# Check Point Software Technologies Ltd. Security Gateway and Maestro Hyperscale Appliances R81.00 Security Target

Version 0.5
March 16, 2022

***Prepared for:***

**Check Point Software Technologies Ltd.**

Shlomo Kaplan St 5, Tel Aviv-Yafo, 6789159, Israel

***Prepared By:***

Gossamer Laboratories

www.gossamersec.com

## LIST OF TABLES

# 1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is Security Gateway and Maestro Hyperscale Appliances R81.00 provided by Check Point Software Technologies Ltd. The TOE is being evaluated as a network device (and VPN gateway and firewall).

The Security Target contains the following additional sections:

- Conformance Claims (Section 2)
- Security Objectives (Section 3)
- Extended Components Definition (Section 4)
- Security Requirements (Section 5)
- TOE Summary Specification (Section 6)

***Conventions***

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a parenthetical number placed at the end of the component. For example FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

## 1.1 Security Target Reference

**ST Title –** Check Point Software Technologies Ltd. Security Gateway and Maestro Hyperscale Appliances R81.00 Security Target

**ST Version** – Version 0.5

**ST Date** – March 16, 2022

## 1.2 TOE Reference

**TOE Identification** – Check Point Software Technologies Ltd. Security Gateway and Maestro Hyperscale Appliances

**TOE Developer** – Check Point Software Technologies Ltd.

**Evaluation Sponsor** – Check Point Software Technologies Ltd.

## 1.3 TOE Overview

The Target of Evaluation (TOE) is Security Gateway and Maestro Hyperscale Appliances R81.00.

The Target of Evaluation (TOE) is Check Point Software Security Gateway and Maestro Hyperscale Appliances running software version R81.00. Throughout the remainder of this document the Security Gateway appliances and the Maestro Hyperscale appliances are collectively referred to as "Gateways" or "Gateway appliances". The product family is a set of VPN Gateway and packet filtering firewall appliances, a management appliance, and management software. The product provides controlled connectivity between two or more network environments. It mediates information flows between clients and servers located on internal and external networks governed by the firewalls.

## 1.4 TOE Description

Check Point Gateway appliances provide a broad range of services, features and capabilities. This ST makes a set of claims regarding the product's security functionality, in the context of an evaluated configuration. The claimed security functionality is a subset of the product's full functionality. The evaluated configuration is a subset of the possible configurations of the product, established according to the evaluated configuration guidance. This part of the ST describes the physical and logical scope and boundaries of the Target of Evaluation (TOE).

### 1.4.1 TOE Architecture

The TOE consists of a family of network appliances whose primary function is to provide firewall capabilities for filtering traffic based on packet rules. The TOE is a distributed system with support for a security management server, allowing remote administration over a protected IPsec connection. The TOE includes the following distributed components:

- a Security Management Server (labelled "Mgmt SW" in the figure below) and
- one or more Check Point Gateway Appliances (Hardware appliances and virtual)

The administrator also uses the SmartConsole Management software client version R81.00 (running on one or more administrative workstations) to manage the system.

All products run Check Point version R81.00 software. See Section 7 for a list of Hardware models (both physical and virtual) and processors.

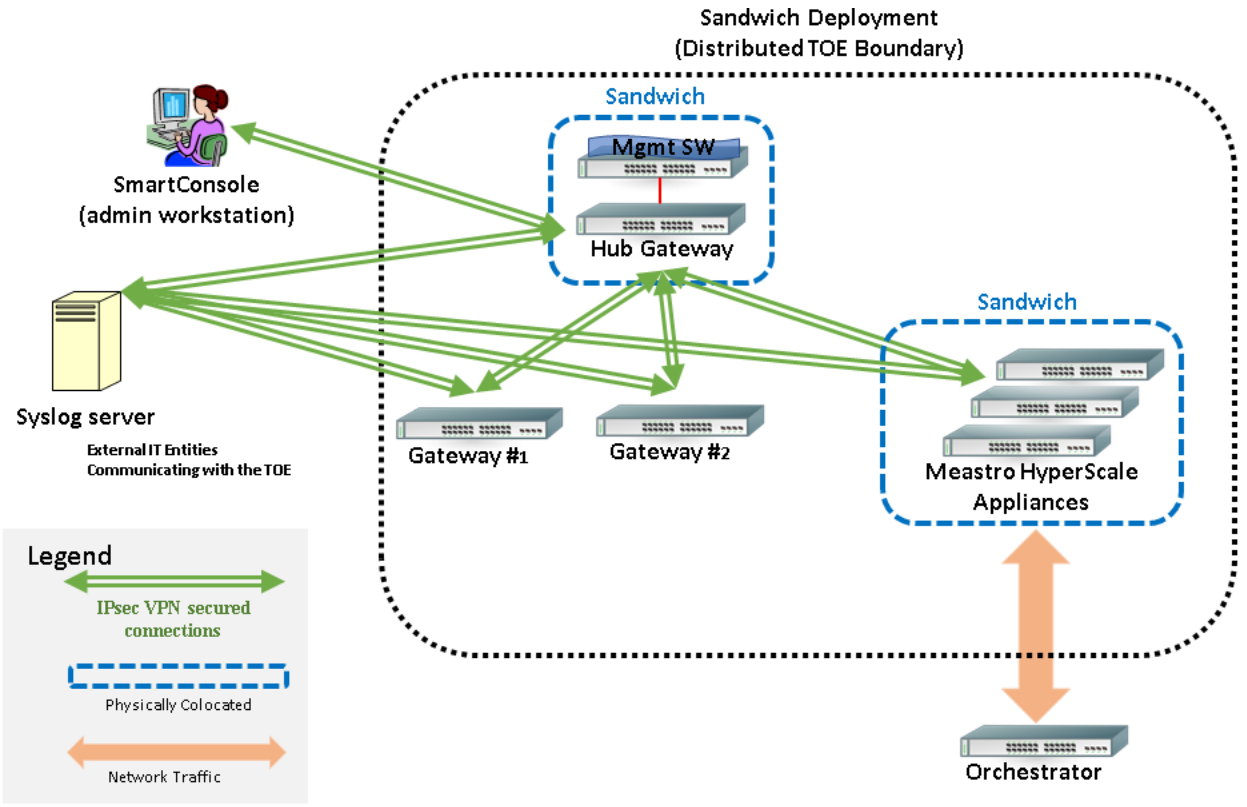


**Figure 1: Distributed TOE Boundaries**

The administrator deploys the TOE (as diagrammed in the figure above) with a Security Management Server appliance physically combined with its "Hub" Gateway. Through Hub gateway, the Security Management Server controls other Gateway appliances enrolled into the TOE.

In the evaluated configuration, the administrator co-locates the Security Management Server and its Hub Security Gateway (for example, racked together) and then uses a direct network connection to join the Security Management Server appliance to its Hub Gateway. Once joined in this (or functionally similar) fashion, the administrator then accesses the Security Management Server through its Hub Gateway.

To avoid confusion between Check Point lexicon and that of the NDcPP22e/VPNGW11/STFFW14e protection profiles, this ST refers to the evaluated configuration as a "Sandwich." Thus, an evaluated configuration consists of a Sandwich ("Sandwich") Deployment with an arbitrary number of additional Gateways managed by the Sandwich. The Sandwich deployment qualifies as a distributed TOE and relies upon IPsec VPN connections to secure both Internal (Intra) TOE Transfers and also rely upon IPsec VPN connections to secure both communications with external TOE entities (e.g., a syslog server) and communications with remote administrative sessions.

Maestro Hyperscale appliances are used with a Maestro Orchestrator to provide load balancing of network traffic. The Maestro Hyperscale appliances have all of the capabilities of Security Gateways including firewall and VPN GW capabilities. The Maestro Orchestrator provides load balancing of traffic prior to it arriving at Maestro Hyperscale appliances. This load balancing does not impact the security operation of the Maestro Hyperscale appliances. In the evaluated configuration, one is typically co-located (i.e., in the same rack) and utilizes a direct network connection to join the Maestro Orchestrator and individual Maestro Hyperscale appliances in order to maximize traffic throughput. The Maestro Orchestrator and its load balancing capabilities, are not part of the TOE and cannot impact security features of the TOE. The Maestro Orchestrator is part of the TOE's Operational Environment but is simply the switch which connects the TOE to another network community much like the neighbor switch to any network device.

As mentioned above, all the products run software version R81.00 and the Gateways and Management server use the same image (note that once installed, the Management Server lacks IPsec and Firewall functionality) and contain the same Check Point Cryptographic Library,

Check Point's SmartConsole software (installed on a Windows 10 workstation), while not part of the TOE (i.e., not a TOE component), allows the administrator to remotely manage a deployment. Like a browser, SmartConsole does not enforce any security functions, but instead interacts with the TOE to facilitate remote administration. The administrator can also locally administer each TOE component through access to a Command Line Interface (CLI) over a console connection. The TOE component, Management server and HUB Gateway all offer a CLI.

Check Point R81.00 software is installed on a hardware platform in accordance with TOE guidance, in a FIPS 140-2 mode. The R81.00 software provides the TOE with storage for audit trail, an IP stack for in-TOE routing, NIC drivers and an execution environment for daemons and security servers.

Check Point Security Gateway and Maestro Hyperscale Appliances mediate information flows between clients and servers located on internal and external networks governed by the firewall. User authentication may be achieved by a remote access client authenticating using IKE, against either a pre-shared key or certificate. Administrators also need to authenticate to the TOE before they can use the Management GUIs to access Security Management.

Check Point's virtual machine engine supports the definition of separate execution domains for Virtual Systems. Incoming IP packets bind to an appropriate VS corresponding to the logical interface (i.e. physical or virtual LAN interface) on which they are received, and the VS that is defined to receive the packet from that interface. The packets are labeled with the VSID, and are handled in the context of that VS's execution domain, until they are dropped, forwarded out of the gateway, or handed to another VS according to administrator-defined rules.

The product additionally imposes traffic-filtering controls on mediated information flows between clients and servers according to the site's security policy rules. By default, these security policy rules deny all inbound and outbound information flows through the TOE. Only an authorized administrator has the authority to change the security policy rules.

#### 1.4.1.1 Physical Boundaries

The TOE is a distributed TOE consistent with Use Case 3 as defined in the NDcPP22e. There are Check Point Security Gateway and Maestro Hyperscale Appliances as well as Security Management Appliances. All platforms use the same image. The difference is mainly in hardware makeup and physical ports. All platforms are x86 based hardware. See Section 7 for a detailed hardware listing.

The SmartConsole Management GUI software is installed on a Windows workstation (Windows 10 Enterprise). Authorized administrators use the GUI software or CLI to remotely manage the TOE. The TOE may be configured to interact with an external syslog server.

#### 1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by R81.00.

- Security audit
- Communication
- Cryptographic support
- User Data Protection
- Stateful Traffic Filtering Firewall
- Identification and authentication
- Security management
- Packet Filtering
- Protection of the TSF
- TOE access
- Trusted path/channels

#### 1.4.1.2.1 Security audit

The TOE generates audit logs and has the capability to store them internally or to send them to an external audit server. The connection between the TOE and the remote audit server is protected with IPsec. The TOE has a disk cleanup procedure where it removes old audit logs to allow space for new ones. When disk space falls below a predefined threshold (the cleanup procedure cannot keep up with the audit collection), the TOE stops collecting audit records.

#### 1.4.1.2.2 Communication

The TOE is a distributed solution consisting of Security Gateway and Maestro Hyperscale Appliances as well as a Security Management Server. The Security Management Server can manage one or more Security Gateway and Maestro Hyperscale Appliances.

#### 1.4.1.2.3 Cryptographic support

The TOE uses the Check Point Cryptographic Library version 1.1 that has received Cryptographic Algorithm Validation Program (CAVP) certificates for all cryptographic functions claimed in this ST. Cryptographic services include key management, random bit generation, encryption/decryption, digital signature and secure hashing.

#### 1.4.1.2.4 User Data Protection

The TOE ensures that residual information is protected from potential reuse in accessible objects such as network packets.

#### 1.4.1.2.5 Stateful Traffic Filtering Firewall

The TOE supports many protocols for packet filtering including icmpv4, icmpv6, ipv4, ipv6, tcp and udp. The firewall rules implement the SPD rules (permit, deny, bypass). Each rule can be configured to log status of packets pertaining to the rule. All codes under each protocol are implemented. The TOE supports FTP for stateful filtering.

Routed packets are forwarded to a TOE interface with the interface's MAC address as the layer-2 destination address. The TOE routes the packets using the presumed destination address in the IP header, in accordance with route tables maintained by the TOE.

IP packets are processed by the Check Point R81.00 software, which associates them with application-level connections, using the IP packet header fields: source and destination IP address and port, as well as IP protocol. Fragmented packets are reassembled before they are processed.

The TOE mediates the information flows according to an administrator-defined policy. Some of the traffic may be either silently dropped or rejected (with notification to the presumed source).

The TOE's firewall and VPN capabilities are controlled by defining an ordered set of rules in the Security Rule Base. The Rule Base specifies what communication will be allowed to pass and what will be blocked. It specifies the source and destination of the communication, what services can be used, at what times, whether to log the connection and the logging level.

#### 1.4.1.2.6 Identification and authentication

The TOE implements a password-based authentication mechanism for authenticating users and requires identification and authentication before allowing access. Only the banner may be presented before authentication is complete. The TOE supports passwords of varying length and allows an administrator to specify a minimum password length between 8 and 100 characters long. The password composition can contain all special characters as required by FIA_PMG_EXT.1.1.

Internally, the TOE keeps track of failed login attempts and if the configured number of attempts is met, the administrator is either locked out for a period of time or until the primary administrator unlocks the account. The local CLI remains available when the remote account is locked out.

The TOE's IPsec implementation supports Pre-Shared Keys (PSKs) and X.509 certificates (both RSA and ECDSA) for IKE authentication.

#### 1.4.1.2.7 Security management

The TOE allows both local and remote administration for management of the TOE's security functions. The TOE creates and maintains roles for configured administrators. An administrator can log in locally to the TOE using a serial connection. The local login operates in a Command Line Interface (CLI). There is one remote administration interface that can be used once the TOE is in its evaluated configuration. The remote administration interface is executed through a Graphical User Interface program named SmartConsole using a connection protected by IPsec.

#### 1.4.1.2.8 Packet Filtering

Please see Section 1.4.1.2.5 Stateful Traffic Filtering Firewall for a description of the TOE's packet filtering mechanism.

#### 1.4.1.2.9 Protection of the TSF

The TOE includes capabilities to protect itself from unwanted modification as well as protecting its persistent data.

The TOE does not store passwords in plaintext; they are obfuscated. The TOE does not support any command line capability to view any cryptographic keys generated or used by the TOE.

The TOE only allows updates after their signature is successfully verified. The TOE update mechanism uses ECDSA with SHA-512 and P-521 to verify the signature of the update package.

The TOE's FIPS executables are signed using ECDSA with SHA-512 and P-521. For all other executables a hash is computed during system installation and configuration and during updates.

During power-up the integrity of all executables is verified. If an integrity test fails in the cryptographic module, the system will enter a kernel panic and will fail to boot up. If an integrity test fails due to a non-matching hash, a log is written. Also during power-up, algorithms are tested in the kernel and user-space. If any of these test fail, the TOE is not operational for users.

The TOE protects all communications among its distributed components with IPsec.

The TOE provides a timestamp for use with audit records, timing elements of cryptographic functions, and inactivity timeouts.

#### 1.4.1.2.10 TOE access

The TOE is able to terminate interactive sessions if the session is inactive for an administrator configured period of time. The TOE also allows a session to be disconnected via a logout command. An administrator can configure a login banner to be displayed before authentication is completed.

#### 1.4.1.2.11 Trusted path/channels

The TOE protects all communications with outside entities using IPsec communications only. The TOE employs IPsec when it sends audit data to an audit server, and when allowing remote administration connections. Any protocol that is part of the distributed TOE must be protected in an IPsec connection.

### 1.4.2 TOE Documentation

- Check Point Software Technologies LTD.Security Gateway Appliances R81.00 Common Criteria Supplement, Version 1.0, March 16, 2022 (**CC Guide**)
- Check Point Software Technologies LTD. R81.00 NIAP Installation Guide, Version 1.0, March 16, 2022 (**Install Guide**)

# 2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
    - Part 3 Conformant
- Package Claims:
    - PP-Configuration for Network Devices, Stateful Traffic Filter Firewalls, and Virtual Private Network (VPN) Gateways (CFG_NDcPP-FW-VPNGW_V1.1), Version 1.1, 2020-07-01.
        - Base-PP: collaborative Protection Profile for Network Devices, Version 2.2e, 23 March 2020 (cPP_ND_v2.2e) herein referenced as NDcPP22e
        - PP-Module: PP-Module for Virtual Private Network (VPN) Gateways, 1.1, 18 June 2020 (MOD_VPNGW_v1.1) herein referenced as VPNGW11
        - PP-Module: PP-Module for Stateful Traffic Filter Firewalls, Version 1.4e, 25 June 2020 (MOD_cPP_FW_v1.4e) herein referenced as STFFW14e
- NIAP Technical Decisions

| TD No. | Applied? | Rationale |
|---|---|---|
| TD0597 – VPNGW11 | Yes | FPF_RUL_EXT.1.6 present |
| TD0592 – NDcPP22e | Yes | NIT Technical Decision for Local Storage of Audit Records |
| TD0591 – NDcPP22e | Yes | NIT Technical Decision for Virtual TOEs and hypervisors |
| TD0590 – VPNGW11 | Yes | Maps PP-Module objectives to Assumptions |
| TD0581 – NDcPP22e | Yes | FCS_CKM.2 SFR present |
| TD0580 – NDcPP22e | Yes | FCS_CKM.2 SFR present |
| TD0572 – NDcPP22e | Yes | FTP_ITC.1 SFR mandatory |
| TD0571 – NDcPP22e | Yes | FIA_AFL.1 SFR mandatory |
| TD0570 – NDcPP22e | Yes | FIA_AFL.1 SFR mandatory |
| TD0569 – NDcPP22e | No | FCS_DTLSS_EXT.1.7 & FCS_TLSS_EXT.1.4 not claimed |
| TD0564 – NDcPP22e | Yes | Public-Vulnerability-Based searching mandatory |
| TD0563 – NDcPP22e | Yes | FAU_GEN.1.2 mandatory |
| TD0556 – NDcPP22e | No | FCS_TLSS_EXT.1.4 not claimed |
| TD0555 – NDcPP22e | No | FCS_TLSS_EXT.1.4 not claimed |
| TD0551 – STFFW14e | Yes | Completes mappings of OEs in sections 5.3.2 and 5.3.4 |
| TD0549 – VPNGW11 | Yes | Adds Assumptions and OSP to section 6.1.2 |
| TD0547 – NDcPP22e | Yes | AVA_VAN mandatory |
| TD0546 – NDcPP22e | No | FCS_DTLSC_EXT.1.1 not claimed |
| TD0545 – STFFW14e | Yes | Applies (and allows TOE design to prevent conflicting rules). |
| TD0538 – NDcPP22e | Yes | Allows PP-Modules |
| TD0537 – NDcPP22e | No | FCS_TLSC_EXT.2.3 not claimed |
| TD0536 – NDcPP22e | Yes | Update Verification Inconsistency |
| TD0528 – NDcPP22e | No | FCS_NTP_EXT.1 not claimed |
| TD0527 – NDcPP22e | Yes | FIA_X509_EXT.1 claimed |

## 2.1 Conformance Rationale

The ST conforms to the NDcPP22e/VPNGW11/STFFW14e. As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the PP.

# 3. Security Objectives

The Security Problem Definition may be found in the NDcPP22e/VPNGW11/STFFW14e and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The NDcPP22e/VPNGW11/STFFW14e offers additional information about the identified security objectives, but that has not been reproduced here and the NDcPP22e/VPNGW11/STFFW14e should be consulted if there is interest in that material.

In general, the NDcPP22e/VPNGW11/STFFW14e has defined Security Objectives appropriate for network device (and VPN gateway and firewall) and as such are applicable to the Gateway Appliances R81.00 TOE.

## 3.1 Security Objectives for the Operational Environment

**OE.ADMIN_CREDENTIALS_SECURE** The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

**OE.COMPONENTS_RUNNING** (applies to distributed TOEs only)

For distributed TOEs, the Security Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The Security Administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly.

**OE.CONNECTIONS** The TOE is connected to distinct networks in a manner that ensures that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks.

**OE.NO_GENERAL_PURPOSE** There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. Note: For vNDs the TOE includes only the contents of its own VM, and does not include other VMs or the VS.

**OE.NO_THRU_TRAFFIC_PROTECTION** The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.

**OE.PHYSICAL** Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

**OE.RESIDUAL_INFORMATION** The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.

**OE.TRUSTED_ADMIN** TOE Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality.

For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.

**OE.UPDATES** The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

**OE.VM_CONFIGURATION** (applies to vNDs only)

For vNDs, the Security Administrator ensures that the VS and VMs are configured to

- reduce the attack surface of VMs as much as possible while supporting ND functionality (e.g., remove unnecessary virtual hardware, turn off unused inter-VM communications mechanisms), and
- correctly implement ND functionality (e.g., ensure virtual networking is properly configured to support network traffic, management channels, and audit reporting).

The VS should be operated in a manner that reduces the likelihood that vND operations are adversely affected by virtualization features such as cloning, save/restore, suspend/resume, and live migration.

If possible, the VS should be configured to make use of features that leverage the VS's privileged position to provide additional security functionality. Such features could include malware detection through VM introspection, measured VM boot, or VM snapshot for forensic analysis.

# 4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the NDcPP22e/STFFW14e/VPNGW11. The NDcPP22e/STFFW14e/VPNGW11 defines the following extended requirements and since they are not redefined in this ST the NDcPP22e/STFFW14e/VPNGW11 should be consulted for more information in regard to those CC extensions.

**Extended SFRs:**

NDcPP22e:FAU_GEN_EXT.1: Security Audit Generation

NDcPP22e:FAU_STG_EXT.1: Protected Audit Event Storage

NDcPP22e:FAU_STG_EXT.4: Protected Local Audit Event Storage for Distributed TOEs

NDcPP22e:FAU_STG_EXT.5: Protected Remote Audit Event Storage for Distributed TOEs

NDcPP22e:FCO_CPC_EXT.1: Component Registration Channel Definition

NDcPP22e:FCS_IPSEC_EXT.1: IPsec Protocol

VPNGW11:FCS_IPSEC_EXT.1: IPsec Protocol

NDcPP22e:FCS_RBG_EXT.1: Random Bit Generation

STFFW14e:FFW_RUL_EXT.1: Stateful Traffic Filtering

STFFW14e:FFW_RUL_EXT.2: Stateful Filtering of Dynamic Protocols

NDcPP22e:FIA_PMG_EXT.1: Password Management

VPNGW11:FIA_PSK_EXT.1: Pre-Shared Key Composition

NDcPP22e:FIA_UAU_EXT.2: Password-based Authentication Mechanism

NDcPP22e:FIA_UIA_EXT.1: User Identification and Authentication

NDcPP22e:FIA_X509_EXT.1/ITT: X.509 Certificate Validation

NDcPP22e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation

VPNGW11:FIA_X509_EXT.1/Rev: X.509 Certificate Validation

NDcPP22e:FIA_X509_EXT.2: X.509 Certificate Authentication

VPNGW11:FIA_X509_EXT.2: X.509 Certificate Authentication

NDcPP22e:FIA_X509_EXT.3: X.509 Certificate Requests

VPNGW11:FIA_X509_EXT.3: X.509 Certificate Requests

VPNGW11:FPF_RUL_EXT.1: Rules for Packet Filtering

NDcPP22e:FPT_APW_EXT.1: Protection of Administrator Passwords

NDcPP22e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

NDcPP22e:FPT_STM_EXT.1: Reliable Time Stamps

NDcPP22e:FPT_TST_EXT.1: TSF testing

VPNGW11:FPT_TST_EXT.1: TSF Testing

VPNGW11:FPT_TST_EXT.3: TSF Self-Test with Defined Methods

NDcPP22e:FPT_TUD_EXT.1: Trusted update

VPNGW11:FPT_TUD_EXT.1: Trusted Update

NDcPP22e:FTA_SSL_EXT.1: TSF-initiated Session Locking

# 5. Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the NDcPP22e/STFFW14e/VPNGW11. The refinements and operations already performed in the NDcPP22e/STFFW14e/VPNGW11 are not identified (e.g., highlighted) here, rather the requirements have been copied from the NDcPP22e/STFFW14e/VPNGW11 and any residual operations have been completed herein. Of particular note, the NDcPP22e/STFFW14e/VPNGW11 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary. Also, where a requirement is duplicated across PP and modules all versions are included in the ST to ensure that all aspects of the requirement are captured.

The SARs are also drawn from the NDcPP22e/STFFW14e/VPNGW11 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the NDcPP22e/STFFW14e/VPNGW11 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The NDcPP22e/STFFW14e/VPNGW11 should be consulted for the assurance activity definitions.

## 5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by Check Point Software Security Gateway and Maestro Hyperscale Appliances TOE.

| Requirement Class | Requirement Component |
|---|---|
| **FAU: Security audit** | NDcPP22e:FAU_GEN.1: Audit Data Generation |
| | STFFW14e:FAU_GEN.1: Security Audit Data Generation |
| | VPNGW11:FAU_GEN.1: Audit Data Generation |
| | NDcPP22e:FAU_GEN.2: User identity association |
| | NDcPP22e:FAU_GEN_EXT.1: Security Audit Generation |
| | NDcPP22e:FAU_STG_EXT.1: Protected Audit Event Storage |
| | NDcPP22e:FAU_STG_EXT.4: Protected Local Audit Event Storage for Distributed TOEs |
| | NDcPP22e:FAU_STG_EXT.5: Protected Remote Audit Event Storage for Distributed TOEs |
| **FCO: Communication** | NDcPP22e:FCO_CPC_EXT.1: Component Registration Channel Definition |
| **FCS: Cryptographic support** | NDcPP22e:FCS_CKM.1: Cryptographic Key Generation |
| | VPNGW11:FCS_CKM.1/IKE: Cryptographic Key Generation (for IKE Peer Authentication) |
| | NDcPP22e:FCS_CKM.2: Cryptographic Key Establishment |
| | NDcPP22e:FCS_CKM.4: Cryptographic Key Destruction |
| | NDcPP22e:FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/Decryption) |
| | VPNGW11:FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/Decryption) |
| | NDcPP22e:FCS_COP.1/Hash: Cryptographic Operation (Hash Algorithm) |
| | NDcPP22e:FCS_COP.1/KeyedHash: Cryptographic Operation (Keyed Hash Algorithm) |
| | NDcPP22e:FCS_COP.1/SigGen: Cryptographic Operation (Signature Generation and Verification) |
| | NDcPP22e:FCS_IPSEC_EXT.1: IPsec Protocol |
| | VPNGW11:FCS_IPSEC_EXT.1: IPsec Protocol |
| | NDcPP22e:FCS_RBG_EXT.1: Random Bit Generation |

| | |
|---|---|
| **FDP: User data protection** | STFFW14e:FDP_RIP.2: Full Residual Information Protection |
| **FFW: Firewall** | STFFW14e:FFW_RUL_EXT.1: Stateful Traffic Filtering |
| | STFFW14e:FFW_RUL_EXT.2: Stateful Filtering of Dynamic Protocols |
| **FIA: Identification and authentication** | NDcPP22e:FIA_AFL.1: Authentication Failure Management |
| | NDcPP22e:FIA_PMG_EXT.1: Password Management |
| | VPNGW11:FIA_PSK_EXT.1: Pre-Shared Key Composition |
| | NDcPP22e:FIA_UAU.7: Protected Authentication Feedback |
| | NDcPP22e:FIA_UAU_EXT.2: Password-based Authentication Mechanism |
| | NDcPP22e:FIA_UIA_EXT.1: User Identification and Authentication |
| | NDcPP22e:FIA_X509_EXT.1/ITT: X.509 Certificate Validation |
| | NDcPP22e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation |
| | VPNGW11:FIA_X509_EXT.1/Rev: X.509 Certificate Validation |
| | NDcPP22e:FIA_X509_EXT.2: X.509 Certificate Authentication |
| | VPNGW11:FIA_X509_EXT.2: X.509 Certificate Authentication |
| | NDcPP22e:FIA_X509_EXT.3: X.509 Certificate Requests |
| | VPNGW11:FIA_X509_EXT.3: X.509 Certificate Requests |
| **FMT: Security management** | NDcPP22e:FMT_MOF.1/ManualUpdate: Management of security functions behaviour |
| | NDcPP22e:FMT_MOF.1/Services: Management of Security Functions Behaviour |
| | NDcPP22e:FMT_MTD.1/CoreData: Management of TSF Data |
| | NDcPP22e:FMT_MTD.1/CryptoKeys: Management of TSF Data |
| | VPNGW11:FMT_MTD.1/CryptoKeys: Management of TSF Data |
| | NDcPP22e:FMT_SMF.1: Specification of Management Functions |
| | VPNGW11:FMT_SMF.1: Specification of Management Functions |
| | STFFW14e:FMT_SMF.1/FFW: Specification of Management Functions |
| | VPNGW11:FMT_SMF.1/VPN: /VPN Specification of Management Functions (VPN Gateway) |
| | NDcPP22e:FMT_SMR.2: Restrictions on Security Roles |
| **FPF: Packet Filtering** | VPNGW11:FPF_RUL_EXT.1: Rules for Packet Filtering |
| **FPT: Protection of the TSF** | NDcPP22e:FPT_APW_EXT.1: Protection of Administrator Passwords |
| | VPNGW11:FPT_FLS.1/SelfTest: Fail Secure (Self-Test Failures) |
| | NDcPP22e:FPT_ITT.1: Basic internal TSF data transfer protection |
| | NDcPP22e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) |
| | NDcPP22e:FPT_STM_EXT.1: Reliable Time Stamps |
| | NDcPP22e:FPT_TST_EXT.1: TSF testing |
| | VPNGW11:FPT_TST_EXT.1: TSF Testing |
| | VPNGW11:FPT_TST_EXT.3: TSF Self-Test with Defined Methods |
| | NDcPP22e:FPT_TUD_EXT.1: Trusted update |
| | VPNGW11:FPT_TUD_EXT.1: Trusted Update |
| **FTA: TOE access** | NDcPP22e:FTA_SSL.3: TSF-initiated Termination |
| | NDcPP22e:FTA_SSL.4: User-initiated Termination |
| | NDcPP22e:FTA_SSL_EXT.1: TSF-initiated Session Locking |
| | NDcPP22e:FTA_TAB.1: Default TOE Access Banners |
| **FTP: Trusted path/channels** | NDcPP22e:FTP_ITC.1: Inter-TSF trusted channel |
| | VPNGW11:FTP_ITC.1/VPN: Inter-TSF Trusted Channel (VPN Communications) |
| | NDcPP22e:FTP_TRP.1/Admin: Trusted Path |
| | NDcPP22e:FTP_TRP.1/Join: Trusted Path |

**Table 5-1 TOE Security Functional Components**

## 5.1.1 Security audit (FAU)

### 5.1.1.1 Audit Data Generation (NDcPP22e:FAU_GEN.1)

**NDcPP22e:FAU_GEN.1.1**

The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions comprising:
- Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).
- Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
- Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
- Resetting passwords (name of related user account shall be logged).
- [***no other actions***];

d) Specifically defined auditable events listed in Table 5-2.

**Table 5-2 Audit Events**

| Requirement | Auditable Events | Additional Content |
|---|---|---|
| **NDcPP22e:FAU_GEN.1** | None | None |
| **STFFW14e:FAU_GEN.1** | None | None |
| **VPNGW11:FAU_GEN.1** | None | None |
| **NDcPP22e:FAU_GEN.2** | None | None |
| **NDcPP22e:FAU_GEN_EXT.1** | None | None |
| **NDcPP22e:FAU_STG_EXT.1** | None | None |
| **NDcPP22e:FAU_STG_EXT.4** | None | None |
| **NDcPP22e:FAU_STG_EXT.5** | None | None |
| **NDcPP22e:FCO_CPC_EXT.1** | Enabling communications between a pair of components. Disabling communications between a pair of components. | Identities of the endpoints pairs enabled or disabled. |
| **NDcPP22e:FCS_CKM.1** | None | None |
| **VPNGW11:FCS_CKM.1/IKE** | None | None |
| **NDcPP22e:FCS_CKM.2** | None | None |
| **NDcPP22e:FCS_CKM.4** | None | None |
| **NDcPP22e:FCS_COP.1/DataEncryption** | None | None |
| **VPNGW11:FCS_COP.1/DataEncryption** | None | None |
| **NDcPP22e:FCS_COP.1/Hash** | None | None |
| **NDcPP22e:FCS_COP.1/KeyedHash** | None | None |
| **NDcPP22e:FCS_COP.1/SigGen** | None | None |
| **NDcPP22e:FCS_IPSEC_EXT.1** | Failure to establish an IPsec SA. | Reason for failure. |
| **VPNGW11:FCS_IPSEC_EXT.1** | Session Establishment with peer | Entire packet contents of packets transmitted/received during session establishment |
| **NDcPP22e:FCS_RBG_EXT.1** | None | None |
| **STFFW14e:FDP_RIP.2** | None | None |
| **STFFW14e:FFW_RUL_EXT.1** | Application of rules configured with the 'log' operation | Source and destination addresses<br>Source and destination ports<br>Transport Layer Protocol<br>TOE Interface |
| **STFFW14e:FFW_RUL_EXT.2** | Dynamical definition of rule and | None |

| | | |
|---|---|---|
| | Establishment of a session | |
| **NDcPP22e:FIA_AFL.1** | Unsuccessful login attempt limit is met or exceeded. | Origin of the attempt (e.g., IP address). |
| **NDcPP22e:FIA_PMG_EXT.1** | None | None |
| **VPNGW11:FIA_PSK_EXT.1** | None | None |
| **NDcPP22e:FIA_UAU.7** | None | None |
| **NDcPP22e:FIA_UAU_EXT.2** | All use of identification and authentication mechanism. | Origin of the attempt (e.g., IP address). |
| **NDcPP22e:FIA_UIA_EXT.1** | All use of identification and authentication mechanism. | Origin of the attempt (e.g., IP address). |
| **NDcPP22e:FIA_X509_EXT.1/ITT** | Unsuccessful attempt to validate a certificate. Any addition, replacement or removal of trust anchors in the TOE's trust store | Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store |
| **NDcPP22e:FIA_X509_EXT.1/Rev** | Unsuccessful attempt to validate a certificate. Any addition, replacement or removal of trust anchors in the TOE's trust store | Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store |
| **VPNGW11:FIA_X509_EXT.1/Rev** | None | None |
| **NDcPP22e:FIA_X509_EXT.2** | None | None |
| **VPNGW11:FIA_X509_EXT.2** | None | None |
| **NDcPP22e:FIA_X509_EXT.3** | None | None |
| **VPNGW11:FIA_X509_EXT.3** | None | None |
| **NDcPP22e:FMT_MOF.1/ManualUpdate** | Any attempt to initiate a manual update. | None |
| **NDcPP22e:FMT_MOF.1/Services** | None | None |
| **NDcPP22e:FMT_MTD.1/CoreData** | None | None |
| **NDcPP22e:FMT_MTD.1/CryptoKeys** | Management of cryptographic keys. | None |
| **VPNGW11:FMT_MTD.1/CryptoKeys** | None | None |
| **NDcPP22e:FMT_SMF.1** | All management activities of TSF data. | None |
| **VPNGW11:FMT_SMF.1** | None | None |
| **STFFW14e:FMT_SMF.1/FFW** | All management activities of TSF data (including creation, modification and deletion of firewall rules). | None |
| **VPNGW11:FMT_SMF.1/VPN** | None | None |
| **NDcPP22e:FMT_SMR.2** | None | None |
| **VPNGW11:FPF_RUL_EXT.1** | Application of rules configured with the 'log' operation | Source and destination addresses Source and destination ports Transport Layer Protocol |
| **NDcPP22e:FPT_APW_EXT.1** | None | None |
| **VPNGW11:FPT_FLS.1/SelfTest** | None | None |
| **NDcPP22e:FPT_ITT.1** | Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions. | Identification of the initiator and target of failed trusted channels establishment attempt. |
| **NDcPP22e:FPT_SKP_EXT.1** | None | None |
| **NDcPP22e:FPT_STM_EXT.1** | Discontinuous changes to time - either Administrator actuated or | For discontinuous changes to time: The old and new values for |

| | | |
|---|---|---|
| | changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1) | the time. Origin of the attempt to change time for success and failure (e.g., IP address). |
| **NDcPP22e:FPT_TST_EXT.1** | None | None |
| **VPNGW11:FPT_TST_EXT.1** | None | None |
| **VPNGW11:FPT_TST_EXT.3** | None | None |
| **NDcPP22e:FPT_TUD_EXT.1** | Initiation of update; result of the update attempt (success or failure). | None |
| **VPNGW11:FPT_TUD_EXT.1** | None | None |
| **NDcPP22e:FTA_SSL.3** | The termination of a remote session by the session locking mechanism. | None |
| **NDcPP22e:FTA_SSL.4** | The termination of an interactive session. | None |
| **NDcPP22e:FTA_SSL_EXT.1** | (if 'lock the session' is selected) Any attempts at unlocking of an interactive session. (if 'terminate the session' is selected) The termination of a local session by the session locking mechanism. | None |
| **NDcPP22e:FTA_TAB.1** | None | None |
| **NDcPP22e:FTP_ITC.1** | Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions. | Identification of the initiator and target of failed trusted channels establishment attempt. |
| **VPNGW11:FTP_ITC.1/VPN** | None | None |
| **NDcPP22e:FTP_TRP.1/Admin** | Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions. | None |
| **NDcPP22e:FTP_TRP.1/Join** | Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions. | None |

**NDcPP22e:FAU_GEN.1.2**

The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, information specified in column three of Table 5-2.

#### 5.1.1.2 Security Audit Data Generation (STFFW14e:FAU_GEN.1)

**STFFW14e:FAU_GEN.1.1**

Unmodified from Base PP.

Since this SFR is simply attempting to add audit events to the Base-PP audit table, the audit events from STFFW14e PP-Module are included in Table 5-2 Audit Events.

#### 5.1.1.3 Audit Data Generation (VPNGW11:FAU_GEN.1)

**VPNGW11:FAU_GEN.1.1**
is refined to include the following auditable events in addition to what is defined in the Base-PP.

Since this SFR is simply attempting to add audit events to the Base-PP audit table, the audit events from VPNGW11:FAU_GEN.1.1 are included in Table 5-2 Audit Events.

#### 5.1.1.4 User identity association (NDcPP22e:FAU_GEN.2)

**NDcPP22e:FAU_GEN.2.1**
For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

#### 5.1.1.5 Security Audit Generation (NDcPP22e:FAU_GEN_EXT.1)

**NDcPP22e:FAU_GEN_EXT.1.1**
The TSF shall be able to generate audit records for each TOE component. The audit records generated by the TSF of each TOE component shall include the subset of security relevant audit events which can occur on the TOE component.

#### 5.1.1.6 Protected Audit Event Storage (NDcPP22e:FAU_STG_EXT.1)

**NDcPP22e:FAU_STG_EXT.1.1**
The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

**NDcPP22e:FAU_STG_EXT.1.2**
The TSF shall be able to store generated audit data on the TOE itself. In addition [

- ***The TOE shall be a distributed TOE that stores audit data on the following TOE components: [Security Gateways and Security Management Servers],***
- ***The TOE shall be a distributed TOE with storage of audit data provided externally for the following TOE components: [Security Gateways transmits its audit data to Security Management Servers or to an external syslog server.].***

]

**NDcPP22e:FAU_STG_EXT.1.3**
The TSF shall [***drop new audit data***] when the local storage space for audit data is full.

#### 5.1.1.7 Protected Local Audit Event Storage for Distributed TOEs (NDcPP22e:FAU_STG_EXT.4)

**NDcPP22e:FAU_STG_EXT.4.1**
The TSF of each TOE component which stores security audit data locally shall perform the following actions when the local storage space for audit data is full: [**drop new audit data**].

#### 5.1.1.8 Protected Remote Audit Event Storage for Distributed TOEs (NDcPP22e:FAU_STG_EXT.5)

**NDcPP22e:FAU_STG_EXT.5.1**
Each TOE component which does not store security audit data locally shall be able to buffer security audit data locally until it has been transferred to another TOE component that stores or forwards it. All transfer of audit records between TOE components shall use a protected channel according to [***FPT_ITT.1, FTP_ITC.1***].

## 5.1.2 Communication (FCO)

### 5.1.2.1 Component Registration Channel Definition (NDcPP22e:FCO_CPC_EXT.1)

**NDcPP22e:FCO_CPC_EXT.1.1**
The TSF shall require a Security Administrator to enable communications between any pair of TOE components before such communication can take place.

**NDcPP22e:FCO_CPC_EXT.1.2**
The TSF shall implement a registration process in which components establish and use a communications channel that uses [***A channel that meets the secure registration channel requirements in FTP_TRP.1/Join***] for at least TSF data.

**NDcPP22e:FCO_CPC_EXT.1.3**
The TSF shall enable a Security Administrator to disable communications between any pair of TOE components.

## 5.1.3 Cryptographic support (FCS)

### 5.1.3.1 Cryptographic Key Generation (NDcPP22e:FCS_CKM.1)

**NDcPP22e:FCS_CKM.1.1**
The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

- ***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,***
- ***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***
- ***FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526]***].

### 5.1.3.2 Cryptographic Key Generation (for IKE Peer Authentication) (VPNGW11:FCS_CKM.1/IKE)

**VPNGW11:FCS_CKM.1.1/IKE**
The TSF shall generate asymmetric cryptographic keys used for IKE peer authentication in accordance with a specified cryptographic key generation algorithm: [

- ***FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3 for RSA schemes,***
- ***FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4 for ECDSA schemes and implementing 'NIST curves' P-256, P-384 and [P-521]***]

and

[- ***FFC Schemes using 'safe-prime' groups that meet the following: 'NIST Special Publication 800-56A Revision 3, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' and [RFC 3526],***]

and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

### 5.1.3.3 Cryptographic Key Establishment (NDcPP22e:FCS_CKM.2)

**NDcPP22e:FCS_CKM.2.1**
The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- ***RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications" Version 2.1,***

- ***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' (TD0581 applied)***
- ***FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [selection: groups listed in RFC 3526, groups listed in RFC 7919] (TD0580 applied)***].

#### 5.1.3.4 Cryptographic Key Destruction (NDcPP22e:FCS_CKM.4)

**NDcPP22e:FCS_CKM.4.1**

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- For plaintext keys in volatile storage, the destruction shall be executed by a [***single overwrite consisting of [zeroes]***];
- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [***logically addresses the storage location of the key and performs a [single] overwrite consisting of [zeroes]***]

that meets the following: No Standard.

#### 5.1.3.5 Cryptographic Operation (AES Data Encryption/Decryption) (NDcPP22e:FCS_COP.1/DataEncryption)

**NDcPP22e:FCS_COP.1.1/DataEncryption**

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [***CBC, GCM***] mode and cryptographic key sizes [***128 bits, 256 bits***] that meet the following: AES as specified in ISO 18033-3, [***CBC as specified in ISO 10116, GCM as specified in ISO 19772***].

#### 5.1.3.6 Cryptographic Operation (AES Data Encryption/Decryption) (VPNGW11:FCS_COP.1/DataEncryption)

**VPNGW11:FCS_COP.1.1/DataEncryption**

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [***CBC, GCM***] and [***no other***] mode and cryptographic key sizes [***128 bits, 256 bits***], and [***no other cryptographic key sizes***] that meet the following: AES as specified in ISO 18033-3, [***CBC as specified in ISO 10116, GCM as specified in ISO 19772***] and [***no other standards***].

#### 5.1.3.7 Cryptographic Operation (Hash Algorithm) (NDcPP22e:FCS_COP.1/Hash)

**NDcPP22e:FCS_COP.1.1/Hash**

The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384, SHA-512***] and message digest sizes [***160, 256, 384, 512***] bits that meet the following: ISO/IEC 10118-3:2004.

#### 5.1.3.8 Cryptographic Operation (Keyed Hash Algorithm) (NDcPP22e:FCS_COP.1/KeyedHash)

**NDcPP22e:FCS_COP.1.1/KeyedHash**

The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm [***HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384. HMAC-SHA-512***] and cryptographic key sizes [**160, 256, 384, 512**] and message digest sizes [***160, 256, 384, 512***] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 'MAC Algorithm 2'.

#### 5.1.3.9 Cryptographic Operation (Signature Generation and Verification) (NDcPP22e:FCS_COP.1/SigGen)

**NDcPP22e:FCS_COP.1.1/SigGen**

The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [

- ***RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048, 4096 bits],***
- ***- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256, 384 or 521 bits]***]

that meet the following: [

- ***For RSA schemes: 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,***
- ***For ECDSA schemes: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 6 and Appendix D, Implementing 'NIST curves' [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4***].

#### 5.1.3.10 IPsec Protocol (NDcPP22e:FCS_IPSEC_EXT.1 and VPNGW11:FCS_IPSEC_EXT.1)

**NDcPP22e:FCS_IPSEC_EXT.1.1**

The TSF shall implement the IPsec architecture as specified in RFC 4301.

**NDcPP22e:FCS_IPSEC_EXT.1.2**

The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

**NDcPP22e:FCS_IPSEC_EXT.1.3**

The TSF shall implement [***tunnel mode***].

**VPNGW11:FCS_IPSEC_EXT.1.3**

The TSF shall implement [***tunnel mode***].

**NDcPP22e:FCS_IPSEC_EXT.1.4**

The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [***AES-CBC-128 (RFC 3602), AES-CBC-256 (RFC 3602), AES-GCM-128 (RFC 4106), AES-GCM-256 (RFC 4106)***] together with a Secure Hash Algorithm (SHA)-based HMAC [***HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512***].

**VPNGW11:FCS_IPSEC_EXT.1.4**

The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [***AES-CBC-128 (RFC 3602), AES-CBC-256 (RFC 3602), AES-GCM-128 (RFC 4106), AES-GCM-256 (RFC 4106)***] and [***no other algorithm***] together with a Secure Hash Algorithm (SHA)-based HMAC [***HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512***].

**NDcPP22e:FCS_IPSEC_EXT.1.5**

The TSF shall implement the protocol: [

- ***IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109, [RFC 4304 for extended sequence numbers], and [no other RFCs for hash functions],***
- ***IKEv2 as defined in RFC 5996 and [with no support for NAT traversal], and [no other RFCs for hash functions]***].

**NDcPP22e:FCS_IPSEC_EXT.1.6**

The TSF shall ensure the encrypted payload in the [***IKEv1, IKEv2***] protocol uses the cryptographic algorithms [***AES-CBC-128, AES-CBC-256 (specified in RFC 3602)***].

**NDcPP22e:FCS_IPSEC_EXT.1.7**

The TSF shall ensure that [

- ***IKEv1 Phase 1 SA lifetimes can be configured by a Security Administrator based on [***
  - ***o length of time, where the time values can be configured within [1 - 24] hours],***
- ***IKEv2 SA lifetimes can be configured by a Security Administrator based on [***
  - ***o length of time, where the time values can be configured within [1 - 24] hours]***].

**NDcPP22e:FCS_IPSEC_EXT.1.8**

The TSF shall ensure that [

- ***IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on [***
  - ***length of time, where the time values can be configured within [1 - 8] hours],***
- ***IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [***
  - ***length of time, where the time values can be configured within [1 - 8] hours]***].

**NDcPP22e:FCS_IPSEC_EXT.1.9**

The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ('x' in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [**224, 256 and 384**] bits.

**NDcPP22e:FCS_IPSEC_EXT.1.10**

The TSF shall generate nonces used in [***IKEv1, IKEv2***] exchanges of length [***- at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash***].

**NDcPP22e:FCS_IPSEC_EXT.1.11**

The TSF shall ensure that IKE protocols implement DH Group(s) [***[14 (2048-bit MODP)], [19 (256-bit Random ECP), 20 (384-bit Random ECP)]***].

**VPNGW11:FCS_IPSEC_EXT.1.11**

The TSF shall ensure that IKE protocols implement DH Group(s)

- 19 (256-bit Random ECP), 20 (384-bit Random ECP) according to RFC 5114 and
- [***[14 (2048-bit MODP)] according to RFC 3526***].

**NDcPP22e:FCS_IPSEC_EXT.1.12**

The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [***IKEv1 Phase 1, IKEv2 IKE_SA***] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [***IKEv1 Phase 2, IKEv2 CHILD_SA***] connection.

**NDcPP22e:FCS_IPSEC_EXT.1.13**

The TSF shall ensure that all IKE protocols perform peer authentication using [***RSA, ECDSA***] that use X.509v3 certificates that conform to RFC 4945 and [***Pre-shared Keys***].

**NDcPP22e:FCS_IPSEC_EXT.1.14**

The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: [***CN: IP address, Distinguished Name (DN)***] and [***no other reference identifier type***].

**VPNGW11:FCS_IPSEC_EXT.1.14**

The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: Distinguished Name (DN), [***CN: IP address***].

#### 5.1.3.11 Random Bit Generation (NDcPP22e:FCS_RBG_EXT.1)

**NDcPP22e:FCS_RBG_EXT.1.1**

The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [***Hash_DRBG (any)***].

**NDcPP22e:FCS_RBG_EXT.1.2**

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***[1] software-based noise source***] with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011Table C.1 'Security Strength Table for Hash Functions', of the keys and hashes that it will generate.

## 5.1.4 User data protection (FDP)

### 5.1.4.1 Full Residual Information Protection (STFFW14e:FDP_RIP.2)

**STFFW14e:FDP_RIP.2.1**

The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] all objects.

## 5.1.5 Firewall (FFW)

### 5.1.5.1 Stateful Traffic Filtering (STFFW14e:FFW_RUL_EXT.1)

**STFFW14e:FFW_RUL_EXT.1.1**

The TSF shall perform stateful traffic filtering on network packets processed by the TOE.

**STFFW14e:FFW_RUL_EXT.1.2**

The TSF shall allow the definition of stateful traffic filtering rules using the following network protocol fields:

- ICMPv4
  - Type
  - Code
- ICMPv6
  - Type
  - Code
- IPv4
  - Source address
  - Destination Address
  - Transport Layer Protocol
- IPv6
  - Source address
  - Destination Address
  - Transport Layer Protocol
  - [***no other field***]
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

and distinct interface.

**STFFW14e:FFW_RUL_EXT.1.3**

The TSF shall allow the following operations to be associated with stateful traffic filtering rules: permit or drop with the capability to log the operation.

**STFFW14e:FFW_RUL_EXT.1.4**

The TSF shall allow the stateful traffic filtering rules to be assigned to each distinct network interface.

**STFFW14e:FFW_RUL_EXT.1.5**

The TSF shall: a) accept a network packet without further processing of stateful traffic filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [***no other protocols***] based on the following network packet attributes: 1. TCP: source and destination addresses, source and destination ports, sequence number, Flags; 2. UDP: source and destination addresses, source and destination ports; 3. [***no other protocols***]. b) Remove existing traffic flows from the set of established traffic flows based on the following: [***session inactivity timeout***].

**STFFW14e:FFW_RUL_EXT.1.6**

The TSF shall enforce the following default stateful traffic filtering rules on all network traffic:

a) The TSF shall drop and be capable of [***logging***] packets which are invalid fragments;
b) The TSF shall drop and be capable of [***logging***] fragmented packets which cannot be re-assembled completely;
c) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a broadcast network;
d) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a multicast network;
e) The TSF shall drop and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
f) The TSF shall drop 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 drop 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) The TSF shall drop and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and i) [***no other rules***].

**STFFW14e:FFW_RUL_EXT.1.7**

The TSF shall be capable of dropping and logging according to the following rules:
a) The TSF shall drop and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;
b) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is a link-local address;
c) The TSF shall drop and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received.

**STFFW14e:FFW_RUL_EXT.1.8**

The TSF shall process the applicable stateful traffic filtering rules in an administratively defined order.

**STFFW14e:FFW_RUL_EXT.1.9**

The TSF shall deny packet flow if a matching rule is not identified.

**STFFW14e:FFW_RUL_EXT.1.10**

The TSF shall be capable of limiting an administratively defined number of half-open TCP connections. In the event that the configured limit is reached, new connection attempts shall be dropped and the drop event shall be [***logged***].

#### 5.1.5.2 Stateful Filtering of Dynamic Protocols (STFFW14e:FFW_RUL_EXT.2)

**STFFW14e: FFW_RUL_EXT.2.1**

The TSF shall dynamically define rules or establish sessions allowing network traffic to flow for the following network protocols: [***FTP***]

### 5.1.6 Identification and authentication (FIA)

#### 5.1.6.1 Authentication Failure Management (NDcPP22e:FIA_AFL.1)

**NDcPP22e:FIA_AFL.1.1**

The TSF shall detect when an Administrator configurable positive integer within [**1 - 120**] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

**NDcPP22e:FIA_AFL.1.2**

When the defined number of unsuccessful authentication attempts has been met, the TSF shall [***prevent the offending Administrator from successfully establishing a remote session using any***

***authentication method that involves a password until [an unlock command is issued] is taken by an Administrator, prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed***].

#### 5.1.6.2 Password Management (NDcPP22e:FIA_PMG_EXT.1)

**NDcPP22e:FIA_PMG_EXT.1.1**

The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [***'!', '@', '#', '$', '%', '&', '*', '(', ')'***];

b) Minimum password length shall be configurable to between [**8**] and [**100 on the SmartConsole and 128 on the Command Line Interface**] characters.

#### 5.1.6.3 Pre-Shared Key Composition (VPNGW11:FIA_PSK_EXT.1)

**VPNGW11:FIA_PSK_EXT.1.1**

The TSF shall be able to use pre-shared keys for IPsec and [***no other protocols***].

**VPNGW11:FIA_PSK_EXT.1.2**

The TSF shall be able to accept text-based pre-shared keys that: - Are 22 characters and [***[up to and including 64 characters]***] ; - composed of any combination of upper and lower case letters, numbers, and special characters (that include: '!', '@', '#', '$', '%', '^', '&', '*', '(', and ')').

**VPNGW11:FIA_PSK_EXT.1.3**

The TSF shall condition the text-based pre-shared keys by using [***[no conditioning]***].

**VPNGW11:FIA_PSK_EXT.1.4**

The TSF shall be able to [***accept***] bit-based pre-shared keys.

#### 5.1.6.4 Protected Authentication Feedback (NDcPP22e:FIA_UAU.7)

**NDcPP22e:FIA_UAU.7.1**

The TSF shall provide only obscured feedback to the administrative user while the authentication is in progress at the local console.

#### 5.1.6.5 Password-based Authentication Mechanism (NDcPP22e:FIA_UAU_EXT.2)

**NDcPP22e:FIA_UAU_EXT.2.1**

The TSF shall provide a local [***password-based***] authentication mechanism to perform local administrative user authentication.

#### 5.1.6.6 User Identification and Authentication (NDcPP22e:FIA_UIA_EXT.1)

**NDcPP22e:FIA_UIA_EXT.1.1**

The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [***no other actions***].

**NDcPP22e:FIA_UIA_EXT.1.2**

The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

#### 5.1.6.7 X.509 Certificate Validation (NDcPP22e:FIA_X509_EXT.1/ITT)

**NDcPP22e:FIA_X509_EXT.1.1/ITT**

The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of two certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [***Certificate Revocation List (CRL) as specified in RFC 5759 Section 5***].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

**NDcPP22e:FIA_X509_EXT.1.2/ITT**

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

#### 5.1.6.8 X.509 Certificate Validation (NDcPP22e:FIA_X509_EXT.1/Rev)

**NDcPP22e:FIA_X509_EXT.1.1/Rev**

The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [***Certificate Revocation List (CRL) as specified in RFC 5759 Section 5***]
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

**NDcPP22e:FIA_X509_EXT.1.2/Rev**

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

#### 5.1.6.9 X.509 Certificate Validation (VPNGW11:FIA_X509_EXT.1/Rev)

**VPNGW11:FIA_X509_EXT.1.1/Rev**

The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.

- The TSF shall validate the revocation status of the certificate using [***Certificate Revocation List (CRL) as specified in RFC 5759 Section 5***].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

**VPNGW11:FIA_X509_EXT.1.2/Rev**
The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

#### 5.1.6.10 X.509 Certificate Authentication (NDcPP22e:FIA_X509_EXT.2)

**NDcPP22e:FIA_X509_EXT.2.1**
The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [***IPsec***], and [***no additional uses***].

**NDcPP22e:FIA_X509_EXT.2.2**
When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [***not accept the certificate***].

#### 5.1.6.11 X.509 Certificate Authentication (VPNGW11:FIA_X509_EXT.2)

**VPNGW11:FIA_X509_EXT.2.1**
The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec and [***no other protocols***], and [***no additional uses***].

**VPNGW11:FIA_X509_EXT.2.2**
Unmodified from Base-PP.

#### 5.1.6.12 X.509 Certificate Requests (NDcPP22e:FIA_X509_EXT.3)

**NDcPP22e:FIA_X509_EXT.3.1**
The TSF shall generate a Certification Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [***Common Name, Organization, Organizational Unit, Country***].

**NDcPP22e:FIA_X509_EXT.3.2**
The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

#### 5.1.6.13 X.509 Certificate Requests (VPNGW11:FIA_X509_EXT.3)

**VPNGW11:FIA_X509_EXT.3.1**
The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [***Common Name, Organization, Organizational Unit, Country***].

**VPNGW11:FIA_X509_EXT.3.2**
The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

## 5.1.7 Security management (FMT)

### 5.1.7.1 Management of security functions behaviour (NDcPP22e:FMT_MOF.1/ManualUpdate)

**NDcPP22e:FMT_MOF.1.1/ManualUpdate**

The TSF shall restrict the ability to enable the functions to perform manual updates to Security Administrators.

### 5.1.7.2 Management of Security Functions Behaviour (NDcPP22e:FMT_MOF.1/Services)

**NDcPP22e:FMT_MOF.1.1/Services**

The TSF shall restrict the ability to start and stop services to Security Administrators.

### 5.1.7.3 Management of TSF Data (NDcPP22e:FMT_MTD.1/CoreData)

**NDcPP22e:FMT_MTD.1.1/CoreData**

The TSF shall restrict the ability to manage the TSF data to Security Administrators.

### 5.1.7.4 Management of TSF Data (NDcPP22e:FMT_MTD.1/CryptoKeys)

**NDcPP22e:FMT_MTD.1.1/CryptoKeys**

The TSF shall restrict the ability to manage the cryptographic keys to Security Administrators.

### 5.1.7.5 Management of TSF Data (VPNGW11:FMT_MTD.1/CryptoKeys)

**VPNGW11:FMT_MTD.1.1/CryptoKeys**

The TSF shall restrict the ability to manage the cryptographic keys and certificates used for VPN operation to Security Administrators.

### 5.1.7.6 Specification of Management Functions (NDcPP22e:FMT_SMF.1)

**NDcPP22e:FMT_SMF.1.1**

The TSF shall be capable of performing the following management functions:

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [***digital signature***] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [***Ability to start and stop services,***
- ***Ability to configure audit behavior (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full),***
- ***Ability to modify the behavior of the transmission of audit data to an external IT entity,***
- ***Ability to manage the cryptographic keys,***
- ***Ability to configure the cryptographic functionality,***
- ***Ability to configure the lifetime for IPsec SAs,***
- ***Ability to configure the interaction between TOE components,***
- ***Ability to re-enable an Administrator account,***
- ***Ability to set the time which is used for time-stamps;,***
- ***Ability to configure the reference identifier for the peer;,***
- ***Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors,***
- ***Ability to import X509v3 certificates to the TOE's trust store***].

#### 5.1.7.7 Specification of Management Functions (VPNGW11:FMT_SMF.1)

**VPNGW11:FMT_SMF.1.1**

The TSF shall be capable of performing the following management functions:

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using digital signature and [no other] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- Ability to manage the cryptographic keys;
- Ability to configure the cryptographic functionality;
- Ability to configure the lifetime for IPsec SAs;
- Ability to import X.509v3 certificates to the TOE's trust store;
- [***Ability to start and stop services,***
- ***Ability to configure audit behavior(e.g. changes to storage locations for audit; changes to behavior when local audit storage space is full),***
- ***Ability to modify the behavior of the transmission of audit data to an external IT entity, the handling of audit data, the audit functionality when Local Audit Storage Space is full,***
- ***Ability to configure the interaction between TOE components,***
- ***Ability to re-enable an Administrator account,***
- ***Ability to set the time which is used for time-stamps,***
- ***Ability to configure the reference identifier for the peer,***
- ***Ability to manage the TOE's trust store and designate X.509v3 certificates as trust anchors***].

#### 5.1.7.8 Specification of Management Functions (STFFW14e:FMT_SMF.1/FFW)

**STFFW14e:FMT_SMF.1.1/FFW**

The TSF shall be capable of performing the following management functions:

- Ability to configure firewall rules.

#### 5.1.7.9 Specification of Management Functions (VPN Gateway) (VPNGW11:FMT_SMF.1/VPN)

**VPNGW11:FMT_SMF.1.1/VPN**

The TSF shall be capable of performing the following management functions:

- Definition of packet filtering rules;
- Association of packet filtering rules to network interfaces;
- Ordering of packet filtering rules by priority;
- [***No other capabilities***].

#### 5.1.7.10 Restrictions on Security Roles (NDcPP22e:FMT_SMR.2)

**NDcPP22e:FMT_SMR.2.1**

The TSF shall maintain the roles: - Security Administrator.

**NDcPP22e:FMT_SMR.2.2**

The TSF shall be able to associate users with roles.

**NDcPP22e:FMT_SMR.2.3**

The TSF shall ensure that the conditions

- The Security Administrator role shall be able to administer the TOE locally;
- The Security Administrator role shall be able to administer the TOE remotely

are satisfied.

## 5.1.8 Packet Filtering (FPF)

### 5.1.8.1 Rules for Packet Filtering (VPNGW11:FPF_RUL_EXT.1)

**VPNGW11:FPF_RUL_EXT.1.1**
The TSF shall perform Packet Filtering on network packets processed by the TOE.

**VPNGW11:FPF_RUL_EXT.1.2**
The TSF shall allow the definition of Packet Filtering rules using the following network protocols and protocol fields:

- 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.

**VPNGW11:FPF_RUL_EXT.1.3**
The TSF shall allow the following operations to be associated with Packet Filtering rules: permit and drop with the capability to log the operation.

**VPNGW11:FPF_RUL_EXT.1.4**
The TSF shall allow the Packet Filtering rules to be assigned to each distinct network interface.

**VPNGW11:FPF_RUL_EXT.1.5**
The TSF shall process the applicable Packet Filtering rules (as determined in accordance with VPNGW11:FPF_RUL_EXT.1.4) in the following order: Administrator-defined.

**VPNGW11:FPF_RUL_EXT.1.6**
The TSF shall drop traffic if a matching rule is not identified.

## 5.1.9 Protection of the TSF (FPT)

### 5.1.9.1 Protection of Administrator Passwords (NDcPP22e:FPT_APW_EXT.1)

**NDcPP22e:FPT_APW_EXT.1.1**
The TSF shall store administrative passwords in non-plaintext form.

**NDcPP22e:FPT_APW_EXT.1.2**
The TSF shall prevent the reading of plaintext administrative passwords.

### 5.1.9.2 Fail Secure (Self-Test Failures) (VPNGW11:FPT_FLS.1/SelfTest)

**VPNGW11:FPT_FLS.1.1/SelfTest**
The TSF shall shut down when the following types of failures occur: failure of the power-on self-tests, failure of integrity check of the TSF executable image, failure of noise source health tests.

### 5.1.9.3 Basic internal TSF data transfer protection (NDcPP22e:FPT_ITT.1)

**NDcPP22e:FPT_ITT.1.1**
The TSF shall protect TSF data from disclosure and detect its modification when it is transmitted between separate parts of the TOE through the use of [***IPsec***].

#### 5.1.9.4 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) (NDcPP22e:FPT_SKP_EXT.1)

**NDcPP22e:FPT_SKP_EXT.1.1**
The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

#### 5.1.9.5 Reliable Time Stamps (NDcPP22e:FPT_STM_EXT.1)

**NDcPP22e:FPT_STM_EXT.1.1**
The TSF shall be able to provide reliable time stamps for its own use.
**NDcPP22e:FPT_STM_EXT.1.2**
The TSF shall [***allow the Security Administrator to set the time***].

#### 5.1.9.6 TSF testing (NDcPP22e:FPT_TST_EXT.1)

**NDcPP22e:FPT_TST_EXT.1.1**
The TSF shall run a suite of the following self-tests [***during initial start-up (on power on)***] to demonstrate the correct operation of the TSF: [**TSF integrity verification, cryptographic algorithm verification**].

#### 5.1.9.7 TSF Testing (VPNGW11:FPT_TST_EXT.1)

**VPNGW11:FPT_TST_EXT.1.1**
The TSF shall run a suite of the following self-tests [***during initial startup (on power on)***] to demonstrate the correct operation of the TSF: noise source health tests, [**TSF integrity verification, cryptographic algorithm verification**].

#### 5.1.9.8 TSF Self-Test with Defined Methods (VPNGW11:FPT_TST_EXT.3)

**VPNGW11:FPT_TST_EXT.3.1**
The TSF shall run a suite of the following self-tests when loaded for execution to demonstrate the correct operation of the TSF: integrity verification of stored executable code.
**VPNGW11:FPT_TST_EXT.3.2**
The TSF shall execute the self-testing through a TSF-provided cryptographic service specified in FCS_COP.1/SigGen.

#### 5.1.9.9 Trusted update (NDcPP22e:FPT_TUD_EXT.1)

**NDcPP22e:FPT_TUD_EXT.1.1**
The TSF shall provide Security Administrators the ability to query the currently executing version of the TOE firmware/software and [***no other TOE firmware/software version***].
**NDcPP22e:FPT_TUD_EXT.1.2**
The TSF shall provide Security Administrators the ability to manually initiate updates to TOE firmware/software and [***no other update mechanism***].
**NDcPP22e:FPT_TUD_EXT.1.3**
The TSF shall provide means to authenticate firmware/software updates to the TOE using a [***digital signature***] prior to installing those updates.

#### 5.1.9.10 Trusted Update (VPNGW11:FPT_TUD_EXT.1)

**VPNGW11:FPT_TUD_EXT.1.1**
Unmodified from Base-PP.
**VPNGW11:FPT_TUD_EXT.1.2**
Unmodified from Base-PP.
**VPNGW11:FPT_TUD_EXT.1.3**
The TSF shall provide means to authenticate firmware/software updates to the TOE using a digital signature mechanism and [***no other mechanisms***] prior to installing those updates.

## 5.1.10 TOE access (FTA)

### 5.1.10.1 TSF-initiated Termination (NDcPP22e:FTA_SSL.3)

**NDcPP22e:FTA_SSL.3.1**
The TSF shall terminate a remote interactive session after a Security Administrator-configurable time interval of session inactivity.

### 5.1.10.2 User-initiated Termination (NDcPP22e:FTA_SSL.4)

**NDcPP22e:FTA_SSL.4.1**
The TSF shall allow Administrator-initiated termination of the Administrator's own interactive session.

### 5.1.10.3 TSF-initiated Session Locking (NDcPP22e:FTA_SSL_EXT.1)

**NDcPP22e:FTA_SSL_EXT.1.1**
The TSF shall, for local interactive sessions, [***terminate the session***] after a Security Administrator-specified time period of inactivity.

### 5.1.10.4 Default TOE Access Banners (NDcPP22e:FTA_TAB.1)

**NDcPP22e:FTA_TAB.1.1**
Before establishing an administrative user session the TSF shall display a Security Administrator-specified advisory notice and consent warning message regarding use of the TOE.

## 5.1.11 Trusted path/channels (FTP)

### 5.1.11.1 Inter-TSF trusted channel (NDcPP22e:FTP_ITC.1)

**NDcPP22e:FTP_ITC.1.1**
The TSF shall be capable of using [***IPsec***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***VPN communications***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

**NDcPP22e:FTP_ITC.1.2**
The TSF shall permit the TSF or the authorized IT entities to initiate communication via the trusted channel.

**NDcPP22e:FTP_ITC.1.3**
The TSF shall initiate communication via the trusted channel for [**audit service, VPN communications**].

### 5.1.11.2 Inter-TSF Trusted Channel (VPN Communications) (VPNGW11:FTP_ITC.1/VPN)

**VPNGW11:FTP_ITC.1.1/VPN**
The TSF shall be capable of using IPsec to provide a communication channel between itself and authorized IT entities supporting VPN communications that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

**VPNGW11:FTP_ITC.1.2/VPN**
The TSF shall permit the authorized IT entities to initiate communication via the trusted channel.

**VPNGW11:FTP_ITC.1.3/VPN**
The TSF shall initiate communication via the trusted channel for [**remote VPN gateways/peers**].

#### 5.1.11.3 Trusted Path (NDcPP22e:FTP_TRP.1/Admin)

**NDcPP22e:FTP_TRP.1.1/Admin**
The TSF shall be capable of using [***IPsec***] to provide a communication path between itself and authorized remote Administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and provides detection of modification of the channel data.

**NDcPP22e:FTP_TRP.1.2/Admin**
The TSF shall permit remote Administrators to initiate communication via the trusted path.

**NDcPP22e:FTP_TRP.1.3/Admin**
The TSF shall require the use of the trusted path for initial Administrator authentication and all remote administration actions.

#### 5.1.11.4 Trusted Path (NDcPP22e:FTP_TRP.1/Join)

**NDcPP22e:FTP_TRP.1.1/Join**
The TSF shall provide a communication path between itself and a joining component that is logically distinct from other communication paths and provides assured identification of [***both joining component and TSF endpoint***] and protection of the communicated data from modification [***none***].

**NDcPP22e:FTP_TRP.1.2/Join**
The TSF shall permit [***the TSF***] to initiate communication via the trusted path.

**NDcPP22e:FTP_TRP.1.3/Join**
The TSF shall require the use of the trusted path for joining components to the TSF under environmental constraints identified in [**CC Guide Section 5**].

## 5.2 TOE Security Assurance Requirements

The SARs for the TOE are the components as specified in Part 3 of the Common Criteria. Note that the SARs have effectively been refined with the assurance activities explicitly defined in association with both the SFRs and SARs.

| Requirement Class | Requirement Component |
|---|---|
| **ADV: Development** | ADV_FSP.1: Basic Functional Specification |
| **AGD: Guidance documents** | AGD_OPE.1: Operational User Guidance |
| | AGD_PRE.1: Preparative Procedures |
| **ALC: Life-cycle support** | ALC_CMC.1: Labelling of the TOE |
| | ALC_CMS.1: TOE CM Coverage |
| **ATE: Tests** | ATE_IND.1: Independent Testing - Conformance |
| **AVA: Vulnerability assessment** | AVA_VAN.1: Vulnerability Survey |
| | AVA_VLA.1: Additional Flaw Hypotheses |

**Table 5-3 Assurance Components**

### 5.2.1 Development (ADV)

#### 5.2.1.1 Basic Functional Specification (ADV_FSP.1)

**ADV_FSP.1.1d**
The developer shall provide a functional specification.

**ADV_FSP.1.2d**
The developer shall provide a tracing from the functional specification to the SFRs.

**ADV_FSP.1.1c**

The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

**ADV_FSP.1.2c**

The TSF shall support mutual authentication of TLS clients using X.509v3 certificates.

**ADV_FSP.1.3c**

The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

**ADV_FSP.1.4c**

The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

**ADV_FSP.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**ADV_FSP.1.2e**

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

### 5.2.2 Guidance documents (AGD)

#### 5.2.2.1 Operational User Guidance (AGD_OPE.1)

**AGD_OPE.1.1d**

The developer shall provide operational user guidance.

**AGD_OPE.1.1c**

The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

**AGD_OPE.1.2c**

The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

**AGD_OPE.1.3c**

The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

**AGD_OPE.1.4c**

The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

**AGD_OPE.1.5c**

The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences, and implications for maintaining secure operation.

**AGD_OPE.1.6c**

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

**AGD_OPE.1.7c**

The operational user guidance shall be clear and reasonable.

**AGD_OPE.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

#### 5.2.2.2 Preparative Procedures (AGD_PRE.1)

**AGD_PRE.1.1d**
The developer shall provide the TOE, including its preparative procedures.

**AGD_PRE.1.1c**
The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

**AGD_PRE.1.2c**
The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

**AGD_PRE.1.1e**
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**AGD_PRE.1.2e**
The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

### 5.2.3 Life-cycle support (ALC)

#### 5.2.3.1 Labelling of the TOE (ALC_CMC.1)

**ALC_CMC.1.1d**
The developer shall provide the TOE and a reference for the TOE.

**ALC_CMC.1.1c**
The TOE shall be labelled with its unique reference.

**ALC_CMC.1.1e**
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

#### 5.2.3.2 TOE CM Coverage (ALC_CMS.1)

**ALC_CMS.1.1d**
The developer shall provide a configuration list for the TOE.

**ALC_CMS.1.1c**
The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

**ALC_CMS.1.2c**
The configuration list shall uniquely identify the configuration items.

**ALC_CMS.1.1e**
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

### 5.2.4 Tests (ATE)

#### 5.2.4.1 Independent Testing - Conformance (ATE_IND.1)

**ATE_IND.1.1d**
The developer shall provide the TOE for testing.

**ATE_IND.1.1c**
The TOE shall be suitable for testing.

**ATE_IND.1.1e**
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**ATE_IND.1.2e**

The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

## 5.2.5 Vulnerability assessment (AVA)

### 5.2.5.1 Vulnerability Survey (AVA_VAN.1)

**AVA_VAN.1.1d**

The developer shall provide the TOE for testing.

**AVA_VAN.1.1c**

The TOE shall be suitable for testing.

**AVA_VAN.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**AVA_VAN.1.2e**

The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

**AVA_VAN.1.3e**

The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

# 6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Communication
- Cryptographic support
- User data protection
- Firewall
- Identification and authentication
- Security management
- Packet Filtering
- Protection of the TSF
- TOE access
- Trusted path/channels

When the term TOE is used, it refers to the all parts of the distributed TOE. If a specific component is being described, it will be named. Throughout this section the term "Security Gateway" is used to refer to both Check Point Security Gateway appliances and Maestro Hyperscale appliances running R81.00 software.

## 6.1 Security audit

The TOE generates audit events (see Table 5-2 Audit Events) and has the capability to store them internally or export them. The TOE stores its internal audit events in a log that is protected so that only the authorized administrator can read the audit events.

The TOE components can be configured to use IPsec to send audit records to the syslog server directly. This transmission happens in real-time.

The TOE has a disk cleanup procedure where it removes old audit logs to allow space for new ones. This is configurable by the authorized administrator. When disk space on the TOE falls below a predefined threshold (the cleanup procedure cannot keep up with the audit collection), the server stops collecting audit records. Each Check Point appliance provides drive space reserved for audit logs. The minimum available space for the CloudGuard (ESXi) appliance is 1GB, while other appliances have minimum storage space ranging from 72GB to 13TB.

Security Gateways and Security Management Servers maintain a queue of log records generated on the Security Gateway and Security Management Server in memory, while they are being transmitted over the network to the defined log servers. If this queue is overrun, i.e. if the Security Gateway and Security Management Server consistently generates log records faster than they can be received by the log server, or if there is a connectivity failure to the log server, the Security Gateway and Security Management Server stores the queued records in local log files, so that no log records are lost.

In the event of failure, e.g. loss of power on the Security Gateway or Security Gateway and Security Management Server, queued audit records that have not been successfully transmitted to the log server may be lost. The maximum number of records that may be lost is equal to the queue size: 4096 records.

The TOE includes a hardware clock that is used to provide reliable time information for the audit records it generates.

The TOE audits all IPsec failures including IKE Auth exchange failures due to revoked certificates and authentication failures. When an admin creates a certificate the TOE's audit includes (among other things) the DN, the Certificate Authority along with a gateway object to which the certificate is attached. Thereafter, the TOE associates that certificate with the specific gateway and a unique set of values can be used to trace the certificate from addition to deletion. The values includes the Vpn.vpnClientsSettingsForGateway.usb1VpnClientSettings and the Object name.

The Security audit function satisfies the following security functional requirements:

- NDcPP22e:FAU_GEN.1, STFFW14e:FAU_GEN.1, VPNGW11:FAU_GEN.1 and NDcPP22e:FAU_GEN_EXT.1: The TOE is able to generate logs for a range of events as required by the requirements in Section 5.1.1. Each event log is unique with the date/time of the event, type of event, subject identity (e.g. IP address), and the outcome of the event.
- NDcPP22e:FAU_GEN.2: The TOE is able to identify each auditable event with specific IP addresses and the TOE's interfaces and gateways.
- NDcPP22e:FAU_STG_EXT.1: The TOE is able to send audit log data to an external audit server. The connection to the audit server is encapsulated in an IPsec tunnel. The Security Gateways can send audit records to the Security Management Server for export or they can export them to a syslog server directly.
- NDcPP22e:FAU_STG_EXT.4: The TOE stores audit data locally until it is exported. In the event that audit records are generated and no space is available to store the records locally, the TOE discards the new audit data.
- NDcPP22e:FAU_STG_EXT.5: As described above, the TOE can store audit records locally prior to transmitting the audit data to an external audit server. Transmission of audit records is protected either by the IPsec connection between TOE components (FPT_ITT.1) or by an IPsec to an external audit server (FTP_ITC.1).

## 6.2 Communication

The TOE is a distributed solution consisting of Security Gateway and Security Management Appliances. The Security Management Appliances can manage one or more Security Gateways. The Administrator is responsible for establishing an IPsec channel between the Security Gateways Gateway Appliance and Security Management Appliance.

The administrator must join Security Gateways to a Security Management Server by creating a new Gateway object within the Security Management Server, and then creates a shared secret (passphrase) unique to the new Gateway object, and then configuring the new Security Gateway to expect a connection from the Security Management Server (along with the IP address and shared secret). After this, the Administrator can join the Security Gateway from the Management Server, thereby enabling communication between the Gateway and Management Server. This new connection between the Management Server and Security Gateway is called the Secure Internal Communication (SIC). The Administrator can disable communication between the Management Server and a chosen Gateway by resetting the Secure Internal Communication (SIC) between the two or by removing the Gateway object entirely. However, the TOE does not allow management communications between or among Gateways, and only allows communication between the Security Gateways and Management Server for management and configuration purposes.

The Communication function satisfies the following security functional requirements:

- NDcPP22e:FCO_CPC_EXT.1: The Administrator establishes a connection between the two TOE components using a channel that meets the secure registration channel requirements in FTP_TRP.1/Join before those components can communicate. Please see FTP_TRP.1/Join for more details.

## 6.3 Cryptographic support

The TOE uses the Check Point Cryptographic Library version 1.1 that has been CAVP tested. The following functions have been CAVP tested to meet the associated SFRs.

| Requirement | Functions | Standard | Cert # |
|---|---|---|---|
| Key Generation | | | |
| NDcPP22e:FCS_CKM.1<br>VPNGW11:FCS_CKM.1/IKE | RSA Key Generation (2048-bit, 4096-bit) | FIPS 186-4 | A2365 |
| | ECDSA Key Generation (P-256, P-384 and P-521) | | A2365 |
| | FFC Schemes using “safe-prime” groups | SP 800-56A Revision 3 | No NIST CAVP |
| Key Establishment | | | |
| NDcPP22e:FCS_CKM.2 | RSA Key Exchange | Vendor affirm 800-56B | Vendor assertion |
| | ECC Key Exchange<br>Curves P-256, P-384 and P-521 | SP 800-56A,<br>CVL KAS ECC | A2365 |
| | FFC Schemes using “safe-prime” groups | SP 800-56A Revision 3 | No NIST CAVP |
| Encryption/Decryption | | | |
| FCS_COP.1/DataEncryption | AES CBC (128 and 256 bits)<br>AES GCM (128 and 256 bits) | FIPS 197, SP 800-38A/D | A2365 |
| Cryptographic signature services | | | |
| FCS_COP.1/SigGen | RSA Digital Signature Algorithm (rDSA) (modulus 2048) | FIPS 186-4<br>SigGen & SigVer | A2365 |
| | ECDSA Signature Algorithm (ECDSA) P-256, P-384 and P-521 | | A2365 |
| Cryptographic hashing | | | |
| FCS_COP.1/Hash | SHA-1, SHA-256, SHA-384, SHA-512 (digest sizes 160, 256, 384, 512) | FIPS 180-4 | A2365 |
| Keyed-hash message authentication | | | |
| FCS_COP.1/KeyedHash | HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 HMAC-SHA-512 (digest sizes 160, 256, 384, 512) | FIPS 198-1 &<br>FIPS 180-4 | A2365 |
| Random bit generation | | | |
| FCS_RBG_EXT.1 | HASH_DRBG with sw based noise sources with a minimum of 256 bits of non-determinism | SP 800-90A<br>(Hash_DRBG) | A2365 |

**Table 6-1 CAVP Algorithms & Certificates**

The TOE provides key generation for asymmetric keys on all components and can generate ECDSA keys using NIST curve sizes P-256, P-384, and P-521 and RSA keys of size 2048 and 4096-bits [according to FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4 and B.3 respectively]. The TOE supports DH group 14 key establishment scheme that meets standard RFC 3526, section 3 (for interoperability) as well as group 19 (ECP-256) and group 20 (ECP-384).

The TOE uses a software-based random bit generator that complies with Special Publication 800-90 using Hash_DRBG when operating in the FIPS mode (which is a subset of CC mode). AES-256 is used in conjunction with a minimum of 256 bits of entropy.

The TOE’s Gateways implement the IPsec architecture as specified in RFC 4301. SPD rules can be configured using the firewall rules and VPN communities1. Firewall rules are used to distinguish between DROP actions and others, while VPN communities distinguish between traffic that is encrypted (PROTECT) and traffic that is not (BYPASS).

VPN communities control how allowed traffic is allowed to flow between gateways. If traffic is part of a VPN community, it will be encrypted and then firewall rules will be applied. Rules are explicit, therefore any packet not matching a rule will be dropped. Rules are processed in order with the first matching rule being applied to the traffic. The TOE supports IKEv1 in tunnel mode and IKEv2 in tunnel mode. When using IKEv1, the TOE does not utilize aggressive mode and exclusively uses main mode. The authorized administrator can configure the TOE to support maximum lifetimes for IKEv1 and IKEv2 SAs based on elapsed time. The administrator can specify (in minutes) the maximum lifetime of the Phase 1/IKE SA and specify (in seconds) the maximum lifetime of the Phase 2/ESP SA.

The TOE implements RFC 4106 conformant AES-GCM-128 and AES-GCM-256, and RFC 3602 conformant AES-CBC-128, and AES-CBC-256 as encryption algorithms for ESP. The TOE also implements HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 as integrity/authentication algorithms as well as Diffie-Hellman Groups 14, 19, and 20. The administrator configures the order the groups will be negotiated with a peer. The encrypted payload for IKEv1 and IKEv2 uses AES-CBC-128, AES-CBC-256 as specified in RFC 3602. The TOE generates the secret value x used in the IKEv1 and IKEv2 Diffie-Hellman key exchange ('x' in gx mod p) using the Hash_DRBG specified in FCS_RBG_EXT.1 and having possible lengths of 224, 256, or 384 bits. The TOE generates nonces used in the IKEv1 and IKEv2 exchanges of at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash. The TOE verifies that the default the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the IKEv1 Phase 1/IKEv2 IKE_SA connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the IKEv1 Phase 2/IKEv2 CHILD_SA connection.

The TOE's IPsec implementation supports Pre-Shared Keys (PSKs) and X.509 certificates (both RSA and ECDSA) for IKE authentication.

The TOE's Gateways use IPsec to protect communications with external IT entities (a remote administrator workstation, or a syslog server) as well as for internal communications (between the TOE's distributed components). The TOE's Security Management Server tunnels its ITC traffic through IPsec provided by its paired Gateway.

The following table presents the crypto security parameters (CSPs), secret keys, and private keys provided by the TOE. The table also identifies when each CSP or key is cleared. For IPsec related keys stored in memory, they are cleared when the session expires by zeroing the memory with system calls before freeing the memory.

| CSP or Key | Stored In | Zeroized upon: | Zeroized by: |
|---|---|---|---|
| VPN IKE_SA Keys (Auth initiator and responder, Encryption initiator and responder) | Memory | When IKE SA Expired | Overwriting with zeros |
| VPN CHILD/IPSEC_SA Keys (Initiator and Responder) | Memory | When child or IKE SA expired | Overwriting with zeros |
| User IPsec X.509v3 Certs (ECDSA) (public) | On Disk | N/A – Public information | N/A – Public information |
| Gateway IPsec X.509v3 Certs (ECDSA) (public) | On Disk | N/A – Public information | N/A – Public information |
| Gateway IPsec X.509v3 Certs (ECDSA) (public) | On Disk | N/A – Public information | N/A – Public information |
| VPN PSK | On Disk | Never (may be replaced | |
| Password hash | On Disk | Never (may be replaced) | |

**Table 6-2 CSP & Keys**

Since the TOE supports only the IPsec security protocol, IPsec is used for all RSA and ECDSA key establishment for VPN connections, trusted channels and remote administration.

The Cryptographic support function satisfies the following security functional requirements:

- NDcPP22e:FCS_CKM.1 and VPNGW11:FCS_CKM.1/IKE: The TOE supports RSA and ECDSA key generation. The key generation is used by the TOE when it creates a Certificate Signing Request (CSR) to apply for a certificate from the Certificate Authority (CA). The TOE enforces a key size of 2048-bit and

4096-bit for RSA and the NIST curves P256, P384 and P521 for ECDSA key pairs. The scheme is used by IPsec.

- NDcPP22e:FCS_CKM.2: See NDcPP22e:FCS_CKM.1 for RSA and ECDSA keys. Additionally, the TOE implementation of Diffie-Hellman-group-14 meets RFC 3526, Section 3 by virtue of using a 2048-bit MODP group for key establishment.
- NDcPP22e:FCS_CKM.4: Keys are cleared by overwriting the storage location with zeros when they are no longer needed by the TOE, refer to Table 6-2.
- NDcPP22e:FCS_COP.1/DataEncryption and VPNGW11:FCS_COP.1/DataEncryption: The TOE performs encryption and decryption using AES in CBC and GCM mode with key sizes of either 128 or 256. The corresponding CAVP certificate is identified in the table above.
- NDcPP22e:FCS_COP.1/Hash: The TOE supports cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512, with digest sizes 160, 256, 384, and 512. The corresponding CAVP certificate is identified in the table above.
- NDcPP22e:FCS_COP.1/KeyedHash: The TOE supports keyed-hash message authentication using HMAC-SHA-1/256/384/512 using SHA-1/256/384/512 with 160/256/384/512-bit keys to produce a 160/256/384/512 output MAC (all respectively listed). The SHA-1 and SHA-256 algorithm has block sizes of 512-bits while SHA384 and SHA512 has a block size of 1024. The corresponding CAVP certificate is identified in the table above.
- NDcPP22e:FCS_COP.1/SigGen: The TOE supports the use of RSA with a 2048-bit and 4096-bit modulus or ECDSA with curves P-256, P-384, P-521 cryptographic signatures. Digital signatures are used on product updates. The corresponding CAVP certificate is identified in the table above.
- NDcPP22e:FCS_IPSEC_EXT.1 and VPNGW11:FCS_IPSEC_EXT.1: The TOE supports IPsec when exporting audit logs to an external server, when performing remote administration and when providing VPN gateway functionality. Refer to details of the IPsec protocol support provided above.
- NDcPP22e:FCS_RBG_EXT.1: The product uses an AES-256 Hash_DRBG with a software based noise source with a minimum of 256 bits of non-determinism.

## 6.4 User data protection

When an incoming network frame is received by the TOE, it is written by the network interface controller into kernel message buffers. Each kernel buffer is associated with a separate header that keeps track of the number of bytes of data in the buffer. The kernel clears the header prior to reading new data, and the header is updated with the count of bytes transferred by the controller.

When the buffer resource is abstracted into a message object, the object is initialized to refer only to data that has actually been overwritten in the context of the current message. This ensures that any residual information that might remain in the kernel buffer resource from previous messages is made unavailable.

State information resources that are allocated as part of the packet processing are cleared before use. This ensures that residual information that might remain from another packet is not retained.

The User data protection function satisfies the following security functional requirements:

- STFFW14e:FDP_RIP.2: The TOE ensures that previous information contents of resources used for new objects are not discernible in any new object, such as network packets, as described above.

## 6.5 Firewall

Every IPv4/v6 packet received by a Security Gateway is intercepted by the firewall kernel. Fragmented packets are first reassembled. IPv4/v6 packets with unauthorized IP options (e.g. source route option) are dropped.

The TOE supports logical interfaces. The logical interface over which the packet was received determines the Virtual System identifier (VSID). The default VSID is 0. Each Virtual System (VS) maintains its own tables, in which only it's associated (physical and logical) interfaces are registered. Each VS is allocated an independent set of processes for information flow processing within its context. An incoming packet is dispatched for processing by the corresponding Virtual System, determining the selection of the state tables and security policy that will be used to process the packet.

When an IP packet is received on a network interface, its source address is compared to topology information configured by the authorized administrator. If the source address does not correspond to the set of network addresses that match the given network interface, the packet is dropped as a spoofed packet. Note that broadcast and loopback addresses are never considered valid source addresses and are therefore rejected.

ESP-encapsulated packets are first decrypted and verified. The packet header attributes are used to match the packet against state tables that contain accepted FTP 'connections'. If the packet is successfully matched and passes packet sanity checks (correct sequence number, acknowledgment number, flags (SYN; ACK; RST; FIN.), then it is concluded that a decision has been already made for this traffic flow, and processing may skip past inspection. New ftp connections are tracked and flags in a state table are used to know when to clear the connection. The state table is cleared when the connection is closed. The TOE maintains and updates the state table to keep track of creation, open, and removal sessions. To help determine whether a packet can be part of a new session or an established session, the TOE uses information in the packet header and protocol header fields to determine the session state to which the FTP packet applies.

For all other packets (non-FTP), inspection is performed against the firewall rules. The rules have 4 possible outcomes:

1. Accept - the packet is allowed through;
2. Drop – the packet is dropped without notification to the sender;
3. Reject – the packet is dropped and the presumed sender is notified.
4. If no rule is matched, packets are dropped.

Firewall rules can be set to filter on protocol, source address, destination address, source port, destination port, ICMP type or ICMP code. All protocols including icmpv4, icmpv6, ipv4, ipv6, tcp, and udp may be used in firewall rules. If any interface is overwhelmed with traffic, it will drop the packets. An administrator can configure logging for a rule by specifying "Log" under the "Track" column of the firewall rule.

The firewall will drop all of the following types of packets and may optionally log them if configured to do so:

1. Packets which are invalid fragments, including a description of what constitutes an invalid fragment
2. Fragments that cannot be completely re-assembled
3. Packets where the source address is equal to the address of the network interface where the network packet was received
4. 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
5. Packets where the source address is defined as being on a broadcast network
6. Packets where the source address is defined as being on a multicast network
7. Packets where the source address is defined as being a loopback address
8. Packets where the source or destination address of the network packet is a link-local address
9. Packets where the source or destination address of the network packet s defined as being an address 'reserved for future use' as specified in RFC 5735 for IPv4
10. 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' as specified in RFC 3513 for IPv6
11. Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified

During the Security Gateway boot process, there is a lag between the time when the network interface is operational, and the time that the Stateful Traffic Filtering functionality is fully functioning. During this time, Boot Security is enforced:

- Traffic flow through the appliance is disabled; and

- Traffic to and from the appliance is controlled by a Default Filter that drops all external traffic to the appliance

The TOE provides "SYN Attack" protections that allow the administrator to configure tracking of half-open TCP connections for all hosts protected by the Security Gateway. Upon configuring a threshold number of half-open TCP connections (default of 1000), the TOE, upon detecting that number of SYN requests and the corresponding SYN-ACK responses (from a host), drops subsequent TCP SYN packets destined for the host. The TOE ages and removes half-open TCP connections based upon the administratively configured global "TCP start timeout" (default of 25 seconds). After expiring enough half-open TCP sessions, the TOE stops dropping new TCP SYN packets until the threshold is exceeded.

The Firewall function satisfies the following security functional requirements:

- STFFW14e:FFW_RUL_EXT.1: The TOE supports all of the required protocols, which include icmpv4 (RFC 792), icmpv6 (RFC 4443), ipv4 (RFC 79 1), ipv6 (RFC 2460), tcp (RFC 793), and udp (RFC 768). Conformance with the RFCs defining these protocols is asserted by Check Point based upon Check Point's implementation and design. The firewall rules implement the SPD rules (permit, deny, bypass). Each rule can be configured to log status of packets pertaining to the rule. All codes under each protocol are implemented. Rules can also be assigned to each network interface. The TOE is able to create one or more VS. Physical interfaces on the TOE hardware can be assigned to each VS. From the rules page, each rule can specify one or more VS. The TOE supports FTP for stateful filtering. The TOE's firewall rules apply to all IP ranges. These rules take precedence in layer 2. Rules applied to specific IP addresses are specific to all gateways. These rules take precedence in layer 3. This allows the TOE's gateway firewall rules to be checked first.

## 6.6 Identification and authentication

The TOE provides a password mechanism for authenticating users. Users are associated with a username, password, and one or more roles. Users may authenticate locally via a serial connection for the CLI or remotely via the SmartConsole application whose connection is encapsulated over IPsec. Passwords can be composed of any alphabetic, numeric, and a wide range of special characters (identified in FIA_PMG_EXT.1 as follows: ***'!', '@', '#', '$', '%', '&', '*', '(', ')'***). Passwords are not echoed back when users logon to the TOE. Internally the TOE keeps track of failed login attempts. If an administrator fails for a configured number of attempts, the administrator is either locked out for a period of time or until the primary administrator unlocks the account. The primary administrator can always log into the device via the local serial CLI connection and can never be locked out from this login.

The TOE requires identification and authentication before allowing access. Only the banner may be presented before authentication is complete. Administrators log into the Security Management Server and manage the associated Security Gateways.

The TOE supports the use of a pre-shared key between the length of 22-64 characters for IPsec authentication. Text based pre-shared keys are not conditioned and the TOE can accept bit-based pre-shared keys as well. The pre-shared key may contain any combination of upper and lower case letters, numbers, and special characters as specified: ('!', '@', '#', '$', '%', '^', '&', '*', '(', and ')'). When using pre-shared keys for authentication, the IPsec endpoints must both be configured to use the same key

The TOE supports X.509v3 certificates for IPsec authentication as well. X.509v3 certificates are stored internally and the store is protected by file permissions. X.509 certificates are manually loaded by the authorized administrator onto the TOE by an administrator.

The authorized administrator configures the VPN peers for administrator and VPN communications, and specifies the DN associated with an IP. When an incoming request comes in, the TOE matches the peer's IP address to its configuration, to find the correct rule and then match the configured DN to the peer certificate. The TOE then validates that it can construct a certificate path from the client's certificate through any intermediary CAs to the CA certificate specified by the user in the VPN configuration. If the TOE can successfully build the certificate path, then the TOE will next check the validity of the certificates (e.g., checking its validity dates and that the CA flag is present in the basic constraints section for all CA certs). There are no exceptions to the rules for extendedKeyUsage

fields. Assuming the certificates are valid, the TOE finally checks the revocation status of all. The TOE will reject any certificate for which it cannot determine the validity and reject the connection attempt. For internal TOE communications (ITT), the TOE automatically associates each distributed component's IP address with its certificate DN.

The Identification and authentication function satisfies the following security functional requirements:

- NDcPP22e:FIA_AFL.1: The TOE allows an administrator to configure a locking policy for administrative logins to SmartConsole (the TOE does not lock out its administrative serial interface). In addition to specifying the maximum number of incorrect attempts, the administrator can specify whether the TOE should unlock the Administrator's account (after a configurable number of minutes), and if not, then the Administrator's account remains locked until another Administrator with sufficient privileges unlocks the affected account. The TOE allows an administrator to set the number of failed attempts to a value from 1-120 and to enable a lock-out time between 1-120 minutes. Again, the local CLI remains available when the remote account is locked out.
- NDcPP22e:FIA_PMG_EXT.1: The TOE supports passwords of varying length and allows an administrator to specify a minimum password length between 8 and 100[1] characters long. The password composition can contain all of the special characters required by this requirement.
- VPNGW11:FIA_PSK_EXT.1: The TOE can support a pre-shared key length of 22-64 characters. Text based pre-shared keys are not conditioned and the TOE can accept bit-based pre-shared keys. The pre-shared key composition can contain all of the special characters required by this requirement.
- NDcPP22e:FIA_UAU.7: Authentication data entered in by an administrator is obscured during login.
- NDcPP22e:FIA_UAU_EXT.2: The TOE's authentication mechanism employs a locally stored database of authentication data.
- NDcPP22e:FIA_UIA_EXT.1: The TOE is able to display a warning banner in accordance with FTA_TAB.1.
- NDcPP22e:FIA_X509_EXT.1/ITT: Certificates are validated between TOE components. This includes revocation checking.
- NDcPP22e:FIA_X509_EXT.1/Rev and VPNGW11:FIA_X509_EXT.1/Rev: CRLs are supported for X509v3 certificate validation. Certificates are validated as part of the authentication process when they are presented to the TOE and when they are loaded into the TOE. All certificate validation checks are the same regardless of the certificate type.
- NDcPP22e:FIA_X509_EXT.2 and VPNGW11:FIA_X509_EXT.2: Certificates are checked and if found not valid are not accepted or if the CRL server cannot be contacted for validity checks, then the certificate is not accepted
- NDcPP22e:FIA_X509_EXT.3 and VPNGW11:FIA_X509_EXT.3: The TOE generates certificate requests and validates the CA used to sign the certificates.

## 6.7 Security management

User accounts are associated with roles. User accounts associated with all privileges in their role are called authorized administrators. Authorized Administrator can access audit configuration data, firewall and VPN settings, user and administrator security attributes (including passwords), warning banner configuration, and cryptographic support settings.

The TOE offers two administrative interfaces – command line and GUI. The TOE offers command line functions which are accessible via the CLI. The CLI is a text-based interface which can only be accessed from a directly

[1]The maximum password length is 100 on the SmartConsole and 128 on the Command Line Interface.

connected terminal. The CLI interface can be accessed on the Security Management Server and each individual Gateway. While the CLI contains much of the base functionality needed to configure the management server and Gateways, it is recommended to use SmartConsole as its available commands are all inclusive of the management server settings. Additionally, changes made via SmartConsole can be made once, in one location, and be pushed to each device in the TOE's topology. However, many individual TOE configurations (primarily during first set-up) must be done via a CLI connection.

The TOE also offers a GUI interface from its Security Management Server accessible via TLS over IPsec for management. The SmartConsole offers access to the same function types as the CLI and can be used either locally or remotely. Typically, most authorized administrators use the GUI interface for management.

All management occurs from the Security Management Server with the exception of manual updates. Each individual component (Security Gateway or Security Management Server) must be directly updated.

Once authenticated, authorized administrators have access to the following security functions:

- Ability to administer the TOE locally and remotely;
- Ability to configure a login banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- Ability to configure the maximum authentication failure limit;
- Ability to start and stop services;
- Ability to configure audit behavior, (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full);
- Ability to modify the behavior of the transmission of audit data to an external IT entity;
- Ability to modify the handling of audit data;
- Ability to modify the audit functionality when Local Audit Storage Space is full;
- Ability to manage the cryptographic keys;
- Ability to configure the cryptographic functionality;
- Ability to configure the IPsec functionality;
- Ability to configure the interaction between TOE components;
- Ability to re-enable an Administrator account;
- Ability to set the time which is used for time-stamps;
- Ability to configure the reference identifier for the peer;
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;
- Ability to import X.509v3 certificates to the TOE's trust store;
- Ability to configure the firewall rules;
- Ability to define and order packet filtering rules and associate those rules to network interfaces in support of the VPN functionality.

The Security management function satisfies the following security functional requirements:

- NDcPP22e:FMT_MOF.1/ManualUpdate: Only the authorized administrator can update the TOE.
- .NDcPP22e:FMT_MOF.1/Services: Only the authorized administrator can perform administrative functions.
- NDcPP22e:FMT_MTD.1/CoreData: Only the administrator can configure TSF-related functions.
- NDcPP22e:FMT_MTD.1/CryptoKeys: Cryptographic keys are managed through the interface provided by the Security Management Server which allows for the generation and import of keys used for certificates as well as import a Pre-Shared keys.
- VPNGW11:FMT_MTD.1/CryptoKeys: Only the authorized administrator can perform operations on cryptographic keys and certificates used for the VPN.

- NDcPP22e:FMT_SMF.1, VPNGW11:FMT_SMF.1, STFFW14e:FMT_SMF.1/FFW and VPNGW11:FMT_SMF.1/VPN: The TOE provides administrative interfaces to perform the functions identified above.
- NDcPP22e:FMT_SMR.2: The TOE supports administrator roles. The TOE is able to create roles for each configured administrator. An administrator can login to the TOE locally or remotely.

## 6.8 Packet Filtering

The packet filtering function is the VPN extended package is addressed entirely by the FFW_RUL_EXT.1 requirement. See Section 6.5.

The Packet Filtering function satisfies the following security functional requirements:

- VPNGW11:FPF_RUL_EXT.1: Please see Section 6.4 Stateful Traffic Filtering Firewall above for a description of the TOE's packet filtering capabilities.

## 6.9 Protection of the TSF

The TOE is a Security Gateway and a Security Management Server. All parts of the TOE are an appliance and are designed to not offer general purpose operating system interfaces to users. The TOE is designed to not provide access to locally stored passwords and also, while cryptographic keys can be entered, the TOE does not disclose any cryptographic keys stored in the TOE. All keys are stored within an internal configuration (effectively a database) and the TOE protects these values by not permitting any operator access to them.

The TOE components are hardware appliances that includes a real-time clock. The TOE uses the clock to support several security functions including timestamps for audit records, timing elements of cryptographic functions, and inactivity timeouts.

During power-up the integrity of all executables is verified with a digital signature. The public key used for signature verification comes pre-installed on the TOE. If an integrity test fails in the FIPS module, the system will enter a kernel panic and will fail to boot up. If an integrity test fails due to a non-matching hash, a log is written in addition.

During power-up algorithms are tested in the kernel and user-space and health tests are executed. If an algorithm test fails in the kernel, the system will enter a kernel panic and will fail to boot up. If an error occurs in user-space, cpstart will not load the modules and the system will remain protected by a low level default security policy that only allows outgoing connections from the gateway and does not allow IP forwarding. However, the TOE is still accessible via the local console login. All algorithms are tested in kernel and user-space apart from ECDSA and RSA as these are only used, and therefore tested, in user-space.

The TOE performed the following power-up cryptographic algorithm known answer tests:

| Algorithm | Implemented in | Description |
|---|---|---|
| AES encryption/decryption | User Crypto Library | Comparison of known answer to calculated value. |
| DRBG random bit generation | User Crypto Library | Comparison of known answer to calculated value. |
| ECDSA sign/verify | User Crypto Library | Comparison of known answer to calculated value. |
| HMAC-SHA | User Crypto Library | Comparison of known answer to calculated value. |
| RSA sign/verify | User Crypto Library | Comparison of known answer to calculated value. |
| SHA hashing | User Crypto Library | Comparison of known answer to calculated value. |
| AES encryption/decryption | Kernel Cryptography | Comparison of known answer to calculated value. |

| | | |
|---|---|---|
| HMAC-SHA | Kernel Cryptography | Comparison of known answer to calculated value. |
| DRBG random bit generation | Kernel Cryptography | Comparison of known answer to calculated value. |
| SHA hashing | Kernel Cryptography | Comparison of known answer to calculated value. |

**Table 6-3 Power-Up Cryptographic Known Answer Tests**

The TOE supports loading updates by the administrator using either management interface. The administrator obtains the update from the Check Point web site, and the TOE automatically verifies its digital signature. An unverified update cannot be installed.

The Protection of the TSF function satisfies the following security functional requirements:

- NDcPP22e:FPT_APW_EXT.1: The TOE does not offer any functions that will disclose to any user a plain text password. Furthermore, locally defined passwords are not stored in plaintext form.
- VPNGW11:FPT_FLS.1/SelfTest: The TOE contains self-tests that are executed during power-up. The TOE enters an error state when it fails self-tests.
- NDcPP22e:FPT_ITT.1: The TOE protects data between separate part of the TOE (Security Gateway to Security Management Server) using the IPsec protocol.
- NDcPP22e:FPT_SKP_EXT.1: The TOE does not provide any means for reading any key or CSP.
- NDcPP22e:FPT_STM_EXT.1: The TOE provides reliable time stamps using an internal clock, that can be manually configured by the administrator.
- NDcPP22e:FPT_TST_EXT.1 and VPNGW11:FPT_TST_EXT.1: The TOE runs self-tests during power-up to demonstrate correct operation. All security self-tests are performed on all TOE components. These tests are sufficient to demonstrate the TSF is performing correctly as they verify the cryptographic operations used through TOE processing and the integrity of the TSF itself.
- VPNGW11:FPT_TST_EXT.3: During power-up the integrity of all executables is verified with a digital signature.
- NDcPP22e:FPT_TUD_EXT.1 and VPNGW11:FPT_TUD_EXT.1: Each TOE component (Security Gateway and Security Management Server) offers an interface to query the current version of itself. Likewise, each TOE component must be updated individually. The TOE is designed to function properly as individual components are updated. The TOE's updates are digitally signed and verified using ECDSA with SHA-512 and P-521 curve.

## 6.10 TOE access

The TOE can be configured to display an administrator-configured message of the day banner that will be displayed before authentication is completed. The banner will be displayed when accessing the TOE via the console or web interfaces.

The TOE provides an inactivity timeout for both SmartConsole and serial CLI connection sessions. The authorized administrator can set the inactivity timeout and it can be different for each type of login (local/CLI and remote/SmartConsole). When an inactivity period is exceeded, the session is terminated. The user will be required to login in after any session has been terminated due to inactivity or after voluntary termination.

The TOE access function satisfies the following security functional requirements:

- NDcPP22e:FTA_SSL.3: The TOE allows remote inactive sessions to disconnect after a set period of time configurable in the GUI.

- NDcPP22e:FTA_SSL.4: The TOE allows session disconnect via a logout command. The logout command for Serial CLI is "exit" when performed at the top level of the CLI. SmartConsole has three methods of logout: select "exit" from the main menu, the keyboard short cut "Alt + F4", and using the red "x" in the top right corner.
- NDcPP22e:FTA_SSL_EXT.1: The TOE is able to terminate a local administrator session after a set inactivity time.
- NDcPP22e:FTA_TAB.1: The TOE supports a message of the day banner that is displayed when an administrator authenticates to the TOE both locally and remotely.

## 6.11 Trusted path/channels

The TOE uses IPsec to protect communications. The TOE employs IPsec when it sends audit data to an audit server, and when allowing remote administration connections. Authorized local administrators can only connect to the TOE directly via a serial connection. While the serial connection is not protected, it requires local, direct access to the individual device. In all remote cases, IPsec ensures traffic is not modified or disclosed. The nature of local connections ensures that interfacing with the CLI is secure.

The TOE can be either an initiator or responder in all IPsec communication. Cryptographic operations used in support of IPsec communication are described in Section 6.3.

The Trusted path/channels function satisfies the following security functional requirements:

- NDcPP22e:FTP_ITC.1: The TOE uses IPsec to provide a trusted communication channel between itself and an audit server and an authentication server.
- VPNGW11:FTP_ITC.1/VPN: The TOE uses IPsec to provide a trusted communication channel between itself and VPN peer endpoints.
- NDcPP22e:FTP_TRP.1/Admin: The TOE implements IPsec to provide a trusted communication path between itself and remote administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and detection of modification of the communicated data. The TOE uses IPsec to protect remote administration using the command line and GUI interfaces. While a TOE remote interface may utilize other protocols, those protocols must be protected by IPsec in the evaluated configuration.
- NDcPP22e:FTP_TRP.1/Join: The TOE requires any Security Gateway joining the Security Management Server to use a dedicated channel to register itself. This registration is done by selecting a passphrase for the Security Gateway during the first-time setup wizard. Once the Security Gateway object is established in SmartConsole, the administrator then enters this passphrase under the object in SmartConsole and the Secure Internal Communication (SIC) connection is established. This channel is dedicated to allow status information to be sent between the Security Management Server and Security Gateway. The ongoing communication between the components is then protected via an IPsec connection to ensure that all data between the components remains secure. Note that if the initial component joining fails, no actions from SmartConsole can be pushed to the Security Gateway that was disconnected or failed to be connected.

# 7. Hardware Platforms

Below is a list of hardware platforms included in the evaluation. All platforms are x86 based hardware. These platforms can be installed as a Security Gateway and all are running the R81.00 software. The list also includes a "Smart-1" appliance functioning as a Security Management Server, running the same R81.00 software. Finally, the list includes an ESXi appliance, upon which one can install the R81.00 software either configured as a Security Gateway or as a Security Management Server.

| Appliance | CPU | CPU Family |
|---|---|---|
| 3600 | Denverton C3558 | Intel Atom® Processor C Series |
| 3800 | Denverton C3758 | Intel Atom® Processor C Series |
| 6200 | Coffee Lake G5400 | Intel® Pentium® Gold Processor Series |
| 154**, 156** | Haswell Single/Dual Xeon E5-2630v3 | Intel® Xeon® Processor E5 v3 Family |
| 6400 | Coffee Lake i3-8100 | Intel® 8th Generation Core™ i3 |
| 6600 | Coffee Lake i5-8500 | Intel® 8th Generation Core™ i5 |
| 6700 | Coffee Lake E-2176G | Intel® Xeon® E Processor |
| 6900 | Coffee Lake i9-9900 KF | Intel® 9th Generation Core™ i9 |
| 7000 | Cascade Lake 4216 | Intel® 2nd Generation Xeon® Scalable |
| 16000 | Cascade Lake Dual Xeon 2x 4208 | Intel® 2nd Generation Xeon® Scalable |
| 16200 | Cascade Lake Dual Xeon 2x 4214 | Intel® 2nd Generation Xeon® Scalable |
| 16600 | Cascade Lake Refresh Dual XEON 2x 4214R | Intel® 2nd Generation Xeon® Scalable |
| 26000 | Cascade Lake Dual Xeon 2x 5220 | Intel® 2nd Generation Xeon® Scalable |
| 28000 | Cascade Lake Dual Xeon 2x 6254 | Intel® 2nd Generation Xeon® Scalable |
| 28600 | Cascade Lake Dual Xeon 2x 6254 | Intel® 2nd Generation Xeon® Scalable |
| Smart-1 525 | Intel Sandy Bridge Xeon E3-1270 | Intel® Xeon® E3 Family |
| ESXi (HPE D360 G10) | Cascade Lake Xeon 4214 | Intel® 2nd Generation Xeon® Scalable |

**Table 7-1 Appliance CPU & CPU Family**

The following are the Ethernet controller that are used in each evaluated Appliance model.

| Appliance | Ethernet Controller |
|---|---|
| 3600<br>3800 | eth5 & Mgmt: Intel Corporation I211 Gigabit Network Connection<br>O\B: Intel Corporation Ethernet Connection X553 1GbE |
| 6200<br>154**, 156**<br>6400 | Intel Corporation I211 Gigabit Network Connection |
| 16000<br>16200<br>26000<br>28000 | Intel Corporation I350 Gigabit Network Connection |
| 6600<br>6700<br>6900<br>7000 | Mgmt & Sync: Intel Corporation I211 Gigabit Network Connection<br>O\B: Intel Corporation I350 Gigabit Network Connection |
| 16600<br>28600 | Mgmt & Sync: Intel Corporation I350 Gigabit Network Connection<br>O\B: Mellanox Technologies MT27800 Family [ConnectX-5] |
| Smart-1 525 | Ethernet controller: Intel Corporation I350 Gigabit Network Connection |
| ESXi (HPE D360 G10) | Intel Ethernet 1Gb 4-port 366FLR Adapter |

**Table 7-2 Ethernet Controllers**

# 8. Requirement Allocation

This section provides a mapping of the distributed TOE components to the SFRs in this ST. This TOE is a distributed TOE consistent with Use Case 3 as defined in the NDcPP22e. The following table presents the required mapping.

| Requirement | Distributed TOE SFR Allocation | Distributed TOE Audit Event Allocation |
|---|---|---|
| **NDcPP22e:FAU_GEN.1** | All | All |
| **STFFW14e:FAU_GEN.1** | Security Gateway | Security Gateway |
| **VPNGW11:FAU_GEN.1** | Security Gateway | Security Gateway |
| **NDcPP22e:FAU_GEN.2** | All | None |
| **NDcPP22e:FAU_GEN_EXT.1** | All | None |
| **NDcPP22e:FAU_STG_EXT.1** | All | None |
| **NDcPP22e:FAU_STG_EXT.4** | All | None |
| **NDcPP22e:FAU_STG_EXT.5** | All | None |
| **NDcPP22e:FCO_CPC_EXT.1** | All | All |
| **NDcPP22e:FCS_CKM.1** | All | None |
| **VPNGW11:FCS_CKM.1/IKE** | All | None |
| **NDcPP22e:FCS_CKM.2** | All | None |
| **NDcPP22e:FCS_CKM.4** | All | None |
| **NDcPP22e:FCS_COP.1/DataEncryption** | All | None |
| **VPNGW11:FCS_COP.1/DataEncryption** | All | None |
| **NDcPP22e:FCS_COP.1/Hash** | All | None |
| **NDcPP22e:FCS_COP.1/KeyedHash** | All | None |
| **NDcPP22e:FCS_COP.1/SigGen** | All | None |
| **NDcPP22e:FCS_IPSEC_EXT.1** | Security Gateway | Security Gateway |
| **VPNGW11:FCS_IPSEC_EXT.1** | Security Gateway | Security Gateway |
| **NDcPP22e:FCS_RBG_EXT.1** | All | None |
| **STFFW14e:FDP_RIP.2** | All | None |
| **STFFW14e:FFW_RUL_EXT.1** | Security Gateway | Security Gateway |
| **NDcPP22e:FIA_AFL.1** | Security Management Server | Security Management Server |
| **NDcPP22e:FIA_PMG_EXT.1** | Security Management Server | Security Management Server |
| **VPNGW11:FIA_PSK_EXT.1** | Security Gateway | Security Gateway |
| **NDcPP22e:FIA_UAU.7** | Security Management Server | None |
| **NDcPP22e:FIA_UAU_EXT.2** | Security Management Server | Security Management Server |
| **NDcPP22e:FIA_UIA_EXT.1** | Security Management Server | Security Management Server |
| **NDcPP22e:FIA_X509_EXT.1/ITT** | Security Gateway | Security Gateway |
| **NDcPP22e:FIA_X509_EXT.1/Rev** | All | All |
| **VPNGW11:FIA_X509_EXT.1/Rev** | Security Gateway | Security Gateway |
| **NDcPP22e:FIA_X509_EXT.2** | All | None |
| **VPNGW11:FIA_X509_EXT.2** | Security Gateway | Security Gateway |
| **NDcPP22e:FIA_X509_EXT.3** | All | None |
| **VPNGW11:FIA_X509_EXT.3** | Security Gateway | Security Gateway |
| **NDcPP22e:FMT_MOF.1/ManualUpdate** | Security Management Server | Security Management Server |
| **NDcPP22e:FMT_MOF.1/Services** | Security Management Server | Security Management Server |
| **NDcPP22e:FMT_MTD.1/CoreData** | Security Management Server | Security Management Server |
| **NDcPP22e:FMT_MTD.1/CryptoKeys** | Security Management Server | Security Management Server |
| **VPNGW11:FMT_MTD.1/CryptoKeys** | Security Management Server | Security Management Server |
| **NDcPP22e:FMT_SMF.1** | Security Management Server | Security Management Server |
| **VPNGW11:FMT_SMF.1** | Security Management Server | Security Management Server |
| **STFFW14e:FMT_SMF.1/FFW** | Security Management Server | Security Management Server |

| | | |
|---|---|---|
| **VPNGW11:FMT_SMF.1/VPN** | Security Management Server | Security Management Server |
| **NDcPP22e:FMT_SMR.2** | Security Management Server | Security Management Server |
| **VPNGW11:FPF_RUL_EXT.1** | Security Gateway | Security Gateway |
| **NDcPP22e:FPT_APW_EXT.1** | Security Management Server | None |
| **VPNGW11:FPT_FLS.1/SelfTest** | All | None |
| **NDcPP22e:FPT_ITT.1** | All | All |
| **NDcPP22e:FPT_SKP_EXT.1** | All | None |
| **NDcPP22e:FPT_STM_EXT.1** | All | All |
| **NDcPP22e:FPT_TST_EXT.1** | All | All |
| **VPNGW11:FPT_TST_EXT.1** | Security Gateway | Security Gateway |
| **VPNGW11:FPT_TST_EXT.3** | Security Gateway | Security Gateway |
| **NDcPP22e:FPT_TUD_EXT.1** | All | All |
| **VPNGW11:FPT_TUD_EXT.1** | Security Gateway | Security Gateway |
| **NDcPP22e:FTA_SSL.3** | Security Management Server | Security Management Server |
| **NDcPP22e:FTA_SSL.4** | Security Management Server | Security Management Server |
| **NDcPP22e:FTA_SSL_EXT.1** | Security Management Server | Security Management Server |
| **NDcPP22e:FTA_TAB.1** | Security Management Server | None |
| **NDcPP22e:FTP_ITC.1** | Security Gateway | Security Gateway |
| **VPNGW11:FTP_ITC.1/VPN** | Security Gateway | Security Gateway |
| **NDcPP22e:FTP_TRP.1/Admin** | Security Management Server | Security Management Server |
| **NDcPP22e:FTP_TRP.1/Join** | All | Security Management Server |

**Table 8-1 Requirement and Audit Allocation**