP type and code).
- ICMPv6: type; code; and header checksum.
- TCP: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options.
- UDP: source port; destination port; length; and UDP checksum.

Section 7.9 of [CCECG] (“Configure Stateful Inspection Filtering”) describes each of the above protocols; how policies are defined; and provides instructions to configure the attributes identified above within packet filtering rules.

Section 7.11.4 of [CCECG] (“Configure L3 & L4 Header Rules”) provides instructions for configuring L3 & L4 Header rules. This section indicates that all protocols and header fields as specified above are supported.

The evaluator shall verify that the operational guidance provides instructions with how to select and/or configure reactions specified in IPS_SBD_EXT.1.5 in the signature rules.

Section 7.11.4 of [CCECG] provides instructions for how to choose the action to take when there is a custom signature match. Options include allow, alert, drop, reset client, reset server, and reset both. Section 7.11.5 of [CCECG] (“Configure Flooding, Reconnaissance, and Attack-Based Protections”) states

© 2024 Leidos. All rights reserved.

some attacks are always dropped and can never be allowed, consistent with the reactions specified in IPS_SBD_EXT.1.5.

IPS_SBD_EXT.1.2
The evaluator shall verify that the operational guidance provides instructions with how to configure rules using the packet payload string-based detection fields defined in IPS_SBD_EXT.1.2.

Section 7.11.6 of [CCECG] ("Configure Vulnerability/Threat Signature Rules") provides instructions for configuring rules using packet payload string-based detection fields and actions defined in the requirement. It states to select context from available custom signature contexts and provides FTP commands and HTTP and SMTP Method Qualifiers. The qualifiers can be added to custom signatures that use related contexts to limit a match. Examples are provided.

The evaluator shall verify that the operational guidance provides instructions with how to configure reactions specified in IPS_SBD_EXT.1.5 for each string-based detection signature.

Section 7.11.4 of [CCECG] provides instructions for how to choose the action to take when there is a custom signature match. Options include allow, alert, drop, reset client, reset server, and reset both.

The evaluator shall verify that the operational guidance provides instructions with how rules are associated with distinct network interfaces that are capable of being associated with signatures.

Section 7.11 of [CCECG] ("Configure Threat Prevention") describes how rules are associated with distinct network interfaces. The TOE groups network interfaces into security zones. A network interface must be assigned to a security zone before it can process traffic. A security zone can have multiple interfaces of the same type (such as Tap, Layer 2, or Layer 3), but an interface can belong to only one zone. The TOE supports three types of rules: security policy rules (i.e., firewall rules); L3 & L4 header rules; and Vulnerability/Threat signature rules. The Security Administrator configures security policy rules (described in section 7.9 of [CCECG]) and assigns them to configured security zones. The TOE applies the security policy rule to traffic that traverses from the configured source security zone to the configured destination security zone. The TOE applies L3 & L4 header rules to packet headers of IPv4, IPv6, ICMPv4, ICMPv6, TCP, and UDP packets, as part of a Zone Protection profile. The TOE applies the header rules to all packets that ae assigned to a security zone in the Zone Protection profile. The Vulnerability/Threat signature rules are for inspecting packet payloads and are applied as part of the security policy rules.

IPS_SBD_EXT.1.3
The evaluator shall verify that the operational guidance provides instructions with configuring rules to identify the attacks defined in IPS_SBD_EXT.1.3 as well as the reactions to these attacks as specified in IPS_SBD_EXT.1.5.

Section 7.11.5 of [CCECG] provides instructions for configuring the TOE to protect against the attacks identified in IPS_SBD_EXT.1.3 as well as the reactions/best practices to these attacks.

© 2024 Leidos. All rights reserved.

IPS_SBD_EXT.1.4

The evaluator shall verify that the operational guidance provides instructions with configuring rules to identify the attacks defined in IPS_SBD_EXT.1.4 as well as the reactions to these attacks as specified in IPS_SBD_EXT.1.5.

Section 7.9.1 of [CCECG] ("Zone Protection Profile") describes how to configure a Zone Protection profile to protect against SYN, ICMP, IMCPv6, UDP, and other IP flood attacks. Section 7.11.5 of [CCECG] provides instructions to configure *Enable Reconnaissance Protection* on all zones to defend against port scans, host sweeps, and IP protocol scan attacks as specified in IPS_SBD_EXT.1.4. The instructions include how to configure actions to these attacks as specified in IPS_SBD_EXT.1.5.

IPS_SBD_EXT.1.5

The guidance EAs for this element are performed in conjunction with IPS_SBD_EXT.1.1, IPS_SBD_EXT.1.3, and IPS_SBD_EXT.1.4.

The guidance EAs for this element are performed in conjunction with IPS_SBD_EXT.1.1, IPS_SBD_EXT.1.3, and IPS_SBD_EXT.1.4.

IPS_SBD_EXT.1.6

The evaluator shall verify that the operational guidance provides configuration instructions, if needed, to detect payload across multiple packets.

Section 6.11 of [ST] states the TOE performs stream reassembly to detect malicious payload split across multiple non-fragmented packets, and this capability does not require any configuration.

2.11.4.3 Test Activities

Modified in accordance with TD0722.

IPS_SBD_EXT.1.1

Test 1: The evaluator shall use the instructions in the operational guidance to test that packet header signatures can be created and/or configured with the selected and/or configured reactions specified in IPS_SBD_EXT.1.5 for each of the attributes listed below. Each attribute shall be individually assigned to its own unique signature:

- IPv4: Version; Header Length; Packet Length; ID; IP Flags; Fragment Offset; Time to Live (TTL); Protocol; Header Checksum; Source Address; Destination Address; IP Options; and, if selected, type of service (ToS).
- IPv6: Version; payload length; next header; hop limit; source address; destination address; routing header; and, if selected, traffic class and/or flow label.
- ICMP: Type; Code; Header Checksum; and, if selected, other Header fields (varies based on the ICMP type and code).
- ICMPv6: Type; Code; and Header Checksum.
- TCP: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options.
- UDP: Source port; destination port; length; and UDP checksum.

The evaluator shall generate traffic to trigger a signature and shall then use a packet sniffer to capture traffic that ensures the reactions of each rule are performed as expected.

© 2024 Leidos. All rights reserved.

The evaluator created signatures for each of the attributes. The evaluator directed packets at the TOE that matched each of the signatures. The evaluator verified that the TOE detected packets that match each of the signatures and that the TOE enforced the configured action on the packets.

Test 2: The evaluator shall repeat the test above to ensure that signature-based IPS policies can be defined for each distinct network interface type capable of applying signatures as supported by the TOE.

The evaluator observed that regardless of the physical interface type used by the TOE, the logical interface is always configured as Ethernet. There are no other types of logical interfaces for which stateful traffic filtering rules can be defined.

IPS_SBD_EXT.1.2

Test 1: The evaluator shall use the instructions in the operational guidance to test that packet payload string-based detection rules can be assigned to the reactions specified in IPS_SBD_EXT.1.5 using the attributes specified in IPS_SBD_EXT.1.2. However it is not required (nor is it feasible) to test all possible strings of protocol data, the evaluator shall ensure that a selection of strings in the requirement is selected to be tested. At a minimum at least one string using each of the following attributes from IPS_SBD_EXT.1.2 should be tested for each protocol. The evaluator shall generate packets that match the string in the rule and observe the corresponding reaction is as configured.

- Test at least one string of characters for ICMPv4 data: beyond the first 4 bytes of the ICMP header.
- Test at least one string of characters for ICMPv6 data: beyond the first 4 bytes of the ICMP header.
- TCP data (characters beyond the 20 byte TCP header):
    i. Test at least one FTP (file transfer) command: help, noop, stat, syst, user, abort, acct, allo, appe, cdup, cwd, dele, list, mkd, mode, nlst, pass, pasv, port, pass, quit, rein, rest, retr, rmd, rnfr, rnto, site, smnt, stor, stou, stru, and type.
    ii. HTTP (web) commands and content:
        (1) Test both GET and POST commands
        (2) Test at least one administrator-defined strings to match URLs/URIs, and web page content.
    iii. Test at least one SMTP (email) state: start state, SMTP commands state, mail header state, mail body state, abort state.
    iv. Test at least one string in any additional attribute type defined within the “other types of TCP payload inspection” assignment, if any other types are specified.
- Test at least one string of UDP data: characters beyond the first 8 bytes of the UDP header;
- Test at least one string for each additional attribute type defined in the “other types of packet payload inspection” assignment, if any other types are specified.

The evaluator created signatures for each of the packet payload fields to be inspected. The evaluator directed packets at the TOE that are otherwise permitted to flow across the TOE. The evaluator verified that the TOE’s IPS functionality detected and blocked the packets which matched the configured payload fields.

Test 2: The evaluator shall repeat Test 1 above to ensure that signature-based IPS policies can be defined for each distinct network interface type capable of applying signatures as supported by the TOE.

© 2024 Leidos. All rights reserved.

The evaluator observed that regardless of the physical interface type used by the TOE, the logical interface is always configured as Ethernet. There are no other types of logical interfaces for which stateful traffic filtering rules can be defined.

IPS_SBD_EXT.1.3

The evaluator shall create and/or configure rules for each attack signature in IPS_SBD_EXT.1.3. For each attack, the TOE should apply its corresponding signature and enable it to each distinct network interface type capable of applying the signatures. The evaluator shall use packet captures to ensure that the attack traffic is detected by the TOE and a reaction specified in IPS_SBD_EXT.1.5 is triggered and stops the attack. Each attack should be performed one after another so as to ensure that its corresponding signature successfully identified and appropriately reacted to a particular attack.

The evaluator configured signatures for each attack specified in IPS_SBD_EXT.1.3. The evaluator directed traffic matching each of the attack signatures at the TOE. The evaluator verified that the TOE detected and responded to each of the attack signatures specified in IPS_SBD_EXT.1.3.

IPS_SBD_EXT.1.4

The evaluator shall configure individual signatures for each attack in IPS_SBD_EXT.1.4. For each attack, the TOE should apply its corresponding signature and enable it to each distinct network interface type capable of applying signatures. The evaluator shall use packet captures to ensure that the attack traffic is detected by the TOE and a reaction specified in IPS_SBD_EXT.1.5 is triggered and stops the attack. Each attack should be performed one after another so as to ensure that its corresponding signature successfully identified and appropriately reacted to a particular attack.

The evaluator configured signatures for each attack specified in IPS_SBD_EXT.1.4. The evaluator directed traffic matching each of the attack signatures at the TOE. The evaluator verified that the TOE detected and responded to each of the attack signatures specified in IPS_SBD_EXT.1.4.

IPS_SBD_EXT.1.5

The test EAs for this element are performed in conjunction with those for IPS_SBD_EXT.1.1, IPS_SBD_EXT.1.2, IPS_SBD_EXT.1.3, and IPS_SBD_EXT.1.4.

Per the assurance activity this has been performed in conjunction with IPS_SBD_EXT.1.1, IPS_SBD_EXT.1.2, IPS_SBD_EXT.1.3, and IPS_SBD_EXT.1.4.

IPS_SBD_EXT.1.6

The evaluator shall repeat one of the tests in IPS_SBD_EXT.1.2 Test 1 but generate multiple non-fragmented packets that contain the string in the rule defined. The evaluator shall verify that the malicious traffic is still detected when split across multiple non-fragmented packets.

The evaluator took traffic that was used for IPS_SBD_EXT.1.2 Test 1 and split the payload into two non-fragmented packets. The evaluator directed the traffic at the TOE and verified that the TOE enforced the signature on the packets even when the payload is split across two non-fragmented packets.

© 2024 Leidos. All rights reserved.

3 Security Assurance Requirements

3.1 Class ASE: Security Targeted Evaluation

General ASE

When evaluating a Security Target, the evaluator performs the work units as presented in the CEM. In addition, the evaluator shall ensure the content of the TSS in the ST satisfies the EAs specified in Section 2 (Evaluation Activities for SFRs).

3.1.1 ASE_TSS.1 TOE Summary Specification for Distributed TOEs

This section is N/A for this evaluation because the TOE is not distributed.

3.2 Class ADV: Development

3.2.1 ADV_FSP.1 Basic Functional Specification

The EAs for this assurance component focus on understanding the interfaces (e.g., application programming interfaces, command line interfaces, graphical user interfaces, network interfaces) described in the AGD documentation, and possibly identified in the TOE Summary Specification (TSS) in response to the SFRs. Specific evaluator actions to be performed against this documentation are identified (where relevant) for each SFR in Section 2, and in EAs for AGD, ATE and AVA SARs in other parts of Section 5.

The EAs presented in this section address the CEM work units ADV_FSP.1-1, ADV_FSP.1-2, ADV_FSP.1-3, and ADV_FSP.1-5.

The EAs are reworded for clarity and interpret the CEM work units such that they will result in more objective and repeatable actions by the evaluator. The EAs in this SD are intended to ensure the evaluators are consistently performing equivalent actions.

The documents to be examined for this assurance component in an evaluation are therefore the Security Target, AGD documentation, and any required supplementary information required by the cPP: no additional “functional specification” documentation is necessary to satisfy the EAs. The interfaces that need to be evaluated are also identified by reference to the EAs listed for each SFR and are expected to be identified in the context of the Security Target, AGD documentation, and any required supplementary information defined in the cPP rather than as a separate list specifically for the purposes of CC evaluation. The direct identification of documentation requirements and their assessment as part of the EAs for each SFR also means that the tracing required in ADV_FSP.1.2D (work units ADV_FSP.1-4, ADV_FSP.1-6 and ADV_FSP.1-7) is treated as implicit and no separate mapping information is required for this element.

3.2.1.1 ADV_FSP.1 Evaluation Activity

The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.

In this context, TSFI are deemed security relevant if they are used by the administrator to configure the TOE, or to perform other administrative functions (e.g. audit review or performing updates). Explicitly labeling TSFI as security relevant or non-security relevant is not necessary. A TSFI is implicitly security relevant if it is used to satisfy an evaluation activity, or if it is identified in the ST or guidance documentation as adhering to the security policies (as presented in the SFRs). The intent is that these

© 2024 Leidos. All rights reserved.

interfaces will be adequately tested and having an understanding of how these interfaces are used in the TOE is necessary to ensure proper test coverage is applied. According to the description above 'security relevant' corresponds to the combination of 'SFR-enforcing' and 'SFR-supporting' as defined in CC Part 3, paragraph 224 and 225.

The set of TSFI that are provided as evaluation evidence are contained in the Security Target and the guidance documentation.

Section 2.2.1 of [ST] identifies the security relevant TSFIs as remote syslog server, VPN peer, Palo Alto Network Global Protect, and workstation (SSH/HTTPS with optional IPsec). The TSS describes these logical interfaces as TLS, IPsec trusted channels and SSH/HTTPS, IPsec trusted paths and defines their operation in terms of the relevant FCS and FTP requirements. Section 2.2.1 of [ST] also defines the external physical interfaces of the TOE in sufficient detail to determine their security relevance (e.g., noting that USB ports are disabled in FIPS-CC mode except to provide power and that the Micro USB Console only provides self-test output while in FIPS-CC mode).

3.2.1.2 ADV_FSP.1 Evaluation Activity

The evaluator shall check the interface documentation to ensure it identifies and describes the parameters for each TSFI that is identified as being security relevant.

The vendor developed [CCECG] specifically to address the security functionality as identified in [ST]. The Guidance Activities for the individual SFRs demonstrate that the guidance documentation includes sufficiently detailed instructions to configure and use the TSFIs in the manner required by [ST]. This is demonstrated by the evaluation activities for FCS_SSH_EXT.1, FCS_SSHS_EXT.1, FCS_HTTPS_EXT.1, FCS_IPSEC_EXT.1, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, and FMT_MTD.1/CoreData.

3.2.1.3 ADV_FSP.1 Evaluation Activity

The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.

The evaluator shall use the provided documentation and first identifies, and then examines a representative set of interfaces to perform the EAs presented in Section 2, including the EAs associated with testing of the interfaces.

It should be noted that there may be some SFRs that do not have a TSFI that is explicitly “mapped” to invoke the desired functionality. For example, generating a random bit string or destroying a cryptographic key that is no longer needed are capabilities that may be specified in SFRs, but are not invoked by an interface.

The required EAs define the design and interface information required to meet ADV_FSP.1. If the evaluator is unable to perform some EA, then the evaluator shall conclude that an adequate functional specification has not been provided.

Section 6.8 of [ST] associates the TOE’s remote logical interfaces with the FTP_ITC.1 and FTP_TRP.1/Admin SFRs. Additionally, this section identifies the trusted channel protocol and which end of the connection the TOE is (client or server) as needed to specifically associate each logical interface further with FCS_HTTPS_EXT.1, FCS_IPSEC_EXT.1, FCS_SSH_EXT.1, FCS_SSHS_EXT.1, FCS_TLSC_EXT.1, and FCS_TLSS_EXT.1, as appropriate.

Additionally, the intended usage of each logical interface can be inferred from the TSS to the extent that their applicability to other SFRs can be determined. Specifically, the syslog interface is also used in support

© 2024 Leidos. All rights reserved.

of FAU_STG_EXT.1 and the remote management interface is used to enforce the various FMT requirements and supports the enforcement of the various FIA and FTA requirements through its usage.

3.3 Class AGD: Guidance Documents

It is not necessary for a TOE to provide separate documentation to meet the individual requirements of AGD_OPE and AGD_PRE. Although the EAs in this section are described under the traditionally separate AGD families, the mapping between the documentation provided by the developer and AGD_OPE and AGD_PRE requirements may be many-to-many, as long as all requirements are met in documentation that is delivered to Security Administrators and users (as appropriate) as part of the TOE.

Note that additional Evaluation Activities for the guidance documentation in the case of a distributed TOE are defined in Appendix B.4.2.1.

3.3.1 AGD_OPE.1 Operational User Guidance

The evaluator performs the CEM work units associated with the AGD_OPE.1 SAR. Specific requirements and EAs on the guidance documentation are identified (where relevant) in the individual EAs for each SFR. For the related evaluation activities, the evaluation evidence documents Security Target, AGD documentation (user guidance) and functional specification documentation (if provided) shall be used as input documents. Each input document is subject to ALC_CMS.1-2 requirements.

In addition, the evaluator performs the EAs specified below.

3.3.1.1 AGD_OPE.1 Evaluation Activity

The evaluator shall ensure the Operational guidance documentation is distributed to Security Administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that Security Administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

The [CCECG] is published with the Security Target at the https://www.niap-ccevs.org/ website. Section 1.3 of [CCECG] (“Documentation References”) lists other documentation (comprising [ADMIN], [API], [CLI], [GUI], and [VM]) referenced from [CCECG] that the vendor publishes on its web site. This includes URLs for each document. The evaluator verified the URLs link to the correct documents on the vendor web site. The distribution of the documentation provides a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

3.3.1.2 AGD_OPE.1 Evaluation Activity

The evaluator shall ensure that the Operational guidance is provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

Section 1.2 of [CCECG] identifies all of the platforms claimed for the TOE as identified in [ST].

Section 2.2 of [CCECG] identifies the specific conditions that are expected to be met by the operational environment and/or administrators. Section 2.1 identifies the components in the operational environment.

© 2024 Leidos. All rights reserved.

[CCECG] identifies the supported TOE version as 11.1. It also identifies the supported virtual machine platforms.

3.3.1.3 AGD_OPE.1 Evaluation Activity

The evaluator shall ensure that the Operational guidance contains instructions for configuring any cryptographic implementation associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic implementations was not evaluated nor tested during the CC evaluation of the TOE.

Section 6.2 of [CCECG] provides instructions for configuring the cryptographic engine by changing the operational mode of the TOE from normal to FIPS-CC mode. It also warns the administrator the operational mode must be FIPS-CC mode in order for the TOE to be in its evaluated configuration.

3.3.1.4 AGD_OPE.1 Evaluation Activity

The evaluator shall ensure the Operational guidance makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs.

The Scope of Evaluation section in section 1.1 of [CCECG] lists the functionality that was excluded from evaluation and makes it explicitly clear that only the functionality claimed in [ST] was evaluated.

3.3.1.5 AGD_OPE.1 Evaluation Activity

In addition, the evaluator shall ensure that the following requirements are also met.

a. The guidance documentation shall contain instructions for configuring any cryptographic implementation associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic implementations was not evaluated nor tested during the CC evaluation of the TOE.
b. The evaluator shall verify that this process includes instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).
c. The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The guidance documentation shall make it clear to an administrator which security functionality is covered by the Evaluation Activities.

Part a) is addressed by section 3.3.1.3 above.

For part b), section 7.12 of [CCECG] describes the update process. Specifically, administrators use the TOE to check for updates made available on the Palo Alto support site. The ‘request system software download’ command is used to acquire an update. Once downloaded, the ‘request system software install’ command initiates the update process, which automatically checks the validity of the digital signature. This section notes the TOE’s behavior in the event of an update failure.

Part c) is addressed by section 3.3.1.4 above.

© 2024 Leidos. All rights reserved.

3.3.2 AGD_PRE.1 Preparative Procedures

The evaluator performs the CEM work units associated with the AGD_PRE.1 SAR. Specific requirements and EAs on the preparative documentation are identified (and where relevant are captured in the Guidance Documentation portions of the EAs) in the individual EAs for each SFR.

Preparative procedures are distributed to Security Administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that Security Administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

In addition, the evaluator performs the EAs specified below.

3.3.2.1 AGD_PRE.1 Evaluation Activity

The evaluator shall examine the Preparative procedures to ensure they include a description of how the Security Administrator verifies that the operational environment can fulfil its role to support the security functionality (including the requirements of the Security Objectives for the Operational Environment specified in the Security Target).

The documentation should be in an informal style and should be written with sufficient detail and explanation that they can be understood and used by the target audience (which will typically include IT staff who have general IT experience but not necessarily experience with the TOE product itself).

Section 2.2 of [CCECG] identifies the assumptions that state the specific conditions that are expected to be met by the operational environment and/or administrators. Section 1.1 describes the scope of evaluation, required configuration, assumptions, and operational environment that the system must be in to ensure a secure deployment. Section 3 of [CCECG] lists various items for consideration by the administrator prior to installing the TOE in its evaluated configuration. These sections are written in an informal style and provide sufficient detail and explanation that they can be understood and used by the target audience.

3.3.2.2 AGD_PRE.1 Evaluation Activity

The evaluator shall examine the preparative procedures to ensure they are provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

Section 1.2 of [CCECG] identifies all TOE models and platforms as claimed for the TOE in the security target. The instructions and guidance contained in the [CCECG] are applicable to all TOE platforms.

3.3.2.3 AGD_PRE.1 Evaluation Activity

The evaluator shall examine the preparative procedures to ensure they include instructions to successfully install the TSF in each Operational Environment.

The evaluators reviewed [Admin] and [CCECG] and determined that they describe how set up the TOE's logical interfaces for initial use and to configure the external interfaces specified as the TOE's Operational Environment by [ST]. The evaluators reviewed [VM], which provides instructions to install VM-Series virtual appliances in each of the evaluated virtual environments.

The instructions and guidance contained in the [CCECG] is applicable to all TOE platforms.

© 2024 Leidos. All rights reserved.

3.3.2.4 AGD_PRE.1 Evaluation Activity

The evaluator shall examine the preparative procedures to ensure they include instructions on how to manage the TSF as a product and as a component of the larger Operational Environment in a manner that allows to preserve integrity of the TSF.

The intent of this requirement is to ensure there exists adequate preparative procedures (guidance in most cases) to put the TSF in a secure state (i.e., evaluated configuration). AGD_PRE.1 lists general requirements, the specific assurance activities implementing it are performed as part of FMT_SMF.1, FMT_MTD.1 and FMT_MOF.1 series of SFRs.

The evaluators observed that the TOE's documentation includes guidance on configuring the TOE's interactions with its operational environment where needed. This includes setting up trusted channels to the TOE's Operational Environment even though some of the data carried over those channels is outside the scope of [NDcPP]. This ensures that the TOE can be deployed properly in its intended context while being configured in a secure manner as claimed by [ST].

3.3.2.5 AGD_PRE.1 Evaluation Activity

In addition the evaluator shall ensure that the following requirements are also met.

The preparative procedures must

a. include instructions to provide a protected administrative capability; and
b. identify TOE passwords that have default values associated with them and instructions shall be provided for how these can be changed.

The evaluators reviewed [CCECG] and observed that the CLI uses SSH and HTTPS by default. However, [CCECG] also includes instructions for enabling CC-FIPS mode, which ensures that SSH and TLS are configured in the manner claimed by [ST]. It also includes instructions for additional configuration steps to ensure that the SSH configuration is consistent with [ST].

The evaluators reviewed [CCECG] and observed that section 6.3 states that the admin account's default password is 'paloalto' along with instructions for how to change this.

3.4 Class ALC: Life-Cycle Support

3.4.1 ALC_CMC.1 Labeling of the TOE

When evaluating that the TOE has been provided and is labelled with a unique reference, the evaluator performs the work units as presented in the CEM.

The evaluation team verified this through the completion of the ALC_CMC.1 work units described in the CEM. The results of this analysis are included in the proprietary ETR produced by the laboratory.

3.4.2 ALC_CMS.1 TOE CM Coverage

When evaluating the developer's coverage of the TOE in their CM system, the evaluator performs the work units as presented in the CEM.

The evaluation team verified this through the completion of the ALC_CMS.1 work units described in the CEM. The results of this analysis were included in the proprietary ETR produced by the laboratory.

© 2024 Leidos. All rights reserved.

3.4.3 ALC_FLR.3 Systematic Flaw Remediation

When evaluating the developer's procedures regarding systematic flaw remediation, the evaluator performs the work units as presented in the CEM.

The evaluation team verified this through the completion of the ALC_FLR.3 work units described in the CEM. The results of this analysis were included in the proprietary ETR produced by the laboratory.

3.5 Class ATE: Tests

3.5.1 ATE_IND.1 Independent Testing – Conformance

The focus of the testing is to confirm that the requirements specified in the SFRs are being met. Additionally, testing is performed to confirm the functionality described in the TSS, as well as the dependencies on the Operational guidance documentation is accurate.

The evaluator performs the CEM work units associated with the ATE_IND.1 SAR. Specific testing requirements and EAs are captured for each SFR in Sections 2, 3 and 4.

The evaluator shall consult Appendix B when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation.

Note that additional Evaluation Activities relating to evaluator testing in the case of a distributed TOE are defined in section B.4.3.1.

The evaluators developed a test plan ([Test]) to list all of the individual test evaluation activities for the TOE based on the claimed SFRs. The test plan lists, for each evaluation activity, the platform(s) the test was executed on, the test results (including supporting evidence), and the testing verdict. In all cases, the tests were observed to be passing.

The TOE consists of multiple variants. The evaluation lab provided a detailed test report which contains an equivalency argument that discusses which variants were tested, and why this subset can reasonably be expected to cover the full set of variants covered by the evaluation.

Specifically, all functional tests were conducted six times: once on each supported VM hypervisor, and once on the PA-3260, PA-5430, and PA-5450 hardware platforms. Equivalency between VM hypervisors could not be argued so testing was repeated on each to show equivalent behavior. For the hardware platforms, the firmware and security-relevant external interfaces are identical between models. The primary differences are the processors used by each hardware model. However, the only security-relevant impact to processor differences is handling of cryptographic algorithm primitives. The certificate worksheet in the proprietary Evaluation Technical Report shows how every claimed TOE model was tested, either through an exact match with the DP and MP processor used, or through an equivalency argument where the processor used by that model is the same microarchitecture as one that was tested.

Testing of the TOE was done using a variety of tools. OpenSSL's s_server and s_client functions were used for TLS testing, along with a Leidos proprietary TLS packet modification tool. Both standard and modified versions of Strongswan were used for IPsec testing. Proprietary Leidos python scripts were used for fuzz testing and firewall tests. All test connections were captured using Wireshark. Exported TOE audit records were sent to a server running rsyslogd.

© 2024 Leidos. All rights reserved.

As stated in the ST, the physical boundary of the TOE comprises the hardware firewall appliances, grouped by their model series:

1. PA-400 Series
    a. PA-410
    b. PA-410R-5G
    c. PA-415
    d. PA-415-5G
    e. PA-440
    d. PA-445
    e.PA-450R-5G
    f. PA-450
    g. PA-450R
    h. PA-455
    i. PA-460
2. PA-800 Series
    a. PA-820
    b. PA-850
3. PA-1400 Series
    a. PA-1410
    b. PA-1420
4. PA-3200 Series
    a. PA-3220
    b. PA-3250
    c. PA-3260
5. PA-3400 Series
    a. PA-3410
    b. PA-3420
    c. PA-3430
    d. PA-3440
6. PA-5200 Series
    a. PA-5220
    b. PA-5250
    c. PA-5260
    d. PA-5280
7. PA-5400 Series
    a. PA-5410
    b. PA-5420
    c. PA-5430
    d. PA-5440
    e. PA-5445
8. PA-5450
9. PA-7000 Series
    a. PA-7050
    b. PA-7080
    c. PA-7500

© 2024 Leidos. All rights reserved.

All TOE instances provide the same security functionality.

The devices in groups 1 through 7 are all physical devices with fixed interfaces. Thus, the only differences between the models in these groups are the number of ports supported by the device, the maximum supported throughput which can be supported by the device. All of the devices have ACVP certificates for the respective underlying processor and thus proves that the cryptographic functionality will accurately work on all of the devices. Therefore, the differences between the devices in these groups are not security relevant and the testing which is performed on the PA-3260 is considered sufficient to ensure the security functionality of all devices covered in the devices of groups 1 through 7.

The PA-5450 and the PA-7000 series devices, groups 8 and 9, are different from the devices in groups 1 through 7 as these devices possess modular blades which can be utilized to customize the ports. Specifically, the number of available ethernet and type of port (RJ-45, the SFP, the SFP+, and the QSFP) which are present in the device. As the only security relevant difference between these blades would be the processor which powers the specific blade and there are ACVP certificates for all of the 'Data plane' processors, which are the processors which power the blades. The testing on the PA-5450 is considered sufficient to ensure the security functionality of all of the devices in groups 8 and 9.

All VM series models are built from the same PAN-OS source code. The software package (.xva, .ova, or .vhdx file) that is used to deploy the VM-Series firewall is common across all models. All VM-series have the same software image, designed to interoperate with all supported hypervisors. This base image is packaged into a single RAW disk image. This common disk image is converted into a compatible format for each hypervisor (i.e. ova, qcow2, and xva). When the customer applies the capacity license on the VM-Series firewall, the model number and the associated capacities are implemented on the firewall. Capacity is defined in terms of the number of sessions, rules, security zones, address objects, IPSec VPN tunnels, and SSL VPN tunnels that the VM-Series firewall is optimized to handle. The support site for Palo Alto Networks then distributes the formatted disk images for deployment. One of each of the Hypervisors was tested during the evaluation.

The evaluated configuration of the TOE covered Microsoft Hyper-V running on Server 2019. This is considered equivalent to Hyper-V running on Server 2016 assuming appropriate algorithm certificates are present. Similarly, the evaluated configuration of the TOE covered KVM 4 running on Ubuntu 20.04, which is considered equivalent to Ubuntu 18.04 with the same assumption.

© 2024 Leidos. All rights reserved.

3.6 Class AVA: Vulnerability Assessment

3.6.1 AVA_VAN.1 Vulnerability Survey

While vulnerability analysis is inherently a subjective activity, a minimum level of analysis can be defined and some measure of objectivity and repeatability (or at least comparability) can be imposed on the vulnerability analysis process. In order to achieve such objectivity and repeatability it is important that the evaluator shall follow a set of well-defined activities and documents their findings so others can follow their arguments and come to the same conclusions as the evaluator. While this does not guarantee that different evaluation facilities will identify exactly the same type of vulnerabilities or come to exactly the same conclusions, the approach defines the minimum level of analysis and the scope of that analysis and provides Certification Bodies a measure of assurance that the minimum level of analysis is being performed by the evaluation facilities.

In order to meet these goals some refinement of the AVA_VAN.1 CEM work units is needed. The following table indicates, for each work unit in AVA_VAN.1, whether the CEM work unit is to be performed as written, or if it has been clarified by an Evaluation Activity. If clarification has been provided, a reference to this clarification is provided in the table.

Because of the level of detail required for the evaluation activities, the bulk of the instructions are contained in Appendix A, while an "outline" of the assurance activity is provided below.

From the [Firewall SD]

Type 1 Hypotheses – Public-Vulnerability-Based

The list of public sources of vulnerability information selected by the iTC is given in Section A.4 of [SD-ND]. Any additional sources specifically for firewalls will be specified in chapter A.4 of this document.

The evaluators shall perform a search on the sources listed in Section A.4 of [SD-ND] to determine a list of potential flaw hypotheses that are more recent that the publication date of the PP-Module, and those that are specific to the TOE and its components as specified by the additional documentation mentioned above. Any duplicates – either in a specific entry, or in the flaw hypothesis that is generated from an entry from the same or a different source – can be noted and removed from consideration by the evaluation team.

The search criteria to be used when searching the sources published after the publication date of the cPP shall include:

- The term "firewall"
- The following protocols: TCP, UDP, IPv4, IPv6
- Any protocols not listed above supported (through an SFR) by the TOE.
- The TOE name (including appropriate model information as appropriate)

As part of type 1 flaw hypothesis generation for the specific components of the TOE, the evaluator shall also search the component manufacturer's websites to determine if flaw hypotheses can be generated on this basis (for instance, if security patches have been released for the version of the component being evaluated, the subject of those patches may form the basis for a flaw hypothesis).

Type 2 Hypotheses – iTC-Sourced

Section A.5 of [SD-ND] contains the list of flaw hypothesis generated by the iTC for this technology that must be considered by the evaluation team as flaw hypotheses in performing the vulnerability

© 2024 Leidos. All rights reserved.

assessment. Section A.5 of this document contains additional flaw hypothesis generated by the iTC specifically for firewalls.

If the evaluators discover a Type 3 or Type 4 flaw that they believe should be considered as a Type 2 flaw in future versions of this PP-Module, they should work with their Certification Body to determine the appropriate means of submitting the flaw for consideration by the iTC.

Type 3 Hypotheses – Evaluation-Team-Generated

Type 3 flaws are formulated by the evaluator based on information presented by the product (through on-line help, product documentation and user guides, etc.) and product behaviour during the (functional) testing activities. The evaluator is also free to formulate flaws that are based on material that is not part of the baseline evidence (e.g., information gleaned from an Internet mailing list, or reading interface documentation on interfaces not included in the set provided by the developer), although such activities have the potential to vary significantly based upon the product and evaluation facility performing the analysis.

If the evaluators discover a Type 3 flaw that they believe should be considered as a Type 2 flaw in future versions of this PP-Module, they should work with their Certification Body to determine the appropriate means of submitting the flaw for consideration by the iTC.

Type 4 Hypotheses – Tool-Generated

There are no Type 4 hypotheses that apply to the TOE beyond those defined by [SD-ND].

If the evaluators discover a Type 4 flaw that they believe should be considered as a Type 2 flaw in future versions of this PP-Module, they should work with their Certification Body to determine the appropriate means of submitting the flaw for consideration by the iTC.

The results of these activities from the [FW-SD] are included in the discussion below and documented in [VA].

3.6.1.1 AVA_VAN.1 Evaluation Activity (Documentation)

In addition to the activities specified by the CEM in accordance with Table 2, the evaluator shall perform the following activities.

The evaluator shall examine the documentation outlined below provided by the developer to confirm that it contains all required information. This documentation is in addition to the documentation already required to be supplied in response to the EAs listed previously.

The developer shall provide documentation identifying the list of software and hardware components that compose the TOE. Hardware components should identify compute-capable hardware components, at a minimum that must include the processor, and where applicable, discrete crypto ASICs, TPMs, etc. used by the TOE. Software components include applications, the operating system and other major components that are independently identifiable and reusable (outside of the TOE), for example a web server, protocol or cryptographic implementations, (independently identifiable and reusable components are not limited to the list provided in the example). This additional documentation is merely a list of the name and version number of the components and will be used by the evaluators in formulating vulnerability hypotheses during their analysis.

As per [ST] and the evidence used for the evaluation of AGD_OPE.1 and AGD_PRE.1, the evaluators identified the following materials:

- Processors used by the TOE: identified in [ST] Section 2.2.1.
- Software components used by the TOE: identified in section 2.2.1 of [ST].

© 2024 Leidos. All rights reserved.

- Hardware models: identified in section 2.2.1 of [ST].
- Materials related to distributed TOE requirements are N/A because the TOE is not distributed.

If the TOE is a distributed TOE then the developer shall provide:

a. documentation describing the allocation of requirements between distributed TOE components as in [NDcPP, 3.4]

b. a mapping of the auditable events recorded by each distributed TOE component as in [NDcPP, Table 2]

c. additional information in the Preparative Procedures as identified in the refinement of AGD_PRE.1 in additional information in the Preparative Procedures as identified in 3.4.1.2 and 3.5.1.2.

The TOE is not distributed.

3.6.1.2 AVA_VAN.1 Evaluation Activity

The evaluator shall formulate hypotheses in accordance with process defined in Appendix A. The evaluator shall document the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in Appendix A.3. The evaluator shall perform vulnerability analysis in accordance with Appendix A.2. The results of the analysis shall be documented in the report according to Appendix A.3.

The evaluators conducted vulnerability research and penetration testing to determine the vulnerability of the TSF to attackers with Basic Attack Potential.

The evaluators conducted searches in public vulnerability repositories for the following Type 1 flaws based on the guidance specified in [ND-SD] and [FW-SD]:

- The list of software and hardware components that comprise the TOE
- The term “firewall”
- The following protocols: TCP, UDP, IPv4, IPv6
- Any protocols not listed above supported (through an SFR) by the TOE
- The TOE name (including model information as appropriate)

These search criteria were applied as follows:

- The list of software and hardware components that comprise the TOE:
    - Processor:
        - AMD EPYC 7352
        - AMD EPYC 7452
        - AMD EPYC 7642
        - AMD EPYC 7742
        - AMD EPYC 7003
        - Cavium Octeon CN7130
        - Cavium Octeon CN7240
        - Cavium Octeon CN7350
        - Cavium Octeon CN7360
        - Cavium Octeon CN7885
        - Cavium Octeon CN7890
        - Intel Atom C3436L

© 2024 Leidos. All rights reserved.

        - Intel Atom C3558R
        - Intel Atom C3708
        - Intel Atom C3758R
        - Intel Atom C5325
        - Intel Atom C5335C1
        - Intel Atom P5332
        - Intel Atom P5342
        - Intel Atom P5352
        - Intel Atom P5362
        - Intel Atom P5752
        - Intel Pentium D1517
        - Intel Xeon D1548
        - Intel Xeon D1567
        - Intel Xeon D-2187NT
        - Intel D-2798NX
        - Intel Xeon Gold 6248
    - The processors encompass the following microarchitectures:
        - MIPS64
        - Zen 2
        - Zen 3
        - Denverton
        - Tremont
        - Skylake
        - Ice Lake
        - Broadwell
    - Software:
        - PAN-OS 11.1
        - NGINX (note, the vendor considers the specific version number used within the TOE to be proprietary information—the version number was provided to the evaluation team and used in the search).
- “Palo Alto Firewall”, “Palo Alto Networks Firewall”, and “PA-400 Series”, “PA-800 Series”, “PA-1400 Series”, “PA-3200 Series”, “PA-3400 Series”, “PA-5200 Series”, “PA-5400 Series”, “PA-5450”, and “PA-7000 Series” as variations of the TOE name (and which additionally include the term “firewall”)
- Protocols:
    - TCP
    - UDP
    - IPv4
    - IPv6
    - TLS
    - SSH
    - HTTPS
    - IPsec.

The evaluation team performed a search of the following public vulnerability databases:

- National Vulnerability Database (https://nvd.nist.gov/).

© 2024 Leidos. All rights reserved.

- US-Cert (https://www.kb.cert.org/vuls/html/search)
- Tipping Point Zero Day Initiative (https://www.zerodayinitiative.com/advisories/published/)
- Palo Alto Networks Security Advisories (https://security.paloaltonetworks.com/).

The evaluators performed these searches several times, most recently on September 17, 2024.

Additionally, the evaluators performed fuzz testing of the TOE as specified in Section A.1.4 of [ND-SD]. The evaluators observed the TOE did not react adversely to the packets directed at the TOE or respond to the packets. This testing did not discover any vulnerabilities in the TOE. The fuzz testing evidence is included in section 11 of the proprietary test report ([Test]).

The evaluation team determined that no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential. This information is documented in [VA].

© 2024 Leidos. All rights reserved.