Check Point Software Technologies Ltd. Security Gateway Appliances R77.30 (NDPP11e3/VPN/FW) Security Target

Version 0.91
12/29/15

Prepared for:

Check Point Software Technologies Ltd.

5 Ha'Solelim Street, Tel Aviv 67897, 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 Appliances R77.30 provided by Check Point Software Technologies Ltd.. The TOE is being evaluated as a network infrastructure device.

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 letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
    - 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 …").
- The NDPP uses an additional convention – the 'case' – which defines parts of an SFR that apply only when corresponding selections are made or some other identified conditions exist. Only the applicable cases are identified in this ST and they are identified using **bold** text.
- 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 Appliances R77.30 (NDPP11e3/VPN/FW) Security Target

ST Version – Version 0.91

ST Date – 12/29/15

1.2 TOE Reference

TOE Identification – Check Point Software Technologies Ltd. Security Gateway Appliances R77.30 (see Section 7 for specific hardware information)

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 Appliances R77.30. The product is a VPN Gateway and packet filtering firewall appliance. 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 Security 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). This description effectively partitions product functionality into two classes:

- Claimed security functionality that is evaluated in the context of this ST;
- Other functionality that is in the TOE but is not evaluated in the context of this ST except for the determination that it cannot compromise any claimed security functionality;

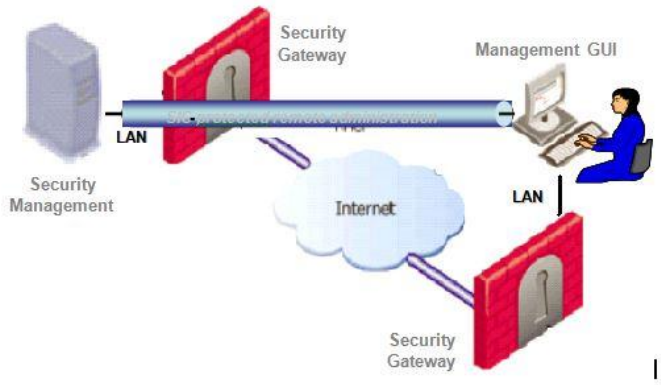
The TOE Description consists of the following subsections:

- TOE Architecture – describes the high level TOE components and their relationship to each other.
- Physical Boundaries - describes the hardware, firmware, and software parts that constitute the TOE
- Logical Boundaries - – describes the claimed logical security features offered by the TOE
- TOE Guidance – identifies the guidance documentation that is considered to be part of the TOE.

1.4.1 TOE Architecture

The TOE is a network device with firewall capabilities for filtering traffic based on packet rules. It is a distributed system with support for a security management server, allowing remote administration over a protected IPsec connection. The TOE includes the following components:

- Check Point Security Gateway Appliances, including Security Gateway software, Gaia operating system, and appliance hardware; and
- Security Management Servers, including Security Management software, Security Gateway software, and hardware platform; and
- SmartConsole Management GUI software



Check Point Security Gateway R77.30 Security Gateway software is installed on a hardware platform in combination with an operating system (OS), in accordance with TOE guidance, in a FIPS 140-2 mode. The OS supports the TOE by providing 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 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. The TOE can be optionally configured to perform user authentication with the support of external authentication servers in the IT environment.

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.

Security Management is performed using the SmartConsole Management GUI software. The Security Management software, OS and hardware platform are collectively identified in this ST as the 'Security Management server'.

One or more Security Gateway appliances are managed by a Security Management server installation that maintains security policy information for the gateways, and collects audit records from the gateways for review by TOE administrators. The audit records may also be sent to an external audit server.

The evaluated configuration supports both local and remote administration. Local administration is via a directly connected console. Remote administration is via an IPsec protected connection between the Security Management Server and the Gateway Appliance or via a remote CLI protected via an IPsec connection.

1.4.1.1 Physical Boundaries

There are three different hardware platforms for the Check Point Security Gateway Appliances and Security Management Appliances including Check Point IAS appliances integrated with HP and Fujitsu. All platforms use

the same image. The difference is mainly in hardware makeup and physical ports. All platforms are x86 based hardware.

The SmartConsole Management GUI software is installed on a Windows workstation (Windows 8, Windows 7). Authorized administrators use the GUI software or CLI to remotely manage the TOE.

The TOE may be configured to interact with external servers:

- External Certificate Authority (CA).
- External certificate validation server (HTTP or LDAP CRLDP, OCSP).
- External NTP time-synchronization server
- External audit server (OPSEC)

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by R77.30:

- Security audit
- 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 Gateway Appliances can be configured to store logs locally, forward logs to the Security Management Server, or both. If configured to send logs to the Security Management Server, in the event of a loss of network connectivity to the Security Management Server, then the Gateway Appliance will store locally until the connection is restored. The TOE can be configured to send audit logs to a syslog server as well. The connection between the TOE and remote server is protected with IPsec. Finally, note that the Gateway Appliances can be configured such that if they run out of disk space for local logs, they can block all connections

1.4.1.2.2 Cryptographic support

The TOE uses a Check Point cryptographic module 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.

1.4.1.2.3 User data protection

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

1.4.1.2.4 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 Security Gateway Appliances 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.

1.4.1.2.5 Identification and authentication

The TOE implements a password based authentication mechanism that identifies operators via usernames. Passwords are stored obfuscated, and passwords for local login are stored Unix hashed. The TOE supports passwords with lengths 15 or greater characters and all special characters as required by FIA_PMG_EXT.1.1.

1.4.1.2.6 Security management

The TOE allows both local and remote administration for management of the TOE's security functions. The TOE creates and maintains profiles for configured administrators. An administrator can log in locally to the TOE using a serial connection. The administrator is greeted with a console environment, where configuration is mainly done through command-line syntax. The local login operates in a Unix shell. There are two remote administration interfaces. The first remote administration interface is executed through a Graphical User Interface using TLS over IPsec. Though the connections from a browser to the TOE are TLS connections, the TOE requires an IPSec connection to wrap the TLS connection. The second remote administration interface is a command line interface (CLI) using SSH over IPsec.

1.4.1.2.7 Packet Filtering

Please see Section 1.4.1.2.4 Stateful Traffic Filtering Firewall for a description of the TOE's packet filtering mechanism.

1.4.1.2.8 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, and UNIX shell login passwords are stored as a UNIX hash. 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 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 is able to terminate interactive sessions if the session is inactive for a set period of time. The time can be configured via the TOE configuration. Also, the TOE can lock a user out based on the number of failed logins. This can also be configured via the TOE configuration.

1.4.1.2.9 TOE access

Access to the TOE is mainly through a Security Management Server. The connection between the Security Management Server and the TOE is secured via IPsec. The second remote administration interface is CLI and is also protected with IPsec. The TOE also provides a local login console, which is a Unix shell environment.

1.4.1.2.10 Trusted path/channels

The TOE protects all communications with outside entities using IPsec communications only. This is mainly to fulfill a Commercial Solutions for Classified (CSfC) requirement for communications. Any other protocol (such as SSH or TLS) is wrapped in an IPsec tunnel.

1.4.2 TOE Documentation

- Check Point Software Technologies LTD. Security Gateway Appliances R77.30 Common Criteria Supplement, Version 0.2, December 29, 2015
- Check Point Software Technologies LTD. R77.30 Installation Guide, Version1.0, December 9, 2015

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 4, September 2012.
    - Part 2 Extended and CSfC Selections for VPN Gateways
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.
    - Part 3 Conformant
- Protection Profile for Network Devices, Version 1.1 (with Errata #3), 8 June 2012 (NDPP11e3) with the following two extended packages:
    - Network Device Protection Profile Extended Package Stateful Traffic Filter Firewall, Version 1.0, 19 December 2011 (FW)
    - Network Device Protection Profile Extended Package VPN Gateway, Version 1.1, 15 April 2013(VPN) as amended by CSfC Selections for VPN Gateways

2.1 Conformance Rationale

The ST conforms to the NDPP11e3/VPN/FW. The security problem definition, security objectives, and security requirements have been drawn from the PPs and EPs.

3. Security Objectives

The Security Problem Definition may be found in the NDPP11e3/VPN/FW and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The NDPP11e3/VPN/FW offers additional information about the identified security objectives, but that has not been reproduced here and the NDPP11e3/VPN/FW should be consulted if there is interest in that material.

In general, the NDPP11e3/VPN/FW has defined Security Objectives appropriate for network infrastructure device and as such are applicable to the Security Gateway Appliances R77.30 TOE.

3.1 Security Objectives for the Operational Environment

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.

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

OE.TRUSTED_ADMIN TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

OE.CONNECTIONS TOE administrators will ensure that the TOE is installed in a manner that will allow the TOE to effectively enforce its policies on network traffic flowing among attached networks.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the NDPP11e3/VPN/FW. The NDPP11e3/VPN/FW defines the following extended requirements and since they are not redefined in this ST the NDPP11e3/VPN/FW should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- FAU_STG_EXT.1: External Audit Trail Storage
- FCS_CKM_EXT.4: Cryptographic Key Zeroization
- FCS_IPSEC_EXT.1: Explicit: IPSEC
- FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
- FFW_RUL_EXT.1: Stateful Traffic Filtering
- FIA_PMG_EXT.1: Password Management
- FIA_PSK_EXT.1: Extended: Pre-Shared Key Composition
- FIA_UAU_EXT.2: Extended: Password-based Authentication Mechanism
- FIA_UIA_EXT.1: User Identification and Authentication
- FIA_X509_EXT.1: Extended: X.509 Certificates
- FPF_RUL_EXT.1: Packet Filtering
- FPT_APW_EXT.1: Extended: Protection of Administrator Passwords
- FPT_SKP_EXT.1: Extended: Protection of TSF Data (for reading of all symmetric keys)
- FPT_TST_EXT.1: TSF Testing
- FPT_TUD_EXT.1: Extended: Trusted Update
- 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 NDPP11e3/VPN/FW. The refinements and operations already performed in the NDPP11e3/VPN/FW are not identified (e.g., highlighted) here, rather the requirements have been copied from the NDPP11e3 and any residual operations have been completed herein. Of particular note, the NDPP11e3/VPN/FW 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.

The SARs are also drawn from the NDPP11e3/VPN/FW which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the NDPP11e3/VPN/FW that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The NDPP11e3/VPN/FW should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by Security Gateway Appliances R77 TOE.

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit Data Generation
	FAU_GEN.2: User Identity Association
	FAU_STG_EXT.1: External Audit Trail Storage
FCS: Cryptographic support	FCS_CKM.1: Cryptographic Key Generation (for asymmetric keys)
	FCS_CKM_EXT.4: Cryptographic Key Zeroization
	FCS_COP.1(1): Cryptographic Operation (for data encryption/decryption)
	FCS_COP.1(2): Cryptographic Operation (for cryptographic signature)
	FCS_COP.1(3): Cryptographic Operation (for cryptographic hashing)
	FCS_COP.1(4): Cryptographic Operation (for keyed-hash message authentication)
	FCS_IPSEC_EXT.1: Explicit: IPSEC
	FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
FDP: User data protection	FDP_RIP.2: Full Residual Information Protection
FFW: Stateful Traffic Filtering Firewall	FFW_RUL_EXT.1: Stateful Traffic Filtering
FIA: Identification and authentication	FIA_AFL.1: Authentication Failure Handling
	FIA_PMG_EXT.1: Password Management
	FIA_PSK_EXT.1: Extended: Pre-Shared Key Composition
	FIA_UAU.7: Protected Authentication Feedback
	FIA_UAU_EXT.2: Extended: Password-based Authentication Mechanism
	FIA_UIA_EXT.1: User Identification and Authentication
	FIA_X509_EXT.1: Extended: X.509 Certificates
FMT: Security management	FMT_MOF.1: Management of Security Functions Behavior
	FMT_MTD.1: Management of TSF Data (for general TSF data)

	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.2: Restrictions on Security Roles
FPF: Packet Filtering	FPF_RUL_EXT.1: Packet Filtering
FPT: Protection of the TSF	FPT_APW_EXT.1: Extended: Protection of Administrator Passwords
	FPT_FLS.1: Fail Secure
	FPT_ITT.1: Basic Internal TSF Data Transfer Protection
	FPT_SKP_EXT.1: Extended: Protection of TSF Data (for reading of all symmetric keys)
	FPT_STM.1: Reliable Time Stamps
	FPT_TST_EXT.1: TSF Testing
	FPT_TUD_EXT.1: Extended: Trusted Update
FTA: TOE access	FTA_SSL.3: TSF-initiated Termination
	FTA_SSL.4: User-initiated Termination
	FTA_SSL_EXT.1: TSF-initiated Session Locking
	FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	FTP_ITC.1: Inter-TSF trusted channel
	FTP_TRP.1: Trusted Path

Table 1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Audit Data Generation (FAU_GEN.1)

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;
d) Specifically defined auditable events listed in **Table 2 Auditable Events**.

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 PP/ST, information specified in column three of **Table 2 Auditable Events**.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	
FAU_GEN.2	None.	
FAU_STG_EXT.1	None.	
FCS_CKM.1	None.	
FCS_CKM_EXT.4	None.	
FCS_COP.1(1)	None.	
FCS_COP.1(2)	None.	
FCS_COP.1(3)	None.	

Requirement	Auditable Events	Additional Audit Record Contents
FCS_COP.1(4)	None.	
FCS_RBG_EXT.1	None.	
FCS_IPSEC_EXT.1	Session Establishment with peer.	Entire packet contents of packets transmitted/received during session establishment.
FDP_RIP.2	None.	
FFW_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports TOE Interface
	Indication of packets dropped due to too much network traffic	TOE interface that is unable to process packets
FIA_AFL_.1	None.	
FIA_PMG_EXT.1	None.	
FIA_PSK_EXT.1	None	
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	
FIA_X509_EXT.1	Establishing session with CA	Entire packet contents of packets transmitted/received during session establishment
FMT_MOF.1	None.	
FMT_MTD.1	None.	
FMT_SMF.1	None.	
FMT_SMR.2	None.	
FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE Interface
	Indication of packets dropped due to too much network traffic	TOE interface that is unable to process packets
FPT_SKP_EXT.1	None.	
FPT_APW_EXT.1	None.	
FPT_FLS.1	None.	
FPT_ITT.1	None.	
FPT_STM.1	Changes to the time.	The old and new values for the time. Origin of the attempt (e.g., IP address).
FPT_TUD_EXT.1	Initiation of update.	No additional information.
FPT_TST_EXT.1	None.	
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	No additional information.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	No additional information.
FTA_SSL.4	The termination of an interactive session.	No additional information.
FTA_TAB.1	None.	
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.
FTP_TRP.1	Initiation of the trusted channel.	Identification of the claimed user

Requirement	Auditable Events	Additional Audit Record Contents
	Termination of the trusted channel. Failures of the trusted path functions.	identity.

Table 2 Auditable Events

5.1.1.2 User Identity Association (FAU_GEN.2)

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.

5.1.1.3 External Audit Trail Storage (FAU_STG_EXT.1)

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 implementing the [***IPsec***] protocol.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1)

FCS_CKM.1.1

Refinement: The TSF shall generate asymmetric cryptographic keys used for key establishment in accordance with ***NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' for elliptic curve-based key establishment schemes and implementing 'NIST curves' P-256, P-384 and P-521 (as defined in FIPS PUB 186-3, 'Digital Signature Standard'; NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes)*** and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

FCS_CKM.1.2

Refinement: The TSF shall generate asymmetric cryptographic keys used for IKE peer authentication in accordance with a: FIPS PUB 186-3, "Digital Signature Standard (DSS)",***Appendix B.4 for ECDSA schemes and implementing NIST curves P-256, P-384 and P-521,*** and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

5.1.2.2 Cryptographic Key Zeroization (FCS_CKM_EXT.4)

FCS_CKM_EXT.4.1

The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

5.1.2.3 Cryptographic Operation (for data encryption/decryption) (FCS_COP.1(1))

FCS_COP.1(1).1

Refinement: The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm AES operating in [***CBC, GCM***] and cryptographic key sizes 128-bits and 256-bits that meets the following:

- FIPS PUB 197, 'Advanced Encryption Standard (AES)'
- [***NIST SP 800-38A, NIST SP 800-38D***].

5.1.2.4 Cryptographic Operation (for cryptographic signature) (FCS_COP.1(2))

FCS_COP.1(2).1

Refinement: The TSF shall perform cryptographic signature services in accordance with a: [***Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater***] that meets FIPS PUB 186-3, 'Digital Signature Standard' with 'NIST curves' P-256, P-384 and ***[P-521]*** (as defined in FIPS PUB 186-3, 'Digital Signature Standard')].

5.1.2.5 Cryptographic Operation (for cryptographic hashing) (FCS_COP.1(3))

FCS_COP.1(3).1

Refinement: The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [***SHA-256, SHA-384, SHA-512***] and message digest sizes [***256, 384,512***] bits that meet the following: FIPS Pub 180-3, 'Secure Hash Standard.'

5.1.2.6 Cryptographic Operation (for keyed-hash message authentication) (FCS_COP.1(4))

FCS_COP.1(4).1

Refinement: The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-[***SHA-256***], key size [**256 key size (in bits)**], and message digest sizes [***256***] bits that meet the following: FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code', and FIPS Pub 180-3, 'Secure Hash Standard.'

5.1.2.7 Extended: Internet Protocol Security (IPsec) Communications (FCS_IPSEC_EXT.1)

.

FCS_IPSEC_EXT.1.1

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

FCS_IPSEC_EXT.1.2

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

FCS_IPSEC_EXT.1.3

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

FCS_IPSEC_EXT.1.4

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

FCS_IPSEC_EXT.1.5

The TSF shall implement the protocol: ***IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23***) and [***no other RFCs for hash functions***]

FCS_IPSEC_EXT.1.6

The TSF shall ensure the encrypted payload in the [***IKEv2***] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 6379 and [***no other algorithm***].

FCS_IPSEC_EXT.1.7

The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.

FCS_IPSEC_EXT.1.8

The TSF shall ensure that [***IKEv2 SA lifetimes can be configured by an Administrator based on number of packets or length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs***].

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.

FCS_IPSEC_EXT.1.10
The TSF shall generate nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^ [***256***].

FCS_IPSEC_EXT.1.11
The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), 19 (256-bit Random ECP), and [***20 (384-bit Random ECP***].

FCS_IPSEC_EXT.1.12
The TSF shall ensure that all IKE protocols perform peer authentication using a [***ECDSA***] that use X.509v3 certificates that conform to RFC 4945 and [***Pre-shared Keys***].

FCS_IPSEC_EXT.1.13
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 ***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 [***IKEv2 CHILD_SA***] connection.

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

FCS_RBG_EXT.1.1
The TSF shall perform all random bit generation (RBG) services in accordance with [***NIST Special Publication 800-90 using [Hash_DRBG (SHA-256)]***] seeded by an entropy source that accumulated entropy from [***[1] TSF software-based noise source***].

FCS_RBG_EXT.1.2
The deterministic RBG shall be seeded with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

5.1.3 User data protection (FDP)

5.1.3.1 Full Residual Information Protection (FDP_RIP.2)

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.

5.1.4 Stateful Traffic Filtering Firewall (FFW)

5.1.4.1 Stateful Traffic Filtering (FFW_RUL_EXT.1)

FFW_RUL_EXT.1.1
The TSF shall perform Stateful Traffic Filtering on network packets processed by the TOE.

FFW_RUL_EXT.1.2
The TSF shall process the following network traffic protocols:
- Internet Control Message Protocol version 4 (ICMPv4)
- Internet Control Message Protocol version 6 (ICMPv6)
- Internet Protocol (IPv4)
- Internet Protocol version 6 (IPv6)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

and be capable of inspecting network packet header fields defined by the following RFCs to the extent mandated in the other elements of this SFR
- RFC 792 (ICMPv4)
- RFC 4443 (ICMPv6)
- RFC 791 (IPv4)
- RFC 2460 (IPv6)

- RFC 793 (TCP)
- RFC 768 (UDP).

FFW_RUL_EXT.1.3

The TSF shall allow the definition of Stateful Traffic Filtering rules using the following network protocol fields:
- ICMPv4
o Type
o Code
- ICMPv6
o Type
o Code
- IPv4
o Source address
o Destination Address
o Transport Layer Protocol
- IPv6
o Source address
o Destination Address
o Transport Layer Protocol
- TCP
o Source Port
o Destination Port
- UDP
o Source Port
o Destination Port
and distinct interface.

FFW_RUL_EXT.1.4

The TSF shall allow the following operations to be associated with Stateful Traffic Filtering rules: permit, deny, and log.

FFW_RUL_EXT.1.5

The TSF shall allow the Stateful Traffic Filtering rules to be assigned to each distinct network interface.

FFW_RUL_EXT.1.6

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***].

FFW_RUL_EXT.1.7

The TSF shall be able to process the following network protocols:
1. FTP,
2. [***none***],
to dynamically define rules or establish sessions allowing network traffic of the following types:
- FTP: TCP data sessions in accordance with the FTP protocol as specified in RFC 959,
- [**none**].

FFW_RUL_EXT.1.8

The TSF shall enforce the following default Stateful Traffic Filtering rules on all network traffic:
1. The TSF shall reject and be capable of logging packets which are invalid fragments;
2. The TSF shall reject and be capable of logging fragmented IP packets which cannot be re-assembled completely;
3. The TSF shall reject 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;
4. The TSF shall reject 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;
5. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a broadcast network;
6. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a multicast network;
7. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
8. The TSF shall reject and be capable of logging network packets where the source address of the network packet is a multicast;
9. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is a link-local address;
10. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as being an address 'reserved for future use' as specified in RFC 5735 for IPv4;
11. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as an 'unspecified address' or an address 'reserved for future definition and use' as specified in RFC 3513 for IPv6;
12. The TSF shall reject and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and
13. [**no other rules**].

FFW_RUL_EXT.1.9

When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall process the applicable Stateful Traffic Filtering rules (as determined in accordance with FFW_RUL_EXT.1.5) in the following order: administrator-defined.

FFW_RUL_EXT.1.10

When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall deny packet flow if a matching rule is not identified.

5.1.5 Identification and authentication (FIA)

5.1.5.1 Authentication Failure Handling (FIA_AFL.1)

FIA_AFL.1.1

Refinement: The TSF shall detect when an Administrator configurable positive integer of successive unsuccessful authentication attempts occur related to administrators attempting to authenticate remotely.

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been met, the TSF shall [***prevent the offending remote administrator from successfully authenticating until an Administrator defined time period has elapsed***].

5.1.5.2 Password Management (FIA_PMG_EXT.1)

FIA_PMG_EXT.1.1

The TSF shall provide the following password management capabilities for administrative passwords:
1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and special characters: [***!, @, #, $, %, ^, &, *, (,)***];
2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater.

5.1.5.3 Extended: Pre-Shared Key Composition (FIA_PSK_EXT.1)

FIA_PSK_EXT.1.1
The TSF shall be able to use pre-shared keys for IPsec and [***no other protocols]***].

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 ")").

FIA_PSK_EXT.1.3
The TSF shall condition the text-based pre-shared keys by using [***no conditioning]***].

FIA_PSK_EXT.1.4
The TSF shall be able to [***accept***] bit-based pre-shared keys

5.1.5.4 Protected Authentication Feedback (FIA_UAU.7)

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.

5.1.5.5 Extended: Password-based Authentication Mechanism (FIA_UAU_EXT.2)

FIA_UAU_EXT.2.1
The TSF shall provide a local password-based authentication mechanism, [***none***] to perform administrative user authentication.

5.1.5.6 User Identification and Authentication (FIA_UIA_EXT.1)

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***].

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.

5.1.5.7 Extended: X.509 Certificates (FIA_X509_EXT.1)

FIA_X509_EXT.1.1
The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec and [***no other protocols***] connections.

FIA_X509_EXT.1.2
The TSF shall store and protect certificate(s) from unauthorized deletion and modification.

FIA_X509_EXT.1.3
The TSF shall provide the capability for authenticated Administrators to load X.509v3 certificates into the TOE for use by the security functions specified in this PP.

FIA_X509_EXT.1.4
The TSF shall generate a Certificate Request Message as specified in RFC 2986 and be able to provide the following information in the request: public key, Common Name, Organization, Organizational Unit, and Country.

FIA_X509_EXT.1.5
The TSF shall validate the certificate using [***a Certificate Revocation List (CRL) as specified in RFC 5759***].

FIA_X509_EXT.1.6
The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension is present and the cA flag is set to TRUE for all CA certificates.

FIA_X509_EXT.1.7
The TSF shall not treat a certificate as a CA certificate if the basicConstraints extension is not present or the cA flag is not set to TRUE.

FIA_X509_EXT.1.8
The TSF shall not establish an SA if a certificate or certificate path is deemed invalid.

FIA_X509_EXT.1.9
The TSF shall support peer identifiers of the following types: [***IP address, Distinguished Name (DN)***] and [***no other reference identifier type***].

FIA_X509_EXT.1.9A
The TSF shall not establish an SA if the presented identifier does not match the configured reference identifier of the peer.

FIA_X509_EXT.1.10
When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall,[***not accept the certificate***].

5.1.6 Security management (FMT)

5.1.6.1 Management of Security Functions Behavior (FMT_MOF.1)

FMT_MOF.1.1
Refinement: The TSF shall restrict the ability to enable, disable, determine and modify the behavior of all of the security functions of the TOE identified in this **VPN** EP to an authenticated Administrator.

5.1.6.2 Management of TSF Data (for general TSF data) (FMT_MTD.1)

FMT_MTD.1.1
The TSF shall restrict the ability to manage the TSF data to the Security Administrators.

5.1.6.3 Specification of Management Functions (FMT_SMF.1)

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 update the TOE, and to verify the updates using [***digital signature***] capability prior to installing those updates;
- [***Ability to configure the cryptographic functionality***
- ***Ability to configure the IPsec functionality***
- ***Ability to enable, disable, determine and modify the behavior of all the security functions of the TOE identified in this VPN EP to the Administrator***
- ***Ability to configure all security management functions identified in other sections of this* VPN *EP***].

5.1.6.4 Restrictions on Security Roles (FMT_SMR.2)

FMT_SMR.2.1
The TSF shall maintain the roles: Authorized Administrator.

FMT_SMR.2.2
The TSF shall be able to associate users with roles.

FMT_SMR.2.3
The TSF shall ensure that the conditions
- Authorized Administrator role shall be able to administer the TOE locally;

- Authorized Administrator role shall be able to administer the TOE remotely;
are satisfied

5.1.7 Packet Filtering (FPF)

5.1.7.1 Packet Filtering (FPF_RUL_EXT.1)

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

FPF_RUL_EXT.1.2
The TSF shall process the following network traffic protocols:
- Internet Protocol (IPv4)
- Internet Protocol version 6 (IPv6)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
and be capable of inspecting network packet header fields defined by the following RFCs to the extent mandated in the other elements of this SFR
- RFC 791 (IPv4)
- RFC 2460 (IPv6)
- RFC 793 (TCP)
- RFC 768 (UDP).

FPF_RUL_EXT.1.3
The TSF shall allow the definition of Packet Filtering rules using the following network protocol fields:
- IPv4
o Source address
o Destination Address
o Protocol
- IPv6
o Source address
o Destination Address
o Next Header (Protocol)
- TCP
o Source Port
o Destination Port
- UDP
o Source Port
o Destination Port
and distinct interface.

FPF_RUL_EXT.1.4
The TSF shall allow the following operations to be associated with Packet Traffic Filtering rules: permit, deny, and log.

FPF_RUL_EXT.1.5
The TSF shall allow the Packet Traffic Filtering rules to be assigned to each distinct network interface.

FPF_RUL_EXT.1.6
The TSF shall process the applicable Packet Filtering rules (as determined in accordance with FPF_RUL_EXT.1.5) in the following order: Administrator-defined.

FPF_RUL_EXT.1.7
The TSF shall deny packet flow if a matching rule is not identified.

5.1.8 Protection of the TSF (FPT)

5.1.8.1 Extended: Protection of Administrator Passwords (FPT_APW_EXT.1)

FPT_APW_EXT.1.1
The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2
The TSF shall prevent the reading of plaintext passwords.

5.1.8.2 Fail Secure (FPT_FLS.1)

FPT_FLS.1.1
Refinement: The TSF shall shutdown 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.

5.1.8.3 Basic Internal TSF Data Transfer Protection (FPT_ITT.1)

FPT_ITT.1.1
Refinement: 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 [***IPsec***].

5.1.8.4 Extended: Protection of TSF Data (for reading of all symmetric keys) (FPT_SKP_EXT.1)

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

5.1.8.5 Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1
The TSF shall be able to provide reliable time stamps for its own use.

5.1.8.6 TSF Testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1
The TSF shall run a suite of self tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

FPT_TST_EXT.1.2
The TSF shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the TSF-provided cryptographic service specified in FCS_COP.1(2).

5.1.8.7 Extended: Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1
The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2
The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3
The TSF shall provide a means to verify firmware/software updates to the TOE using a [***digital signature mechanism***] and [***no other functions***] prior to installing those updates.

5.1.9 TOE access (FTA)

5.1.9.1 TSF-initiated Termination (FTA_SSL.3)

FTA_SSL.3.1

Refinement: The TSF shall terminate a remote interactive session after a Security Administrator-configurable time interval of session inactivity.

5.1.9.2 User-initiated Termination (FTA_SSL.4)

FTA_SSL.4.1

The TSF shall allow Administrator-initiated termination of the Administrator's own interactive session.

5.1.9.3 TSF-initiated Session Locking (FTA_SSL_EXT.1)

FTA_SSL_EXT.1.1

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

5.1.9.4 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1

Refinement: 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.

5.1.10 Trusted path/channels (FTP)

5.1.10.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1

Refinement: The TSF shall use [***IPsec***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***web management server***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data

FTP_ITC.1.2

The TSF shall permit the TSF, or the authorized IT entities to initiate communication via the trusted channel.

FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for [**connections to peers**].

5.1.10.2 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1

Refinement: The TSF shall use [***IPsec***] 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.

FTP_TRP.1.2

Refinement: The TSF shall permit remote administrators to initiate communication via the trusted path.

FTP_TRP.1.3

The TSF shall require the use of the trusted path for initial administrator authentication and all remote administration actions.

5.2 TOE Security Assurance Requirements

The SARs for the TOE are the EAL 1 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

Table 3 EAL 1 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 functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3c
The functional specification shall provide rationale for the implicit categorisation 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 fulfil 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
- 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

6.1 Security audit

The TOE generates audit events (see **Table 2** Auditable Events) and has the capability to store them internally or export them to the Security Management Server. 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 sends all audit logs to the Management Server. The TOE can be configured to use IPsec to protect audit logs exported to the Management Server. The Management Server has a disk cleanup procedure where it removes old audit logs to allow space for new ones. This is configurable by the authorized administrator. The TOE also has the ability to prevent new connections if the Management Server runs out of space for new audit logs.

Security Gateways maintain a queue of log records generated on the Gateway in memory, while they are being transmitted over the network to the defined log servers. If this queue is overrun, i.e. if the gateway 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 gateway 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 Gateway, 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.

When disk space on the Management Server falls below a predefined threshold, the server stops collecting audit records. As explained above, gateways will queue the records, and eventually start logging them to the local disk, until connectivity is resumed.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.

The Security audit function is designed to satisfy the following security functional requirements:

- FAU_GEN.1: The TOE is able to generate logs for a range of events as required by the NDPPe3/VPN/FW. 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.
- FAU_GEN.2: The TOE is able to identify each auditable event with specific IP addresses and the TOE's interfaces and gateways.

- 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.

6.2 Cryptographic support

The TOE includes a CAVP validated crypto module providing supporting cryptographic functions.

The following functions have been CAVP validated in accordance with the identified standards.

Functions	Standards	Cert
Encryption/Decryption		
• AES CBC and GCM (128 or 256 bits)	FIPS Pub 197 NIST SP 800-38A	3418
Cryptographic signature services		
• ECDSA Elliptic Curve Digital Signature Algorithm (modulus 2048)	FIPS Pub 186-3	685
Cryptographic hashing		
• SHA-256 , SHA-385 and SHA-512, (digest sizes 256, 384 and 512 bits)	FIPS Pub 180-3	2824
Keyed-hash message authentication		
• HMAC-SHA-256(digest size 256)	FIPS Pub 198-1 FIPS Pub 180-3	2176
Random bit generation		
• Hash_DRBG with sw based noise sources with a minimum of 256 bits of non-determinism	NIST SP 800-90	823

Table 4 CAVP Algorithms

The TOE generally fulfills all of the NIST SP 800-56A requirements without extensions. The TOE does not perform any operations marked as “shall not” or “should not” and performs all operations marked as “shall” or “should”. For finite-field based key establishment, the TOE implements the following sections of SP 800-56A: 5.6 and all subsections. Likewise, the TOE performs digital signature services that fulfills all of the FIPs Pub 186-3 requirements without extensions.

The TOE implements the IPsec architecture as specified in RFC 4301. SPD rules can be configured using the firewall rules and VPN communities. 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). 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 IKEv2 in tunnel mode. 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-256 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 IKEv2 uses AES-CBC-128, AES-CBC-256 as specified in RFC 6379. The TOE generates the secret value x used in the IKEv2 Diffie-Hellman key exchange ('x' in g^x mod p) using the Hash_DRBG specified in FCS_RBG_EXT.1 and having possible lengths of 224, 256, or 384 bits. When a random number is needed for a nonce, the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^{112}, 2^{128}, or 2^{192}. 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 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 /IKEv2 CHILD_SA connection.

The IPsec implementation supports both ECDSA certificates and pre-shared keys. The TOE only supports pre-shared keys when communicating with an externally managed gateway. This can allow either a remote administrator workstation or a non-Check Point Gateway to be configured for IPsec pre-shared keys. When Check Point Gateway appliances communicate, they use IPsec with certificate authentication. The authorized administrator can configure the TOE to support lifetimes based on elapsed time (measured in seconds) or on amount of data transmitted (measured in kilobytes).

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.

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
VPN PSK	On Disk	Never (may be replaced	
Password hash	On Disk	Never (may be replaced)	

Table 5 CSPs and Keys

The Cryptographic support function is designed to satisfy the following security functional requirements:

- FCS_CKM.1: The TOE supports NIST Special Publication 800-56A Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384, and P-521 (as defined in FIPS PUB 186-3, "Digital Signature Standard").
- FCS_CKM_EXT.4: The TOE provides means to zeroize all cryptographic key data when they are no longer required.
- FCS_COP.1(1): The TOE supports AES-CBC (as specified in NIST SP 800-38A) and AES-GCM (as specified in NIST SP 800-38D). Both AES modes support 128 and 256 bits.
- FCS_COP.1(2): The TOE supports 186-3/-4 ECDSA with curve sizes of P-256, P-384, and P-521.
- FCS_COP.1(3): The TOE supports SHA-256, SHA_384, and SHA-512.
- FCS_COP.1(4): The TOE supports HMAC-SHA-256.
- FCS_IPSEC_EXT.1: The TOE implements the IPsec architecture as specified in RFC 4301. See above for details
- FCS_RBG_EXT.1: The TOE supports a Hash-based SP 800-90 DRBG.

6.3 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 is designed to satisfy the following security functional requirements:

- 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

6.4 Stateful Traffic Filtering Firewall

Every IPv4/v6 packet received by the Check Point Security Gateway Appliances 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 its 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. If this is successful, the decapsulated packet contents are labeled with the VPN community on which the packet was received.

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

For all other packets, 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 configuring 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 address is defined as being a reserved address as specified in RFC 1918 for IPv4, and RFC 3513 for IPv6
9. Packets where the source or destination address of the network packet is a link-local address
10. Packets where the source or destination address of the network packet is defined as being an address 'reserved for future use' as specified in RFC 5735 for IPv4
11. 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
12. Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified

During the Check Point Security Gateway Appliances 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 Stateful Traffic Filtering Firewall function is designed to satisfy the following security functional requirements:

- FFW_RUL_EXT.1: The TOE supports all of the required protocols, which include icmpv4 (RFC 792), icmpv6 (RFC 4443), ipv4 (RFC 791), ipv6 (RFC 2460), tcp (RFC 793), and udp (RFC 768). Conformance with the RFCs defining these protocols is asserted by the Check Point based upon the 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 virtual gateways. Physical interfaces on the TOE hardware can be assigned to each virtual gateway. From the rules page, each rule can specify virtual gateways. The TOE supports FTP for stateful filtering. The TOE's (virtual) gateway 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.

6.5 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 or via the web interface. Passwords can be composed of any alphabetic, numeric, and a wide range of special characters (identified in FIA_PMG_EXT.1). 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 account is unlocked depending upon the configuration.

The TOE requires identification and authentication before allowing access. Only the banner may be presented before authentication is complete.

The TOE supports X.509v3 certificates for 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 web management server. The authorized administrator configures the VPN peers, 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). 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

The Identification and authentication function is designed to satisfy the following security functional requirements:

- FIA_AFL.1: The TOE supports the capability to lock an account indefinitely or for a period of time configurable via the TOE configuration.
- FIA_PMG_EXT.1: The TOE supports passwords that are at least 15 characters long. The password composition can contain all of the special characters required by this SFR.
- FIA_PSK_EXT.1: The TOE supports a pre-shared key length of 64 characters. (see the IPsec discussion in Section 6.2) Text based pre-shared keys are not conditioned and the TOE can accept bit-based pre-shared keys.
- FIA_UAU.7: Authentication data entered in by an operator is obscured during login.
- FIA_UAU_EXT.2: The TOE's authentication mechanism employs a locally stored database of authentication data.
- FIA_UIA_EXT.1: The TOE is able to display a warning banner in accordance with FTA_TAB.1
- FIA_X509_EXT.1: The TOE is able to use X.509v3 certificates. The TOE also performs all certificate checks that are required in this SFR.

6.6 Security management

User accounts are associated with profiles. User accounts associated with all privileges in their profile 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 be accessed from a directly connected terminal. These command line functions can be used to effectively manage every security policy, as well as the non-security relevant aspects of the TOE.

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

The Security management function is designed to satisfy the following security functional requirements:

- FMT_MOF.1: Only the TOE's administrators can enable, disable, determine and modify the behavior of all of the security functions of the TOE.
- FMT_MTD.1: Only administrators with administrator profiles in the TOE can access the TOE's security functions.
- FMT_SMF.1: The TOE allows administrators to manage the TOE and configure the TOE both locally and remotely. The TOE also allows administrators to update the TOE
- FMT_SMR.2: The TOE supports administrator roles. The TOE is able to create profiles for each configured administrator. An administrator can login to the TOE locally or remotely.

6.7 Packet filtering

The packet filtering function is the VPN extended package is addressed entirely by the FFW_RUL_EXT.1 requirement. See Section 6.4.

The Packet Filtering function is designed to satisfy the following security functional requirements:

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

6.8 Protection of the TSF

The TOE can be one or more appliances and a 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.

The TOE components are hardware appliances that includes a real-time clock. The TOE can be configured to synchronize its clock with a time server. 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 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.

During power-up algorithms are tested in the kernel and user-space. 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 operational 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 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 is designed to satisfy the following security functional requirements:

- FPT_APW_EXT.1: Passwords are store obfuscated from view. Local login passwords are stored as a Unix hash.
- FPT_FLS.1: The TOE contains self-tests that are executed during power-up. The TOE enters an error state when it fails its self-tests.
- FPT_ITT.1: The TOE protects data using the IPsec protocol.
- FPT_SKP_EXT.1: The TOE's remote management GUI does not provide any means for reading any key or CSP.
- FPT_STM.1: The TOE provides reliable time stamps using either an internal clock or an NTP server.
- FPT_TST_EXT.1: The TOE runs self-tests during power-up to demonstrate correct operation.
- FPT_TUD_EXT.1: The TOE's updates are digitally signed and verified using ECDSA with SHA-512 and P-521 curve.

6.9 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 console and TLS over IPsec sessions. The authorized administrator can set the inactivity timeout and it can be different of reach type of login (local and remote). 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 is designed to satisfy the following security functional requirements:

- FTA_SSL.3: The TOE allows inactive sessions to disconnect after a set period of time configurable in the GUI.
- FTA_SSL.4: The TOE allows session disconnect via a logout command.
- FTA_SSL_EXT.1: The TOE is able to terminate an administrator session after a set inactivity time.
- FTA_TAB.1: The TOE supports a message of the day banner that pops up when an operator authenticates to the TOE both locally and remotely.

6.10 Trusted path/channels

The TOE uses IPsec to protect communications. The TOE sends audit data to an audit server over an IPsec connection. The TOE also uses IPsec to protect remote administration. Authorized administrators connect to the TOE using an TLS over IPsec connection. In both cases, IPsec ensures traffic is not modified or disclosed

The Trusted path/channels function is designed to satisfy the following security functional requirements:

- FTP_ITC.1: The TOE uses IPsec to provide a trusted communication channel between itself and the web management server and audit server.
- FTP_TRP.1: 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.

7. Hardware Platforms

Below is a list of hardware platforms included in the evaluation. All platforms are x86 based hardware.

Note: The models identified using the '**' convention use a zero-justified numbering system for the licensed software blades, e.g. the 'Check Point 21412 Appliance' would support up to 12 software blade licenses, whereas the 'Check Point 21407' Appliance' would be the same hardware model supporting up to 7 blades

- Check Point 22** Appliances
- Check Point 42**, 44**, 46**, 48** Appliances
- Check Point 122**, 124**, 126**, 135**, 138** Appliances
- Check Point 214**, 216**, 217**, 218** Appliances

The following commodity hardware platforms are included in the evaluated configuration for Security Gateway and Security Management software, running the GAiA R77.30 operating system.

Check Point IAS appliances	D1, D2, D6, D8, R2, R6, R8
Fujitsu	Primergy RX100 S6, S7
	Primergy RX200 S6, S7
	Primergy RX300 S6, S7
HP	ProLiant DL120 G7
	ProLiant DL320e G8
	ProLiant DL360 G7
	ProLiant DL380 G7
	ProLiant DL360p G8
	ProLiant DL380p G8

The following Check Point security appliance models are included in the evaluated configuration for the Security Management software, running the GAiA R77.30 operating system:

- Smart-1 5
- Smart-1 25
- Smart-1 50
- Smart-1 150
- Smart-1 205
- Smart-1 210
- Smart-1 225
- Smart-1 3050
- Smart-1 3150