F5 BIG-IP® 14.1.0 for LTM+AFM Security Target

Document Number:	CC2019-ASE_ST-001
Document Version:	4.6
Date:	July 10, 2019

Prepared By:

Saffire Systems

PO Box 40295

Indianapolis, IN 46240

Prepared For:

F5 Networks, Inc.

801 Fifth Avenue

Seattle, WA 98104

Table of Contents

List of Tables

List of Figures

1 Introduction

This section identifies the Security Target, Target of Evaluation (TOE), conformance claims, ST organization, document conventions, and terminology. It also includes an overview of the evaluated product.

1.1 Security Target Identification

This section will provide information necessary to identify and control the Security Target and the TOE.

ST Title	F5 BIG-IP 14.1.0 for LTM+AFM Security Target
Version:	4.6
Publication Date:	July 10, 2019
Sponsor:	F5 Networks, Inc.
Developer:	F5 Networks, Inc.
ST Author	Michelle Ruppel, Saffire Systems

1.2 TOE Identification

The TOE claiming conformance to this ST is identified as *BIG-IP LTM+AFM Version 14.1.0.3* (build BIGIP-14.1.0.3.0.75.6-ENG, also referred to as 14.1.0.3) with any of the following hardware appliances installed with the LTM+AFM with application mode software and engineering hotfix Hotfix-BIGIP-14.1.0.3.0.75.6-ENG.

Explanation of table columns in the table below.

SKU (stock-keeping unit). A set of product SKUs define the hardware and software that is licensed and shipped. Each row in this table is a delivery option consisting of multiple product SKUs. The SKUs together define the following for appliances:

- Base BIG-IP and platform (F5-BIG-LTM-nnn)
- Additional modules (F5-ADD-BIG-AFM-nnn)
- Appliance mode (F5-ADD-BIG-MODE).

VIPRION devices are the same, but with the addition of VPR to the SKU, and the addition of a SKU specifying the chassis (for example F5-VPR-LTM-C2400-AC).

Note that “XXX” in the SKUs below denotes that the SKU is applicable to a range of platforms or models. “XXX” is part of the actual SKU and not a placeholder.

vCMP?. A “Y” entry in the column notes that the platform supports, and the licensing allows, the use of vCMP.

Part #. This refers to the part number of the hardware device (appliance, blade, and/or chassis) included in the platform SKU.

Model Series. Designates the family of appliances or blades to which the specified SKU belongs.

SKU	VCMP?	Part #	Model Series
F5-BIG-LTM-I5600 F5-ADD-BIG-AFM-I5XXX F5-ADD-BIG-MODE	N	200-0396-02	i5000
F5-BIG-LTM-I7600 F5-ADD-BIG-AFM-I7XXX F5-ADD-BIG-MODE	N	500-0003-03	i7000
F5-BIG-LTM-I10600 F5-ADD-BIG-AFM-I10XXX F5-ADD-BIG-MODE	N	500-0002-03	i10000
F5-BIG-LTM-I11600-DS F5-ADD-BIG-AFMI11XXX F5-ADD-BIG-MODE	Y	500-0015-03	i11000-DS
F5-BIG-LTM-I15600 F5-ADD-BIG-AFMI15XXX F5-ADD-BIG-MODE	N	500-0001-07	i15000
F5-BIG-LTM-I5800 F5-ADD-BIG-AFM-I5XXX F5-ADD-BIG-MODE	Y	200-0396-02	i5000
F5-BIG-LTM-I5820-DF F5-ADD-BIG-AFM-I5XXX F5-ADD-BIG-MODE	Y	500-0017-06	i5000
F5-BIG-LTM-I7800 F5-ADD-BIG-AFM-I7XXX F5-ADD-BIG-MODE	Y	500-0003-03	i7000
F5-BIG-LTM-I7820-DF F5-ADD-BIG-AFM-I7XXX F5-ADD-BIG-MODE	Y	500-0016-06	i7000
F5-BIG-LTM-I10800 F5-ADD-BIG-AFM-I10XXX F5-ADD-BIG-MODE	Y	500-0002-03	i10000

SKU	VCMP?	Part #	Model Series
F5-BIG-LTM-I11800-DS F5-ADD-BIG-AFMI11XXX F5-ADD-BIG-MODE	Y	500-0015-03	i11000-DS
F5-BIG-LTM-I15800 F5-ADD-BIG-AFMI15XXX F5-ADD-BIG-MODE	Y	500-0001-07	i15000
F5-VPR-LTM-C2400-AC F5-VPR-LTM-B2250 F5-ADD-VPR-AFM-C2400 F5-ADD-BIG-MODE F5-ADD-VPR-VCMP-2400	Y	400-0028-10 400-0039-03	C2400 B2250
F5-VPR-LTM-C4480-AC F5-VPR-LTM-B4450 F5-ADD-VPR-AFM-C4400 F5-ADD-BIG-MODE F5-ADD-VPR-VCMP-4480	Y	400-0033-04 400-0053-10	C4480 B4450
F5-BIG-LTM-10350V-F F5-ADD-BIG-AFM-10000 F5-ADD-BIG-MODE	Y	200-0398-00	10000 Series (FIPS)

Table 1: Supported Hardware Models

Each of the hardware platforms includes a third party proprietary cryptographic acceleration card. All hardware platforms, except the B2250, include the Intel Coleto Creek (8955). The B2250 and 10350V-F models include the Cavium Nitrox (CN3540-500-C20).

1.3 Document Terminology

Please refer to CC Part 1 Section 4 for definitions of commonly used CC terms.

1.3.1 ST Specific Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the CC Part 2 are not reiterated here, unless stated otherwise.

Administrators

Administrators are administrative users of the TOE, i.e. those users defined in the TOE to be authorized to access the configuration interfaces of the TOE. Different roles can be assigned to administrators, including the Administrator role -- the name of the role is not to be

confused with the general reference to an administrator being an administrative user of the TOE in any role.

User

Humans or machines interacting with the TOE via the provided user and programmatic interfaces. The TOE deals with different types of users -- administrators in charge of configuring and operating the TOE, traffic users who are subject to the TOE's firewalling capabilities. User interactions with the TOE are transparent to the user, and in most cases the users are not aware of the existence of the TOE.

1.3.2 Acronyms

ADF	Application Delivery Firewall
CC	Common Criteria
CMI	Central Management Infrastructure
CRL	Certificate Revocation List
CRLDP	Certificate Revocation List Distribution Point
DTLS	Datagram Transport Layer Security
EAL2	Evaluation Assurance Level 2
FPGA	Field-Programmable Gate Array
GUI	Graphical User Interface
HSB	High-Speed Bridge
HSL	High-Speed Logging
LTM	Local Traffic Manager
OSP	Organisational Security Policy
PP	Protection Profile
SFP	Security Function Policy
SFR	Security Functional Requirement
SOAP	Simple Object Access Protocol
SOF	Strength of Function
TLS	Transport Layer Security
TMM	Traffic Management Microkernel
TMOS	Traffic Management Operating System
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy
vCMP	Virtual Clustered Multi-Processing

1.4 TOE Type

The TOE type is a Firewall network device.

1.5 TOE Overview

The BIG-IP products subject to this evaluation represent Application Delivery Controllers based on F5's Traffic Management Operating System (TMOS). In particular,

- Application Delivery Firewall, which includes the Local Traffic Manager (LTM) and Advanced Firewall Manager (AFM) modules, provides network traffic management and firewall capabilities.

BIG-IP products run on appliance hardware provided by F5. In addition, BIG-IP running as a guest instance on F5 appliances that support F5's Virtual Clustered Multiprocessing (vCMP) environment is included. (vCMP implements a purpose-built hypervisor that allows organizations to run multiple virtual instances of BIG-IP on the same hardware.)

The TOE's Traffic Management Microkernel (TMM), along with additional software, provides basic networking functionality, with the TOE operating as a network switch and reverse proxy. This includes the following security functions:

- **Security Audit:** BIG-IP implements syslog capabilities to generate audit records for security-relevant events. In addition, the BIG-IP protects the audit trail from unauthorized modifications and loss of audit data due to insufficient space.
- **Cryptographic Support:** In BIG-IP, cryptographic functionality is provided by the OpenSSL cryptographic module. The BIG-IP provides a secure shell (SSH) to allow administrators to connect over a dedicated network interface. BIG-IP also implements the TLS protocol to allow administrators to remotely manage the TOE. BIG-IP implements a TLS client for interactions with other TLS servers. These cryptographic implementations utilize the cryptographic module which provides random number generation, key generation, key establishment, key storage, key destruction, hash operations, encryption/decryption operations, and digital signature operations.
- **User Data Protection:** BIG-IP implements residual information protection on network packets traversing through it. In other words, network packets traversing through the BIG-IP do not contain any residual data.
- **Identification and Authentication:** An internal password-based repository is implemented for authentication of management users. BIG-IP enforces a strong password policy and disabling user accounts after a configured number of failed authentication attempts.
- **Security Function Management:** A command line interface (available via the traffic management shell "tmsh"), web-based GUI ("Configuration utility"), a SOAP-based API ("iControl API"), and a REST-based API (“iControl REST API”) are offered to administrators for all relevant configuration of security functionality. The TOE manages configuration objects in a partition which includes users, server pools, etc. This includes the authentication of administrators by user name and password, as well as access control based on pre-defined roles and, optionally, groups of objects ("Profiles"). "Profiles" can be defined for individual servers and classes of servers that the TOE forwards traffic from clients to, and for traffic that matches certain characteristics, determining the kind of treatment applicable to that traffic. Management capabilities offered by the TOE include the definition of templates for certain configuration options. The management functionality also implements roles for separation of duties.
- **Protection of the TSF:** BIG-IP implements many capabilities to protect the integrity and management of its own security functionality. These capabilities include the protection of sensitive data, such as passwords and keys, self-tests, product update verification, and reliable time stamping.

- **TOE Access:** Prior to interactive user authentication, the BIG-IP can display an administrative-defined banner. BIG-IP terminates interactive sessions after an administrator-defined period of inactivity and allows users to terminate their own authenticated session.
- **Trusted Path / Channels:** The TOE protects remote connections to its management interfaces with TLS and SSH. The TOE also protects communication channels with audit servers using TLS.
- **Firewall:** The TOE offers basic firewall functionality, including stateful packet inspection and network address translation, and logic to mitigate denial-of-service attacks.

1.6 TOE Description

1.6.1 Introduction

Figure 1 provides a schematic example of the TOE's role and location in a networking environment. The F5 hardware hosting BIG-IP is depicted by the two redundant network devices in the diagram. In this example:

- Internet connections (dark red network connection) are mediated by BIG-IP to provide access to certain resources located in an organization's internal server pool (yellow network connection), for example to a web-based e-commerce system presenting a storefront to consumers
- Users in the organization's Intranet (orange network connection) also access resources in the server pools to interact with the internal server pool. Although not included in the TOE, BIG-IP provides server termination of traffic flowing to a backend server by implementing a TLS client protocol.
- Network administrators connect to BIG-IP via a dedicated network interface (dark green network connection) to administer the TOE
- The TOE is set up in a redundant failover configuration, with heartbeat monitoring and reporting via a data link between the two instances (light green connections)

When deployed as two redundant systems configured in an active/standby failover configuration, the two systems can synchronize their configuration data and provide state and persistence monitoring. The TOE will fail over to the redundant system while maintaining a secure configuration if failures the active device sends a request to the standby device or if the standby device detects missing heartbeats from the active device. The new active device will continue to enforce security policies for new (and possibly active) connections mediated by the TOE. BIG-IP uses CMI (Central Management Infrastructure), a proprietary protocol, for the incremental exchange of configuration data and failover status between TOE instances; CMI is encapsulated in TLS to provide integrity and confidentiality protections. In this configuration a physical network port will be dedicated on each device for the exchange of synchronization data and failover monitoring with the standby device. Failover / redundancy is not in the scope of the evaluated configuration.

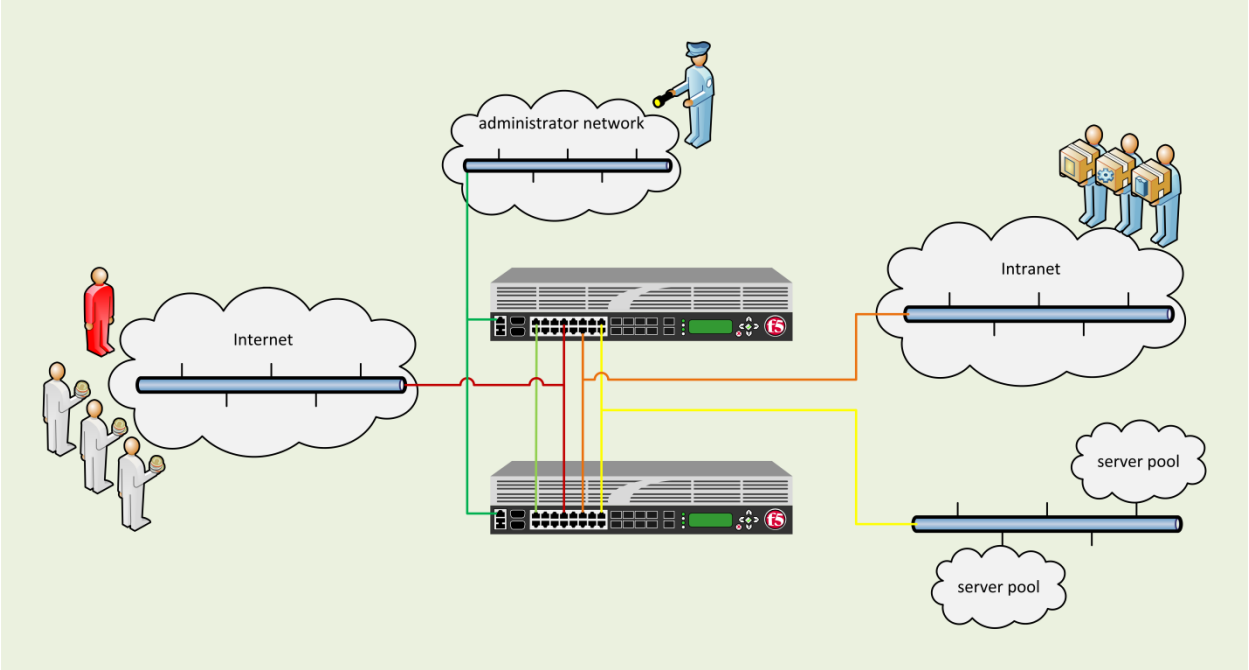

Figure 1: Schematic example of a BIG-IP network environment

1.6.2 Architecture Description

The TOE is separated into two (2) distinct planes, the control plane and the data plane. The control plane validates, stores, and passes configuration data to all necessary systems. It also provides all administrative access to the TOE. The data plane passes user traffic through the TOE.

The TOE implements and supports the following network protocols: TLS (client and server), SSH, HTTPS, FTP. The TOE protects remote connections to its management interfaces with TLS and SSH. The TOE also protects communication channels with audit servers using TLS (TLSv1.1 and TLSv1.2). The cryptographic functionality implemented in the TOE is provided by OpenSSL.

The TOE is divided into five (5) subsystems: Appliance (hardware or virtual), Traffic Management Operating System (TMOS), Traffic Management Micro-kernel (TMM), Local Traffic Manager (LTM), and Advanced Firewall Manager (AFM). F5's TMOS is a Linux-based operating system customized for performance and to execute on the TOE appliance hardware or in the TOE Virtual Clustered Multiprocessing (vCMP) environment. The vCMP is a hypervisor that allows multiple instances of the TOE to execute on the same underlying hardware. The TMM is the data plane of the product and all data plane traffic passes through the TMM. The LTM controls network traffic coming into or exiting the local area network (LAN) and provides the ability to intercept and redirect incoming network traffic. The AFM implements stateful traffic filtering on Level 2 and Level 4 network traffic packets using administrator-defined packet-filtering rules that are based on network packet attributes.

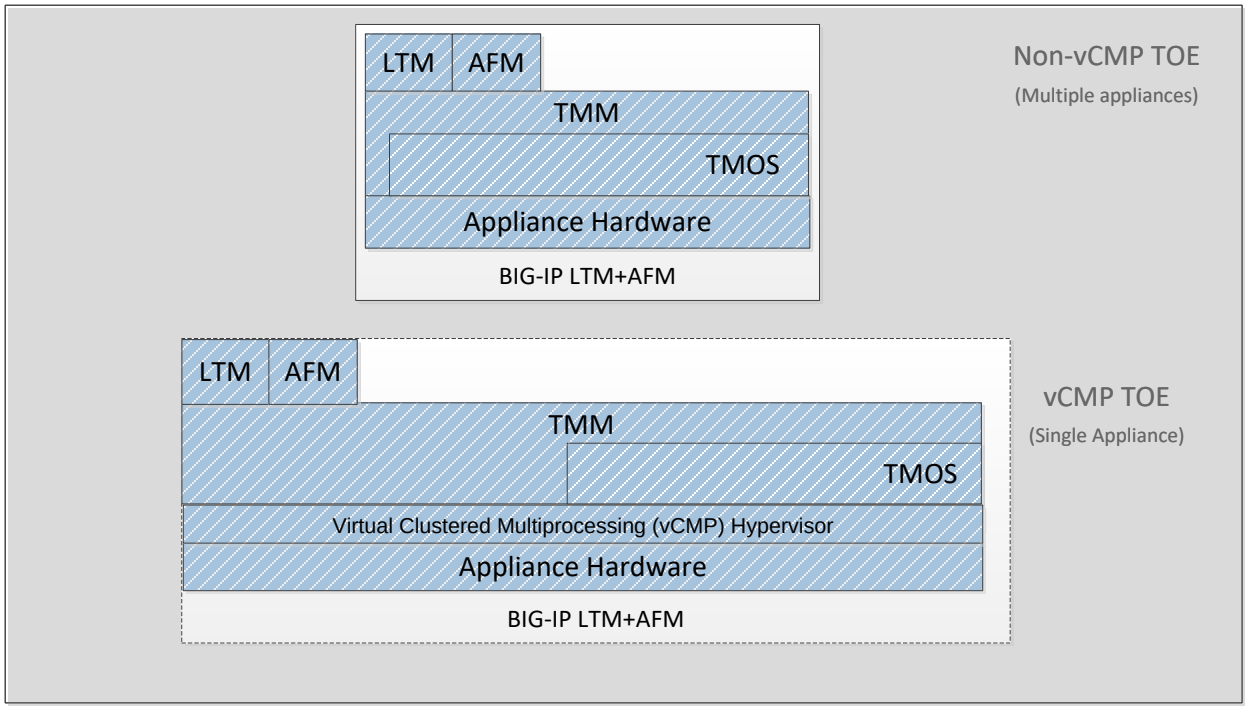

Figure 2: BIG-IP Subsystems

TMOS is a Linux operating system that runs directly on appliance hardware or in a vCMP environment. TMOS is a modified version of the RedHat Linux kernel. In addition to providing the standard operating system features (such as process management, file management, etc), the TMOS provides the following security features for the TOE:

- Auditing functionality, using the host system's syslog capabilities. (In addition, a concept called "high-speed logging" (HSL) allows TMM instances to send certain log traffic directly to external audit servers.)
- Time stamping
- Management functionality, presented to consumers via a dedicated shell providing a command line interface (traffic management shell, "tmsh") that can be reached by administrators via SSH (OpenSSH); and via a web GUI (“Configuration Utility”), a SOAP protocol interface ("iControl API"), or REST interface (“iControl REST API”) that can be reached through a network interface via HTTPS. Those management interfaces are implemented in the background by a central management control program daemon (mcpd) that provides configuration information to individual TOE parts and coordinates its persistent storage.
- Authentication functionality is enforced on all administrative interfaces. Administrative interfaces implement an internal password-based repository for authentication of administrative users.
- Cryptographic algorithms provided by OpenSSL.
- Individual daemons introduced by BIG-IP packages, such as the modules implementing the LTM and AFM logic.

At the core of BIG-IP is a concept referred to as Traffic Management Microkernel (TMM), representing the data plane of the product when compared to traditional network device architectures. It is implemented by a daemon running with root privileges, performing its own memory management, and having direct access to the network hardware. TMM implements a number of sequential filters both for

the “client-side” and “server-side” network interfaces served by BIG-IP. The filters implemented in TMM include a TCP, TLS, compression, and HTTP filter, amongst others. If the hardware provides more than one CPU, TMM runs multi-threaded (one thread per CPU). In this case, disaggregators implemented in hardware or, depending on the underlying appliance, firmware, are responsible for de-multiplexing and multiplexing network traffic for handling by an individual TMM thread. In addition to the actual switch hardware, F5 appliance hardware also contains a High-Speed Bridge (HSB, implemented by means of an FPGA) that performs basic traffic filtering functionality as instructed by TMM.

Additional plug-in filters can be added to this queue by individual product packages. These plug-ins typically have a filter component in TMM, with additional and more complex logic in a counter-part implemented in a Linux-based daemon (module). The plug-in modules relevant to this evaluation shown in Figure 3 include:

- Local Traffic Manager (LTM): authentication of HTTP (based on Apache) traffic and advanced traffic forwarding directives
- Advanced Firewall Manager (AFM): network filtering as described in FWcPP.

A diagram depicting aspects of the TOE’s architecture and the boundaries of the TOE are provided in Figure 3.

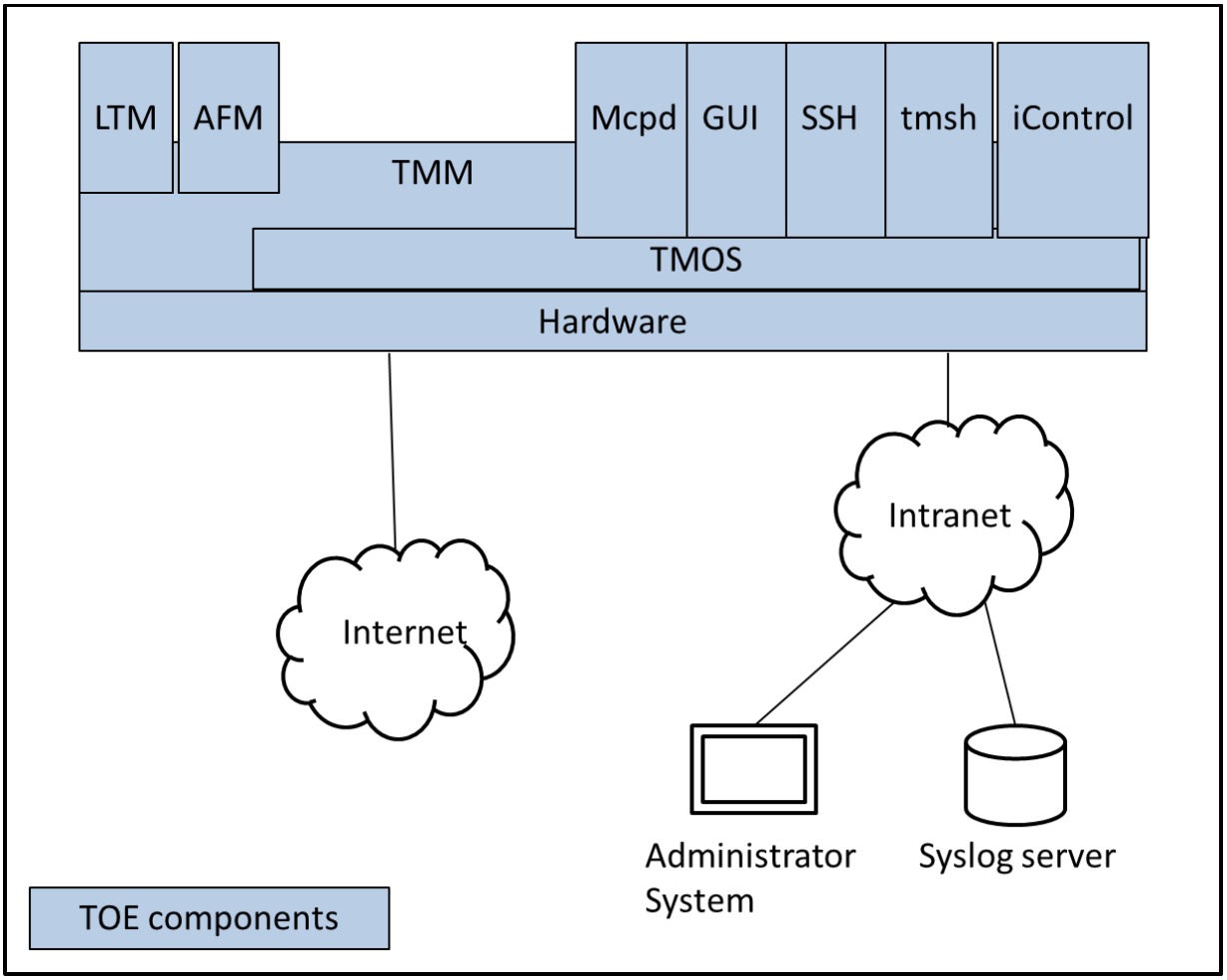

Figure 3: Architectural aspects of BIG-IP

1.6.3 Physical Boundaries

This section lists the hardware and software components of the product and denotes which are in the TOE and which are in the environment.

1.6.3.1 Physical boundaries

The TOE includes the hardware and software components as identified in Section 1.2.

The evaluated configuration of *BIG-IP LTM+AFM Version 14.1.0.3* represents a licensing option with the following F5 modules present and operational.

- Traffic Management Operating System (TMOS),
- Traffic Management Microkernel (TMM),
- Local Traffic Manager (LTM), and
- Advanced Firewall Manager (AFM).

The following required components can be found in the operating environment of the TOE on systems other than those hosting the TOE:

- audit servers.

Client software (e.g., the BIG-IP Client for TLS VPN connections, endpoint inspection software executed on clients) are optional components that are not part of the TOE.

1.6.3.2 Guidance Documentation

Relevant guidance documents for the secure operation of BIG-IP that are part of the TOE are:

- *BIG-IP Common Criteria Evaluation Configuration Guide BIG-IP LTM+AFM and BIG-IP LTM+APM Release 14.1.0*
- *K98644890: Common Criteria Certification for BIG-IP 14.1.0*
- *BIG-IP AFM: Network Firewall Policies and Implementations*
- *BIG-IP AFM Operations Guide*
- *BIG-IP Device Service Clustering: Administration*
- *BIG-IP Digital Certificates: Administration*
- *BIG-IP Engineering Hotfix README*
- *BIG-IP Local Traffic Manager: Implementations*
- *BIG-IP Local Traffic Manager: Monitors Reference*
- *BIG-IP Local Traffic Manager: Profiles Reference*
- *BIG-IP Release Note*
- *BIG-IP System: Essentials*
- *BIG-IP System: SSL Administration*
- *BIG-IP System: User Account Administration*
- *BIG-IP Systems: Getting Started Guide*
- *BIG-IP TMOS: Implementations*
- *BIG-IP TMOS: Routing Administration*
- *External Monitoring of BIG-IP Systems: Implementations*
- *GUI Help Files*
- *iControl SDK*
- *iControl REST API User Guide*

- *K12042624: Restricting access to the Configuration utility using client certificates (12.x – 14.x)*
- *K13092: Overview of securing access to the BIG-IP system*
- *K13123: Managing BIG-IP product hotfixes (11.x – 15.x)*
- *K13302: Configuring the BIG-IP system to use an SSL chain certificate (11.x – 14.x)*
- *K13454: Configuring SSH public key authentication on BIP-IP systems (11.x – 14.x)*
- *K14620: Managing SSL Certificates for BIG-IP systems using the Configuration utility*
- *K14783: Overview of the Client SSL profile (11.x – 14.x)*
- *K14806: Overview of the Server SSL profile (11.x – 15.x)*
- *K15497: Configuring a secure password policy for the BIG-IP system (11.x – 14.x)*
- *K15664: Overview of BIG-IP device certificates (11.x – 14.x)*
- *K42531434: Replacing the Configuration utility's self-signed SSL certificate with a CA-signed SSL certificate*
- *K5532: Configuring the level of information logged for TMM-specific events*
- *K6068: Configuring a pre-login or post-login message banner for the BIG-IP or Enterprise Manager system*
- *K7683: Connecting a serial terminal to a BIG-IP system*
- *K7752: Licensing the BIG-IP system*
- *K80425458: Modifying the list of ciphers and MAC algorithms used by the SSH service on the BIG-IP system or BIG-IQ system*
- *K9908: Configuring an automatic logout for idle sessions*
- *Platform Guide: 10000 Series*
- *Platform Guide: i5000/i7000/i10000/i11000 Series*
- *Platform Guide: i15000 Series*
- *Platform Guide: VIPRION® 2200*
- *Platform Guide: VIPRION® 4400 Series*
- *vCMP for Appliance Models: Administration*
- *vCMP for VIPRION Systems: Administration*
- *Traffic Management Shell (tmsh) Reference Guide (versions 14.1.0 and 12.0.0[1])*

1.6.4 Logical Boundaries

The following security functions provided by the TOE are described in more detail in the subsections below:

- Security Audit
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TSF

[1] The tmsh reference guide version 14.1.0 zipfile contains the pages for each of the tmsh commands. The 12.0.0 pdf contains additional general information that is still valid in 14.1.0 but not reproduced in the 14.1.0 zipfile.

- TOE Access
- Trusted Path/Channels
- Firewall

The following configuration specifics apply to the evaluated configuration of the TOE:

- Appliance mode is licensed. Appliance mode disables root access to the TOE operating system and disables bash shell.
- Certificate validation is performed using CRLs.
- Disabled interfaces:
  - All command shells other than tmsh are disabled. For example, bash and other user-serviceable shells are excluded.
  - Management of the TOE via SNMP is disabled.
  - Management of the TOE via the appliance's LCD display is disabled.
  - Remote (i.e., SSH) access to the Lights Out / Always On Management[2] capabilities of the system is disabled.
  - SSH client

1.6.4.1 Security Audit

BIG-IP implements auditing functionality based on standard syslog functionality. This includes the support of remote audit servers for capturing of audit records. Audit records are generated for all security-relevant events, such as the use of configuration interfaces by administrators, the authentication of traffic, and the application of network traffic rules.

While the TOE can store audit records locally for cases when an external log server becomes unavailable, in the evaluated configuration an external log server is used as the primary means of archiving audit records.

In the evaluated configuration, BIG-IP logs a warning to notify the administrator when the local audit storage exceeds a configurable maximum size. Once the configurable maximum size is reached, BIG-IP overwrites the oldest audit records.

1.6.4.2 Cryptographic Support

All cryptographic operations, including algorithms and key generation used by the TOE are provided by the F5 cryptographic module (OpenSSL) within the TMOS.

Various security functions in BIG-IP rely on cryptographic mechanisms for their effective implementation. Trusted paths for the TOE administrator are provided by SSH for the tmsh administrative interface and by TLS for the Configuration utility, iControl API and iControl REST API. For administrative sessions, the TOE always acts as a server. For traffic sessions, the TOE may act as a TLS client or server. Trusted channels between the TOE and external entities, such as a syslog server, are provided by TLS connections.

[2] Lights Out / Always On Management is an add-on module providing a management system for non-security related features not required for operation of the TOE.

For TLS sessions, the TOE implements certificate validation using the OpenSSL crypto library.

The TOE utilizes cryptographic algorithms that have been validated using the NIST CAVS tests.

The underlying hardware platforms of the TOE include a third party proprietary cryptographic acceleration card that is used to provide both sufficient entropy to support random number generation (RNG) and acceleration.

1.6.4.2.1 Key Generation

The TOE can generate asymmetric keys using RSA schemes and ECC schemes. The underlying hardware platforms of the TOE include a third party proprietary cryptographic acceleration card that is used to provide sufficient entropy to support RNG. The TOE provides a total of four entropy sources. The TOE can generate keys (and certificates) for a number of uses, including:

- Keypairs for the SSH server functionality
- TLS server and client certificates for the administrative sessions
- Session keys for SSH and TLS sessions

1.6.4.3 User Data Protection

BIG-IP is designed to ensure that it does not reuse old packet information when transmitting new packets through the device.

1.6.4.4 Identification and Authentication

The TOE identifies individual administrative users by user name and authenticates them by passwords stored in a local configuration database; the TOE can enforce a password policy based on overall minimum length and number of characters of different types required. BIG-IP obscures passwords entered by users.

Authentication of administrators is enforced at all configuration interfaces, i.e. at the shell (tmsh, via SSH), the Configuration utility (web-based GUI), iControl API, and iControl REST API.

1.6.4.5 Security Management

The TOE allows administrators to configure all relevant aspects of security functionality implemented by the TSF. For this purpose, BIG-IP offers multiple interfaces to administrators:

- Configuration utility
  The Configuration utility presents a web-based GUI available to administrators via HTTPS that allows administration of most aspects of the TSF.
- traffic management shell (tmsh)
  tmsh is a shell providing a command line interface that is available via SSH. It allows administration of all aspects of the TSF.
- iControl API
  The iControl API is a SOAP based protocol interface that allows programmatic access to the TSF configuration via HTTPS.
- iControl REST API
  The iControl REST API is effectively a front-end to tmsh and is built on the Representational State Transfer (REST), which allows programmatic access to the TSF via HTTPS.

The TOE provides the ability to administer the TOE both locally and remotely using any of the four

administrative interfaces. Local administration is performed via the serial port console. By default and in the evaluated configuration, remote access to the management interfaces is only made available on the dedicated management network port of a BIG-IP system.

BIG-IP implements a hierarchy of roles that are pre-defined to grant administrators varying degrees of control over the basic configuration of the TOE, and additional roles are introduced for module-specific tasks. These roles can be assigned to users by authorized administrators.

In addition to roles, the TOE allows the definition of partitions. Configuration objects, such as server pools or service profiles, can be assigned to individual partitions, as can administrative users. This allows administrative access of individual administrators to be restricted to configuration objects that belong to the partition that has been assigned to the user.

1.6.4.6 Protection of the TSF

The TOE is designed to protect critical security data, including keys and passwords. In addition, the TOE includes self-tests that monitor continue operation of the TOE to ensure that it is operating correctly. The TOE also provides a mechanism to provide trusted updates to the TOE firmware or software and reliable timestamps in order to support TOE functions, including accurate audit recording.

1.6.4.7 TOE access

The TOE implements session inactivity time-outs for Configuration utility and tmsh sessions and displays a warning banner before establishing an interactive session between a human user and the TOE.

1.6.4.8 Trusted Path/Channels

This chapter summarizes the security functionality provided by the TOE in order to protect the confidentiality and integrity of network connections described below.

1.6.4.8.1 Generic network traffic

The BIG-IP LTM allows the termination of data plane TLS connections on behalf of internal servers or server pools. External clients can thus connect via TLS to the TOE, which acts as a TLS server and decrypts the traffic and then forwards it to internal servers for processing of the content. It is also possible to (re-) encrypt traffic from the TOE to servers in the organization with TLS, with the TOE acting as a TLS client.

1.6.4.8.2 Administrative traffic

The TOE secures administrative traffic (i.e., administrators connecting to the TOE in order to configure and maintain it) as follows:

- Remote access to the traffic management shell (tmsh) is secured via SSH.
- Remote access to the web-based Configuration utility, iControl REST API, and iControl API is secured via TLS.

1.6.4.8.3 OpenSSH

The TOE SSH implementation is based on OpenSSH; however, the TOE OpenSSH configuration sets the implementation via the sshd_config as follows:

- Supports two types of authentication, RSA public-key and password-based
- Packets greater than (256*1024) bytes are dropped

- The transport encryption algorithms are limited to AES-CBC-128 and AES-CBC-256
- The transport mechanism is limited to SSH_RSA public key authentication
- The transport data integrity algorithm is limited to HMAC-SHA1 and HMAC-SHA2-256
- The SSH protocol key exchange mechanism is limited to ecdh-sha2-nistp256 and ecdh-sha2-nistp384.

1.6.4.8.4 Remote logging

The TOE offers the establishment of TLS sessions with external log hosts in the operational environment for protection of audit records in transfer.

1.6.4.9 Firewall

BIG-IP Version 12.1.3.4 LTM+AFM implements a full-featured stateful firewall for Level 3 / Level 4 network traffic, exceeding the requirements of the FWcPP.

Administrators can define packet filtering rules based on network packet attributes, such as the origin and destination IP addresses, ports, sequence number, code, etc. BIG-IP will only permit traffic to reach its intended destination if it matches such a rule, and does not violate certain other protocol characteristics that generally are considered to represent malicious traffic (such as IP packets specifying the Loose Source Routing option).

BIG-IP takes the state of stateful protocols into account when enforcing firewall rules. For example, TCP traffic will only be permitted if the TCP session was properly established and the initial packets match a firewall rule permitting such traffic.

In addition, the TOE implements SYN cookies in order to identify invalid TCP connection attempts and deal with SYN flooding attempts.

BIG-IP is also capable of generating dynamic rule sets for the FTP protocol which requires more than one connection.

1.6.5 Delivery

1.6.5.1 Hardware

The F5 BIG-IP hardware is manufactured and shipped via common carrier from an authorized subcontractor, Flextronics, headquartered in Milpitas, California. Manufacturing for the BIG-IP product consists of assembling the hardware, loading the BIG-IP software image onto the hard disk drive and performing test and inspection activities. Flextronics has been qualified by F5 Networks to manufacture, test, and deliver the BIG-IP product through an on-site assessment, process evaluation and F5 Networks Supplier Approval Program.

1.6.5.2 Software

The BIG-IP system arrives from the factory with the SKU-specified software pre-installed. However, to guard against potential tampering during shipping, customers are directed to reinstall the software from the F5 download website. Instructions for this are in the Guidance document.

1.6.5.3 Documentation

Administrator, Configuration, and Installation manuals are made available to customers on the F5 Website by product model number and applicable revision. Manuals are not shipped with the product.

In addition, an ISO of the customer documentation referenced by this evaluation is available in the same download directory as the product ISO. The documentation ISO, like the product ISO, is available only over a TLS or HTTPS connection. For additional security, the sha256 checksum of the ISO is also published with the ISO; its file name is the ISO file name concatenated with ".sha256".

2 Conformance Claims

2.1 CC Conformance Claims

This ST was developed to Common Criteria (CC) for Information Technology Security Evaluation – April 2017, Version 3.1, Revision 5, CCMB-2017-04-001

The ST claims to be:

CC Version 3.1 Part 2 extended

CC Version 3.1 Part 3 conformant

2.2 PP and Package Claims

The ST is claims conformance to the following Protection Profiles:

- collaborative Protection Profile for Stateful Traffic Filter Firewalls (FWcPP), Version 2.0 + Errata 20180314 (Version 2.0e), 14-March-2018 conformant

The ST is compliant with the following FWcPP technical decision:

NIAP TD	Applicability
0425 – NIT Technical Decision for Cut-and-paste Error for Guidance AA	Applicable
0423 – NIT Technical Decision for Clarification about application of Rfl#201726rev2	Applicable
0412 – NIT Technical Decision for FCS_SSHS_EXT.1.5 SFR and AA discrepancy	Applicable
0411 – NIT Technical Decision for FCS_SSHC_EXT.1.5, Test 1 - Server and client side seem to be confused	Not applicable. The TOE does not include FCS_SSHC_EXT.1.
0410 – NIT technical decision for Redundant assurance activities associated with FAU_GEN.1	Applicable
0409 – NIT decision for Applicability of FIA_AFL.1 to key-based SSH authentication	Applicable
0408 – NIT Technical Decision for local vs. remote administrator accounts	Applicable
0407 – NIT Technical Decision for handling Certification of Cloud Deployments	Applicable. The TOE is not a cloud deployment.
0402 – NIT Technical Decision for RSA-based FCS_CKM.2 Selection	Applicable
0401 – NIT Technical Decision for Reliance on external servers to meet SFRs	Applicable

0400 – NIT Technical Decision for FCS_CKM.2 and elliptic curve-based key establishment	Applicable
0399 – NIT Technical Decision for Manual installation of CRL (FIA_X509_EXT.2)	Applicable
0398 – NIT Technical Decision for FCS_SSH*EXT.1.1 RFCs for AES-CTR	Not applicable. The TOE SSH implementation does not claim compliance with AES-CTR RFC 4344.
0397 – NIT Technical Decision for Fixing AES-CTR Mode Tests	Not applicable. The FCS_COP.1/DataEncryption instance in the ST does not include AES-CTR mode.
0396 – NIT Technical Decision for FCS_TLSC_EXT.1.1, Test 2	Applicable
0395 – NIT Technical Decision for Different Handling of TLS1.1 and TLS1.2	Not applicable. The TOE does not include FCS_TLSS_EXT.2.
0394 – NIT Technical Decision for Audit of Management Activities related to Cryptographic Keys	Applicable
0343 – NIT Technical Decision for Updating FCS_IPSEC_EXT.1.14 Tests	Not applicable. The TOE does not include FCS_IPSEC_EXT.1.
0342 – NIT Technical Decision for TLS and DTLS Server Tests	Applicable
0341 – NIT Technical Decision for TLS wildcard checking	Applicable
0340 – NIT Technical Decision for Handling of the basicConstraints extension in CA and leaf certificates	Applicable
0339 – NIT Technical Decision for Making password-based authentication optional in FCS_SSHS_EXT.1.2	Applicable
0338 – NIT Technical Decision for Access Banner Verification	Applicable
0337 – NIT Technical Decision for Selections in FCS_SSH*_EXT.1.6	Applicable
0336 – NIT Technical Decision for Audit requirements for FCS_SSH*_EXT.1.8	Applicable
0335 – NIT Technical Decision for FCS_DTLS Mandatory Cipher Suites	Applicable

0334 – NIT Technical Decision for Testing SSH when password-based authentication is not supported	Not applicable. The TOE does not include FCS_SSHC_EXT.1.
0333 – NIT Technical Decision for Applicability of FIA_X509_EXT.3	Applicable
0324 – NIT Technical Decision for Correction of section numbers in SD Table 1	Applicable
0323 – NIT Technical Decision for DTLS server testing – Empty Certificate Authorities list	Not applicable. The TOE does not include FCS_DTLSS_EXT.2.
0322 – NIT Technical Decision for TLS server testing – Empty Certificate Authorities list	Not applicable. The TOE does not include FCS_TLSS_EXT.2.
0321 – Protection of NTP communications	Not Applicable. The TOE operational environment does not include an NTP server.
0291 – NIT Technical Decision for DH14 and FCS_CKM.1	Not Applicable. The TOE does not include DH group 14.
0290 – NIT Technical Decision for Physical Interruption of Trusted Path/Channel	Applicable
0289 – NIT Technical Decision for FCS_TLS_EXT.x.1 Test 5e	Applicable
0281 – NIT Technical Decision for Testing both thresholds for SSH rekey	Applicable
0262 – NIT Technical Decision for TLS server testing - Empty Certificate Authorities list Superseded by TD0322	Not applicable. The TOE does not include FCS_TLSS_EXT.2.
0260 – NIT Technical Decision for Typo in FCS_SSHS_EXT.1.4 Superseded by TD0337	Applicable
0259 – NIT Technical Decision for Support for X509 ssh rsa authentication IAW RFC 6187	Applicable
0257 – NIT Technical Decision for Updating FCS_DTLSC_EXT.x.2/FCS_TLSC_EXT.x.2 Tests 1-4	Applicable
0256 – NIT Technical Decision for Handling of TLS connections with and without mutual authentication	Applicable

0228 – NIT Technical Decision for CA certificates - basicConstraints validation	Applicable

The ST was also evaluated against the individual evaluation activities:

- Evaluation Activities for Network Device cPP, Version 2.0 + Errata 20180314, March 2018
- Evaluation Activities for Stateful Traffic Filter Firewalls cPP, Version 2.0, October-2017

2.3 Conformance Rationale

The ST is exactly conformant to the FWcPP V2.0 + Errata 20180314.

3 Security Problem Definition

A network device has a network infrastructure role it is designed to provide. In doing so, the network device communicates with other network devices and other network entities (an entity not defined as a network device) over the network. At the same time, it must provide a minimal set of common security functionality expected by all network devices. The security problem to be addressed by a compliant network device is defined as this set of common security functionality that addresses the threats that are common to network devices, as opposed to those that might be targeting the specific functionality of a specific type of network device. The set of common security functionality addresses communication with the network device, both authorized and unauthorized, the ability to perform valid or secure updates, the ability to audit device activity, the ability to securely store and utilize device and administrator credentials and data, and the ability to self-test critical device components for failures.

The TOE is intended to be used either in environments in which, at most, sensitive but unclassified information is processed, or the sensitivity level of information in both the internal and external networks is equivalent.

This security target includes a restatement of the Security Problem Definition (threats, organizational security policies, and assumptions) from FWcPP. The threats, organizational security policies and assumptions are repeated here for the convenience of the reader. Refer to the FWcPP for additional detail.

3.1 Threat Environment

This section describes the threat model for the TOE and identifies the individual threats that are assumed to exist in the operational environment of the TOE. Figure 1 supports the understanding of the attack scenarios discussed here.

The **assets** to be protected by the TOE are:

- Organizational data hosted on remote systems in physical and virtual network segments connected directly or indirectly to the TOE (depicted as "server pools" in Figure 1). (The TOE can be used to protect the assets on those systems from unauthorized exploitation by mediating network traffic from remote users before it reaches the systems or networks hosting those assets.)
- The TSF and TSF data

The **threat agents** having an interest in manipulating the TOE and TSF behavior to gain access to these assets can be categorized as:

- Unauthorized third parties (“attackers”, such as malicious remote users, parties, or external IT entities) which are unknown to the TOE and its runtime environment. Attackers are traditionally located outside the organizational environment that the TOE is employed to protect, but may include organizational insiders, too.
- Authorized users of the TOE (i.e., administrators) who try to manipulate configuration data that they are not authorized to access. TOE administrators, as well as administrators of the operational environment, are assumed to be trustworthy, trained and to follow the instructions provided to them with respect to the secure configuration and operation of the systems under their responsibility. Hence, only inadvertent attempts to manipulate the safe operation of the TOE are expected from this community.

The motivation of threat agents is assumed to be commensurate with the assurance level pursued by this evaluation, i.e., the TOE intends to resist penetration by attackers with an Enhanced-Basic attack potential.

3.2 Threats

The threats identified in this section may be addressed by the TOE, TOE environment, or a combination of both. The threat agents are authorized persons/processes, unauthorized persons/processes, or external IT entities not authorized to use the TOE itself. The threats identified assume that the threat agent is a person with a low attack potential who possesses an average expertise, few resources, and low to moderate motivation.

T.UNAUTHORIZED_ADMINISTRATOR_ACCESS

Threat agents may attempt to gain Administrator access to the firewall by nefarious means such as masquerading as an Administrator to the firewall, masquerading as the firewall to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between the firewall and a network device. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the firewall and the network on which it resides.

T.WEAK_CRYPTOGRAPHY

Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.

T.UNTRUSTED_COMMUNICATION_CHANNELS

Threat agents may attempt to target firewalls that do not use standardized secure tunneling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the firewall itself.

T.WEAK_AUTHENTICATION_ENDPOINTS

Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g., shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the firewall itself could be compromised.

T.UPDATE_COMPROMISE

Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.

T.UNDETECTED_ACTIVITY

Threat agents may attempt to access, change, and/or modify the security functionality of the firewall without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.

T.SECURITY_FUNCTIONALITY_COMPROMISE

Threat agents may compromise credentials and firewall data enabling continued access to the firewall and its critical data. The compromise of credentials include replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or firewall credentials for use by the attacker.

T.PASSWORD_CRACKING

Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the firewall. Having privileged access to the firewall provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.

T.SECURITY_FUNCTIONALITY_FAILURE

An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

T.NETWORK_DISCLOSURE

An attacker may attempt to "map" a subnet to determine the machines that reside on the network, and obtaining the IP addresses of machines, as well as the services (ports) those machines are offering. This information could be used to mount attacks to those machines via the services that are exported.

T.NETWORK_ACCESS

With knowledge of the services that are exported by machines on a subnet, an attacker may attempt to exploit those services by mounting attacks against those services.

T.NETWORK_MISUSE

An attacker may attempt to use services that are exported by machines in a way that is unintended by a site's security policies. For example, an attacker might be able to use a service to "anonymize" the attacker's machine as they mount attacks against others.

T. MALICIOUS_TRAFFIC

An attacker may attempt to send malformed packets to a machine in hopes of causing the network stack or services listening on UDP/TCP ports of the target machine to crash.

3.3 Organisational Security Policies

The TOE environment must include and comply with the following organizational security policies.

P.ACCESS_BANNER

The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

3.4 Assumptions

The assumptions are ordered into three categories: personnel assumptions, physical environment assumptions, and operational assumptions.

A.PHYSICAL_PROTECTION

The firewall device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the firewall's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the firewall and the data it contains. As a result, the cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will not expect the product to defend against physical access to the firewall that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the firewall.

A.LIMITED_FUNCTIONALITY

The firewall device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example the firewall device should not provide a computing platform for general purpose applications (unrelated to networking functionality).

A.TRUSTED_ADMINISTRATOR

The Security Administrator(s) for the firewall device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the firewall. The firewall device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.

A.REGULAR_UPDATES

The firewall device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

A.ADMIN_CREDENTIALS_SECURE

The Administrator's credentials (private key) used to access the firewall device are protected by the platform on which they reside.

A.RESIDUAL_INFORMATION

The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g., cryptographic keys, keying material, PINs, passwords, etc.) on firewall equipment when the equipment is discarded or removed from its operational environment.

4 Security Objectives

This chapter describes the security objectives for the TOE's operating environment (i.e., security objectives addressed by the IT domain or by non-technical or procedural means).

4.1 Security Objectives for the Operational Environment

The security objectives for the environment are listed below.

OE.PHYSICAL

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

Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

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.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.RESIDUAL_INFORMATION

The Secuity 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.

5 Extended Components Definition

All of the extended components used in this ST are taken from the FWcPP.

The FWcPP defines the following extended security functional requirements (SFRs). Refer to the FWcPP for the definition of these extended SFRs since they are not redefined in this ST.

Security Audit (FAU)

- FAU_STG_EXT.1

Cryptographic Support (FCS)

- FCS_HTTPS_EXT.1
- FCS_RBG_EXT.1
- FCS_SSHS_EXT.1
- FCS_TLSC_EXT.2
- FCS_TLSS_EXT.1

Identification and Authentication (FIA)

- FIA_PMG_EXT.1
- FIA_UIA_EXT.1
- FIA_UAU_EXT.2
- FIA_X509_EXT.1/Rev
- FIA_X509_EXT.2
- FIA_X509_EXT.3

Protection of the TSF (FPT)

- FPT_SKP_EXT.1
- FPT_APW_EXT.1
- FPT_STM_EXT.1
- FPT_TST_EXT.1
- FPT_TUD_EXT.1

TOE Access (FTA)

- FTA_SSL_EXT.1

Firewall (FFW)

- FFW_RUL_EXT.1
- FFW_RUL_EXT.2

6 Security Requirements

The security requirements that are levied on the TOE are specified in this section of the ST. Each of them are drawn from the FWcPP.

TOE Security Functional Requirements (from CC Part 2)		Required	Optional	Selection-Based
FAU_GEN.1	Audit Data Generation	√		
FAU_GEN.2	User Identity Association	√		
FAU_STG.1	Protected Audit Trail Storage		√	
FAU_STG.3/LocSpace	Display Warning for Local Storage Space		√	
FCS_CKM.1	Cryptographic Key Generation	√		
FCS_CKM.2	Cryptographic Key Establishment	√		
FCS_CKM.4	Cryptographic Key Destruction	√		
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)	√		
FCS_COP.1/SignGen	Cryptographic Operation (Signature Generation and Verification)	√		
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)	√		
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)	√		
FDP_RIP.2	Full Residual Information Protection	√		
FIA_AFL.1	Authentication Failure Management	√		
FIA_UAU.7	Protected Authentication Feedback	√		
FMT_MOF.1/ Services	Management of Security Functions Behaviour/Services		√	
FMT_MOF.1/ ManualUpdate	Management of Security Functions Behaviour/ManualUpdate	√		
FMT_MTD.1/CoreData	Management of TSF Data/CoreData	√		
FMT_MTD.1/CryptoKeys	Management of TSF Data/CryptoKeys		√	
FMT_SMF.1	Specification of Management Functions	√		
FMT_SMR.2	Restrictions on Security Roles	√		
FTA_SSL.3	TSF-initiated Termination	√		
FTA_SSL.4	User-initiated Termination	√		
FTA_TAB.1	Default TOE Access Banners	√		
FTP_ITC.1	Inter-TSF Trusted Channel	√		
FTP_TRP.1/Admin	Trusted Path	√		

Extended Security Functional Requirements		Required	Optional	Selection-Based
FAU_STG_EXT.1	Protected Audit Event Storage	√		
FCS_HTTPS_EXT.1	HTTPS Protocol			√
FCS_RBG_EXT.1	Random Bit Generation	√		
FCS_SSHS_EXT.1	SSH Server Protocol			√
FCS_TLSC_EXT.2[1]-[2]	TLS Client Protocol with authentication			√
FCS_TLSS_EXT.1[1]-[4]	TLS Server Protocol			√
FFW_RUL_EXT.1	Stateful Traffic Filtering	√		
FFW_RUL_EXT.2	Stateful Filtering of Dynamic Protocols		√	
FIA_PMG_EXT.1	Password Management	√		
FIA_UIA_EXT.1	User Identification and Authentication	√		
FIA_UAU_EXT.2	Password-based Authentication Mechanism	√		
FIA_X509_EXT.1/Rev	X.509 Certificate Validation			√
FIA_X509_EXT.2	X.509 Certificate Authentication			√
FIA_X509_EXT.3	X.509 Certificate Requests			√
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)	√		
FPT_APW_EXT.1	Protection of Administrator Passwords	√		
FPT_STM_EXT.1	Reliable Time Stamps	√		
FPT_TST_EXT.1	TSF Testing	√		
FPT_TUD_EXT.1	Trusted Update	√		
FTA_SSL_EXT.1	TSF-initiated Session Locking	√		

Table 2: Security Functional Requirements

6.1 Conventions

The CC defines four operations on security functional requirements. The conventions below define the conventions used in this ST to identify the operations completed in the PP and the operations completed in this ST by the ST author. Some of the operations completed in this ST by the ST author are the completion of selections of assignments relevant to on the PP. All operations completed in the ST are surrounded by square brackets ([operation]).

Assignment made in PP: indicated with *italics text*

Selection made in PP: indicated with underlined text

Refinement made in PP: additions indicated **with bold text**

deletions indicated with ~~strikethrough~~ ~~text~~

Iteration made in PP: indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash")

[*Assignment made in ST*]*:* indicated with [*italics text within brackets*]

[Selection made in ST]: indicated with [underlined text within brackets]

[Refinement made in ST]: additions indicated with [**bold text within brackets**]

deletions indicated with [**~~strikethrough bold text within brackets~~**]

Iteration made in ST: indicated with typical CC requirement naming followed by an iteration number in brackets, e.g., [1], [2], [3].

6.2 Security Functional Requirements

6.2.1 Security Audit (FAU)

6.2.1.1 FAU_GEN.1 Audit Data Generation

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).*
- *[Starting and stopping services];*

*d) Specifically defined auditable events listed in **[Table 3]**.*

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 3]***.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG.1	None.	None.
FAU_STG_EXT.1	None.	None.
FAU_STG.3/LocSpace	Low storage space for audit events.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SignGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session	Reason for failure.
FCS_RBG_EXT.1	None.	None.
FCS_SSHS_EXT.1	Failure to establish an SSH Session	Reason for failure.
FCS_TLSC_EXT.2[1]-[2]	Failure to establish a TLS Session	Reason for failure.
FCS_TLSS_EXT.1[1]-[4]	Failure to establish a TLS Session	Reason for failure.
FDP_RIP.2	None.	None.
FIA_AFL.1	Unsuccessful login attempt limits is met or exceeded.	Origin of the attempt (e.g., IP address)
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate	Reason for failure
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/Services	Starting and stopping of services.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	All management activities of TSF data.	None.
FMT_MTD.1/CryptoKeys	Management of cryptographic keys	None.
FMT_SMF.1	None.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time – either Administrator actuated or 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 in the FWcPP.	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1 (if “terminate the session” is selected)	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
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/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.
FFW_RUL_EXT.1	Application of rules configured with '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 Identifier of rule causing packet drop
FFW_RUL_EXT.2	Application of rules configured with 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE Interface

Table 3: Security Functional Requirements and Auditable Events

6.2.1.2 FAU_GEN.2 User Identity Association

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.

6.2.1.3 FAU_STG.1 Protected Audit Trail Storage

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to prevent unauthorised modifications to the stored audit records in the audit trail.

6.2.1.4 FAU_STG_EXT.1 Protected Audit Event Storage

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.

FAU_STG_EXT.1.2 The TSF shall be able to store generated audit data on the TOE itself.

FAU_STG_EXT.1.3 The TSF shall [overwrite previous audit records according to the following rule:

[log files are numbered and the oldest log file is deleted]] when the local storage space for audit data is full.

6.2.1.5 FAU_STG.3/LocSpace Action in case of possible audit data loss

FAU_STG.3.1/LocSpace The TSF shall *generate a warning to inform the Administrator* if the audit trail exceeds *the local audit trail storage capacity*.

6.2.2 Cryptographic Operations (FCS)

6.2.2.1 FCS_CKM.1 Cryptographic Key Generation

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] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;*

] ~~and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards]~~.

6.2.2.2 FCS_CKM.2 Cryptographic Key Establishment

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

- *RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "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, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography".*

] ~~that meets the following: [assignment: *list of standards*]~~.

6.2.2.3 FCS_CKM.4 Cryptographic Key Destruction

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 [*
  - *[**when not in the EEPROM**], logically addresses the storage location of the key and performs a [single-pass] overwrite consisting of [zeroes]*
  - *[**when in EEPROM**], logically addresses the storage location of the key and performs a [single-pass] overwrite consisting of [random data]]*

that meets the following: *No Standard*.

6.2.2.4 FCS_COP.1/DataEncryption Cryptographic operation (AES Data

Encryption/Decryption)

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]*.

6.2.2.5 FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)

FCS_COP.1.1SigGen 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 bits or greater],*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits or greater]*

]

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]; ISO/IEC 14888-3, Section 6.4

].

6.2.2.6 FCS_COP.1/Hash Cryptographic operation (Hash Operation)

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*] ~~and cryptographic key sizes [assignment: *cryptographic key sizes*]~~ **and message digest sizes [*160, 256, 384*] bits** that meet the following: *ISO/IEC 10118-3:2004*.

6.2.2.7 FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)

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]* and cryptographic key sizes [*for SHA-1 the key size is ≥ 160 bits, for SHA-256 the key size is ≥ 256 bits, for SHA-384 the key size is ≥ 384 bits used in HMAC*] **and message digest sizes [160, 256, 384] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

6.2.2.8 FCS_HTTPS_EXT.1 HTTPS Protocol

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3 If a peer certificate is presented, the TSF shall [*not establish the connection*] if the peer certificate is deemed invalid.

6.2.2.9 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [CTR_DRBG (AES)].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [[*two*] software-based noise source, [*two*] hardware-based noise source [**for the non-virtualization platforms except 10000 series],** [*one*] hardware-based noise source [**for 10000 series],** [*one*] hardware-based noise source [**for the VCMP guest platforms**]] with a minimum of [256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

6.2.2.10 FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs [4251, 4252, 4253, 4254, 5656, 6668].

FCS_SSHS_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [password-based].

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*256*1024*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [aes128-cbc, aes256-cbc].

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [ssh-rsa] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [hmac-sha1, hmac-sha2-256] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that [ecdh-sha2-nistp256] and [ecdh-sha2-nistp384] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8 The TSF shall ensure that within SSH connections the same keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

6.2.2.11 FCS_TLSC_EXT.2[1] TLS Client Protocol with authentication (TLS1.1)

FCS_TLSC_EXT.2.1[1] The **[data plane of the]** TSF shall implement [TLS 1.1 (RFC 4346)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492

○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492

].

FCS_TLSC_EXT.2.2[1] The **[data plane of the]** TSF shall verify that the presented identifier matches the reference identifier per RFC 6125 section 6.

FCS_TLSC_EXT.2.3[1] The **[data plane of the]** TSF shall only establish a trusted channel if the server certificate is valid. If the server certificate is deemed invalid, then the TSF shall [not establish the connection].

FCS_TLSC_EXT.2.4[1] The **[data plane of the]** TSF shall [present the Supported Elliptic Curves Extension with the following NIST curves: [secp256r1, secp384r1] and no other curves] in the Client Hello.

FCS_TLSC_EXT.2.5[1] The **[data plane of the]** TSF shall support mutual authentication using X.509v3 certificates.

6.2.2.12 FCS_TLSC_EXT.2[2] TLS Client Protocol with authentication (TLS 1.2)

FCS_TLSC_EXT.2.1[2] The **[data plane of the]** TSF shall implement [TLS 1.2 (RFC 5246)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

○ TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

○ TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268

○ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492

○ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492

○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492

○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492

○ TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246

○ TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246

○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289

○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

○ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

○ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

○ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289

○ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

].

FCS_TLSC_EXT.2.2[2] The **[data plane of the]** TSF shall verify that the presented identifier matches the reference identifier per RFC 6125 section 6.

FCS_TLSC_EXT.2.3[2] The **[data plane of the]** TSF shall only establish a trusted channel if the

server certificate is valid. If the server certificate is deemed invalid, then the TSF shall [not establish the connection].

FCS_TLSC_EXT.2.4[2] The **[data plane of the]** TSF shall [present the Supported Elliptic Curves Extension with the following NIST curves: [secp256r1, secp384r1] and no other curves] in the Client Hello.

FCS_TLSC_EXT.2.5[2] The **[data plane of the]** TSF shall support mutual authentication using X.509v3 certificates.

6.2.2.13 FCS_TLSS_EXT.1[1] TLS Server Protocol (Data Plane Server - TLS 1.1)

FCS_TLSS_EXT.1.1[1] The **[data plane of the]** TSF shall implement [TLS 1.1 (RFC 4346)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492

].

FCS_TLSS_EXT.1.2[1] The **[data plane of the]** TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [none].

FCS_TLSS_EXT.1.3[1] The **[data plane of the]** TSF shall [perform RSA key establishment with key size [2048 bits, 3072 bits]; generate EC Diffie-Hellman parameters over NIST curves [secp256r1 and secp384r1] and no other curves].

6.2.2.14 FCS_TLSS_EXT.1[2] TLS Server Protocol (Data Plane Server - TLS 1.2)

FCS_TLSS_EXT.1.1[2] The **[data plane of the]** TSF shall implement [TLS 1.2 (RFC 5246)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

○ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

○ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289

○ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

].

FCS_TLSS_EXT.1.2[2] The **[data plane of the]** TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [none].

FCS_TLSS_EXT.1.3[2] The **[data plane of the]** TSF shall [perform RSA key establishment with key size [2048 bits, 3072 bits]; generate EC Diffie-Hellman parameters over NIST curves [secp256r1 and secp384r1] and no other curves].

6.2.2.15 FCS_TLSS_EXT.1[3] TLS Server Protocol (Control Plane Server - TLS 1.1)

FCS_TLSS_EXT.1.1[3] The **[control plane of the]** TSF shall implement [TLS 1.1 (RFC 4346)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

○ TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

○ TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268

○ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492

○ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492

].

FCS_TLSS_EXT.1.2[3] The **[control plane of the]** TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [none].

FCS_TLSS_EXT.1.3[3] The **[control plane of the]** TSF shall [perform RSA key establishment with key size [2048 bits, 3072 bits]; generate EC Diffie-Hellman parameters over NIST curves [secp256r1] and no other curves].

6.2.2.16 FCS_TLSS_EXT.1[4] TLS Server Protocol (Control Plane Server - TLS 1.2)

FCS_TLSS_EXT.1.1[4] The **[control plane of the]** TSF shall implement [TLS 1.2 (RFC 5246] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

○ TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

○ TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268

○ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492

○ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492

○ TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246

○ TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246

○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC

5289
○ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
○ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
○ TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
○ TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288].

FCS_TLSS_EXT.1.2[4] The **[control plane of the]** TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [none].

FCS_TLSS_EXT.1.3[4] The **[control plane of the]** TSF shall [perform RSA key establishment with key size [2048 bits, 3072 bits]; generate EC Diffie-Hellman parameters over NIST curves [secp256r1] and no other curves].

6.2.3 User Data Protection (FDP)

6.2.3.1 FDP_RIP.2 Full Residual Information Protection

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.

6.2.4 Identification and Authentication (FIA)

6.2.4.1 FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*1 - 10*] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

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 Administrator defined time period has elapsed*].

6.2.4.2 FIA_PMG_EXT.1 Password Management

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 [**a value between** *15*] and [*255*].

6.2.4.3 FIA_UIA_EXT.1 User Identification and Authentication

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.

6.2.4.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local [password-based] authentication mechanism to perform local administrative user authentication.

6.2.4.5 FIA_UAU.7 Protected Authentication Feedback

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**.

6.2.4.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

FIA_X509_EXT.1.1/Rev The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation **supporting a minimum path length of three certificates**.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag is 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.*

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.

6.2.4.7 FIA_X509_EXT.2 X.509 Certificate Authentication

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

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [allow the administrator to choose whether to accept the certificate in these cases].

6.2.4.8 FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1 The TSF shall generate a Certificatoin 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].

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

6.2.5 Security Management (FMT)

6.2.5.1 FMT_MOF.1/Services Management of security functions behavior

FMT_MOF.1.1/Services The TSF shall restrict the ability to enable and disable the functions **and services** to *Security Administrators*.

6.2.5.2 FMT_MOF.1/ManualUpdate Management of security functions behavior

FMT_MOF.1.1/ManualUpdate The TSF shall restrict the ability to enable the functions *to perform manual update* to *Security Administrators*.

6.2.5.3 FMT_MTD.1/CoreData Management of TSF Data

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

6.2.5.4 FMT_MTD.1/CryptoKeys Management of TSF Data

FMT_MTD.1.1/CryptoKeys The TSF shall restrict the ability to *manage* the *cryptographic keys* to *Security Administrators*.

6.2.5.5 FMT_SMF.1 Specification of Management Functions

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 configure firewall rules;*
- *[*
  - *Ability to configure audit behavior;*
  - *Ability to configure the cryptographic functionality;*
  - *Ability to configure thresholds for SSH rekeying;*
  - *Ability to set the time which is used for time-stamps.]*

6.2.5.6 FMT_SMR.2 Restrictions on security roles

FMT_SMR.2.1 The TSF shall maintain the roles:

- *Security 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

- *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.

6.2.6 Protection of TSF (FPT)

6.2.6.1 FPT_APW_EXT.1 Protection of Administrator Passwords

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.

6.2.6.2 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

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

6.2.6.3 FPT_STM_EXT.1 Reliable Time Stamps

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

FPT_STM_EXT.1.2 The TSF shall [allow the Security Administrator to set the time].

6.2.6.4 FPT_TST_EXT.1/PowerOn TSF Testing (Extended)

FPT_TST_EXT.1.1/PowerOn The TSF shall run a suite of the following self-tests [during initial start-up (on power on), at the conditions *reboot*] to demonstrate the correct operation of the TSF: [*BIOS Power On at power on only, OpenSSL integrity at power on and reboot, software integrity at power on and reboot, , cryptographic algorithm at power on and reboot*].

6.2.6.5 FPT_TST_EXT.1/OnDemand TSF Testing (Extended)

FPT_TST_EXT.1.1/OnDemand The TSF shall run a suite of the following self-tests [at the request of the authorised user] to demonstrate the correct operation of the TSF: [*software integrity*].

6.2.6.6 FPT_TUD_EXT.1 Trusted Update

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 [the most recently installed version of the TOE firmware/software].

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].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [digital signature mechanism] prior to installing those updates.

6.2.7 TOE Access (FTA)

6.2.7.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

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

6.2.7.2 FTA_SSL.3 TSF-initiated Termination (Refinement)

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

6.2.7.3 FTA_SSL.4 User-initiated Termination (Refinement)

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

6.2.7.4 FTA_TAB.1 Default TOE Access Banners (Refinement)

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.

6.2.8 Trusted path/channels (FTP)

6.2.8.1 FTP_ITC.1 Inter-TSF trusted channel (Refinement)

FTP_ITC.1.1 The TSF shall **be capable of using [TLS] to** provide a **trusted** communication channel between itself and ~~**another trusted IT product**~~ **authorized IT entities supporting the following capabilities: audit server, [*no other capabilities*]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from ~~**modification or**~~ 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 [*transmission of syslog records to syslog audit servers*].

6.2.8.2 FTP_TRP.1/Admin Trusted Path (Refinement)

FTP_TRP.1.1/Admin The TSF shall **be capable of using [*SSH, TLS, HTTPS*] to** provide a communication path between itself and **authorized** remote **administrators** ~~users~~ 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.

FTP_TRP.1.2/Admin The TSF shall permit remote **administrators** ~~users~~ to initiate communication via the trusted path.

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

6.2.9 Firewall (FFW)

6.2.9.1 FFW_RUL_EXT.1 Stateful Traffic Filtering

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 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.

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.

FFW_RUL_EXT.1.4 The TSF shall allow the Stateful Traffic Filtering rules to be assigned to each distinct network interface.

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, [ICMP] 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. [ICMP: source and destination addresses, type, code].

b) Remove existing traffic flows from the set of established traffic flows based on the following: [session inactivity timeout, completion of the expected information flow].

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; 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;

e) 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;

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 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;

g) 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

h) [no other rules].

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.

FFW_RUL_EXT.1.8 The TSF shall process the applicable Stateful Traffic Filtering rules in an administratively defined order.

FFW_RUL_EXT.1.9 The TSF shall deny packet flow if a matching rule is not identified.

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].

6.2.9.2 FFW_RUL_EXT.2 Stateful Filtering of Dynamic Protocols

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].

6.3 TOE Security Assurance Requirements

The security assurance requirements (SARs) provide grounds for confidence that the TOE meets its security objectives (for example, configuration management, testing, and vulnerability assessment). The table below identifies the security assurance requirements drawn from CC Part 3: Security Assurance Requirements that are required by the FWcPP.

Assurance Class	Assurance Component ID	Assurance Component Name
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	Labeling of the TOE
	ALC_CMS.1	TOE CM coverage
ASE: Security Target evaluation	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification
ATE: Tests	ATE_IND.1	Independent testing – sample
AVA: Vulnerability assessment	AVA_VAN.1	Vulnerability survey

Table 4: Security Assurance Requirements

In addition, the TOE will provide the evidence necessary for the evaluators to perform the evaluation activities defined in the Evaluation Activities for Stateful Traffic Filter Firewalls and Evaluation Activities for Network Device cPP documents.

6.4 Security Requirements Rationale

This Security Target makes no modifications or additions to the FWcPP security problem definition, security objectives, or security assurance requirements. The security functionality requirements claimed in this ST include all of the required SFRs from the FWcPP, selected optional SFRs from the FWcPP, and the mandatory selection-based SFRs from the FWcPP. There are no additional SFRs or SARS included in this ST. Operations performed on the SFRs comply the corresponding Application Notes in the FWcPP.

6.4.1 Security Functional Requirement Dependencies

All of the security functional requirements claimed in this Security Target are taken directly from the FWcPP version 2.0e, and all operations on the SFRs have been completed correctly. Therefore, the dependency rationale used by the FWcPP version 2.0e is considered applicable and acceptable since the FWcPP has been approved.

7 TOE Summary Specification

This section presents a description of how the TOE SFRs are satisfied.

7.1 Security Audit

BIG-IP uses syslog functionality to generate audit records, including the start-up and shut-down of the audit functions themselves.

BIG-IP systems generate different log types that capture different types of audit records. The audit records include:

- audit events
  events related to the security and administrative functionality implemented by the TOE; this type of audit log captures most of the events specified in this ST
- system events
  events related to the TOE operating system as well as status of TOE components, such as the syslog-ng daemon
- packet filter events
  events related to packet filtering applied by the TOE
- local traffic events
  events related to network traffic handled by the system, including some events related to packet filtering

The TOE provides the ability to configure syslog levels per daemon that generates the respective audit records. The Configuration utility GUI and tmsh provide interfaces to set those log levels.

Depending upon the exact audit record, the outcome is included in the description and / or the status code.

Table 5 shows the information included in the different types of audit logs.

Log content		***Log type***				
		System	**Packet Filter**	**Local Traffic**	**Audit (mcp)**	**Audit (other)**
Description	The description of the event that caused the system to log the message.	X	X	X	X	X
Event	A description of the configuration change that caused the system to log the message.				X	
Host name	The host name of the system that logged the event message.	X	X	X		X
Service	The service that generated the event.	X	X	X		X
Session ID	The ID associated with the user session.					
Status code	The status code associated with the event.		X		X	

Log content		***Log type***				
		System	**Packet Filter**	**Local Traffic**	**Audit (mcp)**	**Audit (other)**
Timestamp	The time and date that the system logged the event message.	X	X	X	X	X
Transaction ID	The identification number of the configuration change initiated by another recorded event. This number can be used to trace back to the initiating audit entry and the associated user name.				X	
User Name	The name of the user who made the configuration change				X	X

Table 5: Audit Logs and Their Content

The TOE includes within each audit record the information required by FAU_GEN.1.2 and specified in Table 3.

For audit records logging the administrative task of generating/import of, changing, or deleting of cryptographic keys, the certificate key file object name is logged to identify the relevant key.

This functionality implements FAU_GEN.1 and FAU_GEN.2.

BIG-IP supports (and the evaluated configuration mandates) logging to external syslog hosts. Audit records in transit to the remote host are protected by TLS channels.

The syslog mechanism provided by the underlying Linux system (which is the operating system of the TOE) is used for the creation and forwarding of audit records. In the evaluated configuration, all audit records are sent to both local and remote storage automatically. The audit records are sent to the remote storage immediately. In addition, BIG-IP implements a high-speed logging mechanism for data traffic (logging packet filter events and local traffic events) in TMM that is compatible with syslog. The TOE supports TLS channels to audit servers for the protection of audit records sent from the TOE to an external audit server.

For the case that the remote syslog host becomes unavailable, audit records are stored locally in syslog files managed, and protected against unauthorized access, by using file permission bits in the underlying Linux host. The TOE will attempt to periodically reestablish the connection with the remote syslog host indefinitely. The TOE retries within seconds of each connection failure. The TOE implements a buffer to store audit records collected during the period of time when the remote syslog host is unavailable. If the connection is reestablished before the buffers overflow, no audit records are lost. If the connection is reestablished after the buffers overflow, audit records are lost. Locally stored audit records are also available for review through the administrative interfaces of the TOE. Only users in the Administrator role can modify those records. The TOE does not support deletion of audit records by authorized users.

BIG-IP logs a warning if the local space for syslog files on the box exceeds a configurable maximum size. The TOE implements a local syslog file rotation scheme that numbers the locally archived syslog files. The TOE will delete the oldest syslog file once the maximum size for local syslog file space is exceeded. A cron job runs every two minutes to check the audit trail storage partition in order to accomplish this. The evaluated configuration requires allocation of 7 GB of audit storage, and a warning to be logged when 90

% of the storage space are exhausted. The administrator receives the warnings when reviewing the log files as instructed the CC guidance document.

This functionality implements FAU_STG.1, FAU_STG_EXT.1, and FAU_STG.3/LocSpace.

7.2 Cryptographic Support

The TOE utilitizes cryptographic algorithms that have been validated using the NIST CAVS tests.

Higher-level protocol stacks can use the F5 cryptographic module (OpenSSL) in order to implement trusted traffic communications:

- Management GUI (browser client to TOE)
- SSH session for tmsh (SSH client to SSH server on TOE)
- Remote logging via syslog (TOE to syslog server)
- TLS user traffic intended to pass through the TOE to internal servers

Replay detection (and rejection) is inherent to the protocols used by BIG-IP to establish communications of a trusted nature, i.e. TLS/HTTPS and SSH.

7.2.1 Key Generation and Establishment

The session keys are generated upon the request of an administrator by a Key Generator process that invokes the OpenSSL library on the Linux host.

The TOE generates asymmetric cryptographic keys that are compliant with FIPS PUB 186-4 and meet the following:

Key Generation Scheme	Key Establishment Scheme	Key sizes / NIST curves	Usage
RSA	RSA NIST SP 800-56B	Key sizes: 2048, 3072	TLS certificate TLS ephemeral session keys SSH key pair The TLS static keys are created once, imported to the TOE, and stored on disk until the Administrator creates a new key. The SSH key pair is crated on first boot. The TOE can act as a receiver or both sender and receiver depending upon the deployment. When acting as a receiver, decryption errors are handled in a side channel resistant method and reported as MAC errors.

Key Generation Scheme	Key Establishment Scheme	Key sizes / NIST curves	Usage
ECC	ECC NIST SP 800-56A	NIST curves: P-256, P-384	For ECDHE and ECDSA in TLS. The TOE can act as a receiver or both sender and receiver depending upon the deployment.

Table 6: Key generation in the TOE

The TOE also generates TLS session keys and SSH session keys.

The TOE offers administrative interfaces for creating a private key and certificate signing request (CSR). See Section 7.4.2 for more information on CSRs.

This implements FCS_CKM.1 and FCS_CKM.2.

7.2.2 Zeroization of Critical Security Parameters

"Cryptographic Critical Security Parameters" are defined in FIPS 140-2 as "security-related information (e.g., secret and private cryptographic keys, and authentication data such as passwords and PINs) whose disclosure or modification can compromise the security of a cryptographic module." Only the TLS and SSH session keys are stored in plaintext form. The rest of the keys are stored in encrypted format. The encrypted keys are stored using the F5 Secure Vault. The F5 Secure Vault uses a Master Key and a Unit Key to protect sensitive configuration attributes. The Master Key is a single, symmetric key that is stored with the data and is used to protect the data. All sensitive configuration attributes, including passwords and passphrases, are encrypted with the Master Key using 128 bit AES encryption. The Unit Key (a key-encrypting-key) is a symmetric key stored in the EEPROM that is associated with the device and is used to protect the Master Key. The Master Key is encrypted with the Unit Key using 256 bit AES encryption. If the Unit Key is replaced in the EEPROM, the old Unit Key is cleared by overwriting it with random data and then the new Unit Key is written to EEPROM.

The following table discusses how the F5 cryptographic module (i.e. OpenSSL used by both data plane and control plane) zeroize critical security parameters that are not needed for operation of the TSF anymore. OPEN_SSL_cleanse() is used to zeroize data, and this routine has been updated to overwrite with zeros, not with pseudo-random data. This also includes key material used by the TSF that is stored outside of the F5 cryptographic module. Keys in volatile and non-volatile storage are destroyed by performing a single overwrite consisting of zeroes.

Application	**Key type**	**Storage Location**	**Volatile/ Non-volatile**	**Zeroized when?**	**Description**
Key generation	seeds, prime numbers	Stack/heap	Volatile	After each key has been generated.	These are zeroized in OpenSSL by calling OPENSSL_cleanse(), which overwrites the memory upon release

Application	Key type	Storage Location	Volatile/ Non-volatile	Zeroized when?	Description
TLS	Session keys	Stack/heap	Volatile	After session has ended	The TLS session keys are created within OpenSSL during session initiation. These are zeroized in OpenSSL by calling OPENSSL_cleanse(), which overwrites the memory upon release
TLS	private keys in TLS certificates	On the disk	Non-volatile	Upon deletion by administrator.	Private keys are zeroized when they are deleted by the administrator. Zeroization is done by overwriting the file once with zeroes and deleting the file. The API used for zeroization is the write(2) system call which is called with buffer filled with zeros as input.
SSH	Session keys	Stack/heap	Volatile	After session has ended	The SSH session keys are created within OpenSSL during session initiation. These are zeroized in OpenSSL by calling OPENSSL_cleanse(), which overwrites the memory upon release
SSH	SSH keys	On the disk	Non-volatile	Upon deletion by administrator.	SSH keys are zeroized when using the key-swap utility. Zeroization is done by overwriting the file once with zeroes and deleting the file. The API used for zeroization is the shred(1) Linux command which uses the write(2) system call which is called with buffer filled with zeros as input.

Table 7: Zeroization of Critical Security Parameters

This implements FCS_CKM.4.

7.2.3 Cryptographic operations in the TOE

The following table summarizes the implementation of cryptographic operations in the TOE:

Algorithm	Key length (bits)	Purpose	Reference	SFR
AES (CBC, GCM modes)	128 256	payload encryption	AES as specified by ISO 18033-3 CBC as specified in ISO 10116 GCM as specified in ISO 19772	FCS_COP.1/DataEncryption
RSA	Modulus of 2048 or greater	certificate-based authentication, key exchange	FIPS PUB 186-4 Section 5.5 using RSASSA-PKCS1v1_5, ISO/IEC 9796-2	FCS_COP.1/SigGen
ECDSA	256 bits or greater NIST curves: P-256, P-384, and no other	certificate-based authentication, key exchange	FIPS PUB 186-4 Section 6 and Appendix D ISO/IEC 14888-3 Section 6.4	FCS_COP.1/SigGen
SHA-1 SHA-256 SHA-384	none	certificate-based authentication / digital signature verification	ISO/IEC 10118-3:2004	FCS_COP.1/Hash
HMAC-SHA-1	Key sizes: ≥ 160 bits Hash Function: SHA-1 Message digest sizes: 160 bits Block size: 512 bits Output MAC length: 160 bits	message integrity	ISO/IEC 9797-2:2011, Section 7	FCS_COP.1/KeyedHash

Algorithm	Key length (bits)	Purpose	Reference	SFR
HMAC-SHA-256	Key sizes: ≥ 256 bits Hash Function: SHA-256 Message digest sizes: 256 bits Block size: 512 bits Output MAC length: 256 bits	message integrity	ISO/IEC 9797-2:2011, Section 7	FCS_COP.1/KeyedHash
HMAC-SHA-384	Key sizes: ≥ 384 bits Hash Function: SHA-384 Message digest sizes: 384 bits Block size: 1024 bits Output MAC length: 384 bits	message integrity	ISO/IEC 9797-2:2011, Section 7	FCS_COP.1/KeyedHash
Random Bit Generation	none	key generation	ISO/IEC 18031:2011 using CTR DRBG (AES)	FCS_RBG_EXT.1

Table 8: Cryptographic primitives in the TOE

7.2.4 Random Number Generation

The TOE transfers one or more random bit-streams from the defined entropy sources to the Linux operating system's entropy pool. The entropy pool is used as a seed source for a digital random number generator (DRNG) via the /dev/random and /dev/urandom special file interfaces. The bit-stream will be transferred as necessary during system operation. The defined sources will be specific to the hardware available on each platform but will include one or more of the following: the jitterentropy-engine, Cavium Nitrox III hardware, Intel QAT hardware, and the Intel rdrand instruction.

The random bit stream from the entropy source will be fed to the Linux DRNG on demand, such that if the entropy in the Linux DRNG runs low (and thus the threshold that causes /dev/random to block will reached soon), fresh entropy is inserted and the entropy estimate in the Linux RNG is increased. This will attempt to ensure that sufficient entropy is available in the Linux DRNG to avoid blocking applications that read from /dev/random, or will release any applications that have become blocked. Since the /dev/urandom interface also draws from the Linux kernel entropy pool input of the random bit stream will also ensure that /dev/urandom is initialized and reseeded. The increase in the entropy estimate caused by the transfer of the random bit stream is not equal to the number of bits transferred, rather it scaled by a

factor which is dependent on the entropy source.

This implements FCS_RBG_EXT.1.

7.2.5 SSH

The TOE implements a SSH v2 server and a SSH v2 client. The SSH client is not used for communication with trusted external IT entities and will be disabled in the TOE. Administrators can connect to the TOE remotely using SSH via a dedicated network interface. Administrators are authenticated locally by user name and password; remote authentication (via LDAP or AD) is not supported by the TOE.

The SSH implementation is compliant with RFCs 4251, 4252, 4253, 4254, 5656, 6668.

SSH connections to the TOE's command line interface are protected using SSH version 2, using transport encryption algorithm AES CBC mode with 128 and 256 bit-sizes keys, transport data integrity protection hashing algorithm HMAC-SHA1 and HMAC-SHA2-256 and public key authentication algorithm ssh-rsa. The SSH implementation monitors packet size on all channels and limits packet size as suggested in RFC 4253 Section 6.1; the maximum packet size is (256*1024) bytes with larger packets being silently dropped. Additionally, the SSH implementation has hard-coded ecdh-sha2-nistp256 and ecdh-sha2-nistp384 key exchange; diffie-hellman-group1-sha1 key exchange is intentionally disabled.

The SSH connection session key will be renegotiated after either of two thresholds has been reached. SSH connection session keys will be renegotiated after one hour of use. In addition, the SSH connection session key will be renegotiated after an administrator-configured maximum amount of data, the RekeyLimit, is transmitted over the connection. The administrative guidance will instruct the user to not set the RekeyLimit to a value greater that 1 GB.

This functionality implements FCS_SSHS_EXT.1.

7.2.6 TLS Protocol

The TOE implements both the TLS server and TLS client protocol.

Administrators remotely connect to the TOE via an HTTPS server implementing TLS over a dedicated network interface used to administer the TOE. Administrators are authenticated locally by user name and password; remote authentication (via LDAP or AD) is not supported by the TOE. Administrator sessions that use the web-based Configuration utility, SOAP protocol (iControl API), or the REST API (iControl REST API) are protected by TLS. TLS sessions are limited to TLS versions 1.2 and 1.1, using the cipher suites identified in Table 9. The TLS server implementation in the TOE will deny SSL 1.0, SSL 2.0, SSL 3.0, and TLS 1.0 session requests.

The TOE implementation of TLS client is capable of presenting a certificate to a TLS server for TLS mutual authentication. The TLS client implemented by the TOE is used to communicate with the external audit server.

The following table summarizes the cipher suites supported by the evaluated configuration for TLS, and SSH connections. All other proposed cipher suites are rejected.

Cipher	Data Plane Client	Data Plane Server	Control Plane Server
TLS_RSA_WITH_AES_128_CBC_SHA	TLS v1.1 TLS v1.2	TLS v1.1 TLS v1.2	TLS v1.1 TLS v1.2

TLS_RSA_WITH_AES_256_CBC_SHA	TLS v1.1 TLS v1.2	TLS v1.1 TLS v1.2	TLS v1.1 TLS v1.2
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	TLS v1.1 TLS v1.2	TLS v1.1 TLS v1.2	TLS v1.1 TLS v1.2
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	TLS v1.1 TLS v1.2	TLS v1.1 TLS v1.2	TLS v1.1 TLS v1.2
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	TLS v1.1 TLS v1.2	TLS v1.1 TLS v1.2	N/A
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	TLS v1.1 TLS v1.2	TLS v1.1 TLS v1.2	N/A
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS v1.2	TLS v1.2	TLS v1.2
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS v1.2	TLS v1.2	TLS v1.2
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	TLS v1.2	TLS v1.2	N/A
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	TLS v1.2	TLS v1.2	N/A
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	TLS v1.2	TLS v1.2	N/A
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	TLS v1.2	TLS v1.2	N/A
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	TLS v1.2	TLS v1.2	TLS v1.2
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	TLS v1.2	TLS v1.2	TLS v1.2
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	TLS v1.2	TLS v1.2	TLS v1.2
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	TLS v1.2	TLS v1.2	TLS v1.2
TLS_RSA_WITH_AES_128_GCM_SHA256	N/A	N/A	TLS v1.2
TLS_RSA_WITH_AES_256_GCM_SHA384	N/A	N/A	TLS v1.2

Table 9: Cipher suites

When acting as a TLS server, BIG-IP does not operate on or process reference identifier fields in the BIG-IP certificate. It's up to an Administrator to load the desired X.509 certificate and up to TLS clients to verify it.

The BIG-IP TLS server only checks the Common Name (CN) and DNS name in SAN when BIG-IP performs client authentication. For BIG-IP acting as TLS client, the TOE checks Common Name (CN) and DNS name. The BIG-IP TLS client supports ECDH in the Client Hello by default. This can optionally be disabled by removing the corresponding cipher suites, although individual curves cannot be configured. The DN or SAN in the certificate is compared by requiring an exact match.

Use of wildcards for reference identifiers constructed by the TOE and certificate pinning for TLS client connections are not supported by the TOE.

When acting as a TLS server, BIG-IP generates key establishment parameters using RSA with key size 2048 and 3072 bits and over NIST curves secp256r1 and secp384r1 (secp384r1 is not available on the TLS server in the control plane). The TLS server key exchange message parameters (ECDH) are as defined / required by RFC 5246 Section 7.4.3 for TLS 1.2, RFC 4346 Section 7.4.3 for TLS 1.1, and RFC 4492. For example, its classic ECDH using named curves with predefined parameters. The TOE does not support DHE_RSA cipher suites, so server key exchange messages are not sent.

This functionality implements FCS_TLSC_EXT.2[1]-[2], FCS_TLSS_EXT.1[1]-[4].

7.2.7 HTTPS Protocol

The BIG-IP provides three interfaces for remote administrators that communicate over HTTPS: Configuration Utility, iControl API, and iControl REST API. HTTP over TLS (HTTPS) is an application-level protocol for distributed, collaborative, hypermedia information systems transmitted over a TLS connection. Checking the validity of peer certificates is described in Section 7.4.2.

The TOE implements HTTPS per RFC 2818, HTTP over TLS. The HTTPS implementation is designed to comply with all mandatory portions of RFC 2818 (as denoted in the RFC by keywords “MUST”, “MUST NOT”, and “REQUIRED”). The optional portions of the RFC are denoted in the RFC by keywords “SHOULD”, “SHOULD NOT”, and “MAY”. Connection Initiation, Connection Closure, Client Behavior, Server Behavior, and Server Identity are implemented. For Connection Closure, the TOE includes a configuration setting in the SSL profile that controls alert protocols and the session close behavior. By default, the TOE is configured to close the underlying TCP connections without exchanging the required TLS shutdown close notify. The TOE can be configured to perform a clean shutdown of all TLS connections by sending a close notify.

This functionality implements FCS_HTTPS_EXT.1.

7.3 User Data Protection

Per the FWcPP application note for FDP_RIP.2, “Resources” in the context of FDP_RIP.2 requirement are network packets being sent through the TOE (as opposed to “to”), therefore, the concern is that outgoing network packets do not unknowingly contain data that contains residual information.

Each outgoing packet is comprised of data from one or more segments of physical memory; each linked with a header that contains the start address and the number of bytes to be written into a part of the outgoing packet. When the packet is ready to be transmitted, for each segment that makes the packet, the corresponding physical address and bytes (obtained from the header for that piece) is sent to the Direct Memory Access (DMA) driver code, which performs the DMA operations.

For any packet that is smaller than minimum payload size, the rest of the bytes that make the minimum size are zeroed out prior to writing the data into the packet.

Only the DMA driver sets the count of bytes for each outgoing packet.

This functionality implements FDP_RIP.2.

7.4 Identification and Authentication

Administrative users (i.e., all users authorized to access the TOE's administrative interfaces) are identified by a user name and authenticated by an individual password associated with that user's account. This is true regardless of how the administrative user interfaces with the TOE. If the supplied user name and password match the user name and password pair maintained by the TOE, the administrative session is successfully established. Otherwise, the user receives an error and the session is not established. In addition, the TOE displays warning banners for interactive sessions as described in Section 7.7.

This functionality implements FIA_UIA_EXT.1, FIA_UAU_EXT.2.

For interactive user authentication at the web-based Configuration utility via HTTPS and the command line tmsh via SSH, BIG-IP obscures passwords entered by users.

This functionality implements FIA_UAU.7.

7.4.1 Password policy and user lockout

The TOE can enforce a password policy for all user accounts managed locally, other than those in the Administrator role. This includes the definition of a minimum password length and required character types (numeric, uppercase, lowercase, others). The minimum password length default value is 6; the valid range is from 6 to 255. This policy is enforced when users change their own passwords.

Other aspects of the authentication policy include the minimum and maximum lengths of time that passwords can be in effect, and the number of previous passwords that BIG-IP should store to prevent users from re-using former passwords.

- The minimum duration specifies the minimum number of days before which users cannot change their passwords; the default is 0 and the valid range is from 0 to 255.
- The maximum duration specifies the maximum number of days a password is valid; users must change their passwords before the maximum duration is reached, the default is 99999 days.
    - User accounts whose password has expired, based on the administrator-defined maximum password duration, are locked and require an administrator to reset them.
- Password memory specifies that the system records the specified number of passwords that the user has used in the past. Users cannot reuse a password that is in the list. The default is 0 and the valid range is from 0 to 127.

Both local and remote access to the TOE for individual users can be disabled ("locked") after a configured number of consecutive, failed authentication attempts on that user account. In the evaluated configuration, the default is 3 consecutive, failed authentication attempts with a valid range from 1 to 10. For each administrative interface (local and remote interfaces), a single centralized module in the TOE verifies user identification and authentication. That module returns authentication success or failure decisions and maintains the user lockout feature. A counter of failed authentication attempts is maintained for each user. If too many failed authentication attempts occur, the associated user account is locked out and access is denied. A counter is kept for each user to track consecutive authentication failures. When a successful authentication occurs, the counter is reset to zero.

In the evaluated configuration, an administrator-configured time value determines the duration of a user's lock out time. The default is 10 minutes (600 seconds).

In the evaluated configuration, it is not possible for all administrative users to be locked out of the TOE, because the primary administrative user account is permitted to login to the local console even if it is locked out when attempting to login through any remote interface.

This functionality implements FIA_AFL.1, FIA_PMG_EXT.1.

7.4.2 Certificate Validation

For TLS and HTTPS sessions, the TOE implements certificate validation using the OpenSSL crypto library.

The TOE supports validation of X.509 digital certificates using a certificate revocation list (CRL) as specified in [RFC5280] Section 5. Administrators create profiles which are used to define the parameters used to communicate with an external entity. These parameters include the ability to require the use of TLS and peer or mutual authentication and a definition of the certificate to use for authentication. This capability

is used to create a mutually authenticated connection with the external audit server. The external audit server provides a certificate to the TOE during establishment of the TLS connection in order to authenticate the external audit server.

The TOE offers administrative interfaces for creating a private key and certificate signing request (CSR). The CSR may include the following information: public key, common name, organization, organizational unit, country, locality, state / province, country, e-mail address, subject alternative name. After the CSR is created, the administrator must export the CSR outside the TOE. Outside the scope of the TOE, the administrator provides the CSR to the CA and then the CA returns the certificate to the administrator. Using the administrative interface, the administrator can then import the certificate into the TOE.

The only method supported by the TOE for obtaining a CA certificate is for the administrator to save a certificate to a text file and import it into the TOE. The certificates are stored in a text file. The TOE is capable of importing X.509v3 certificates and certificates in the PKCS12 format. The TOE is also capable of creating and using a self-signed certificate.

The TOE checks the validity of the certificates when the profile using the certificate is loaded and when the certificate is used by the TOE, including during authentication. If the certificates are modified, the digital signature verification would detect that the certificate had been tampered with and the certificate would be invalid. Administrators can ensure that the certificates presented have not been revoked by importing a certificate revocation list (CRL) into the TOE.

A certificate chain includes the root CA certificate, certificates of intermediate CAs, and the end entity certificate. The certificate chain consists of all the certificates necessary to validate the end certificate. Administrators can upload trusted device certificates (root CA certificates) into the TOE to identify which certificates are trusted. The TOE performs full certificate chain checking using Public Key Infrastructure X.509, verifies the expiration of the certificate (assuming a reliable time), and verifies its revocation using CRLs.

When the validity of a certificate cannot be established, the TOE will allow the administrator to choose whether or not to accept the certificate.

This implements FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3.

7.5 Security Function Management

The TOE provides the ability to administer the TOE both locally and remotely. Local administration is performed via the serial port console. Remote access to the management interfaces is only made available on the dedicated management network port of a BIG-IP system.

The TOE offers administrators four different methods to configure and manage the TSF. They are:

- Configuration Utility (Web-based GUI) - browser-based GUI interface with normal GUI panels and selections. The client browser talks to the Apache HTTP server over HTTPS; then the request passes through tomcat and to the BIG-IP.
- tmsh shell commands – provide a command line interface, accessible through an SSH client
- iControl API – SOAP-based programming interface over HTTPS.
- iControl REST API – REST-based programming interface over HTTPS.

The first three interfaces are independent. The tmsh interface is the most complete; though none of the three are proper subsets of each other. iControl REST APIs utilize tmsh shell command(s) to perform the desired operation, so it is basically a front-end to the tmsh shell commands. As such, the functions provided by the iControl REST API are a proper subset of the set of tmsh commands.

Remote use of these interfaces is performed over protected communication paths as described in Section 7.8. These four administrative interfaces require users to identify and authenticate themselves prior to performing any administrative functions.

The TOE comes with a pre-defined “admin” user with the Administrator role assigned that cannot be deleted. A password is assigned to the “admin” user during setup of the TOE. Local user accounts are managed by administrators in the Administrator or User Manager role and stored in the TOE's local user database. Management includes creating and deleting users, as well as changing another user's password (every user can change their own password), role, or partition the user has access to, and enabling or disabling terminal access for the user. However, User Managers that have access to only one partition cannot change the partition access of other users, and cannot change their own partition access or role. (More on roles can be found in Section 7.5.1.)

Some general configuration options include:

- definition of an administrative IP address for the TOE's management network interface
- configuration of remote logging
- configuration of auditing
- configuration of TOE security functions
- enable/disable services
- manage TSF data, configure the login access banner
- configure session inactivity timeout
- configure cryptographic functionality
- configure the RekeyLimit which defines how much data can be transmitted within an SSH connection before rekeying
- configuration of trusted updates
- configure firewall rules
- set the time period for rejecting logins from an Administrator who has reached the maximum number of unsuccessful authentication attempts, and
- set the time which is used for time stamps.

BIG-IP uses the concept of virtual servers to define destinations that BIG-IP accepts (client) traffic for. Virtual servers are represented by an IP address and service (such as HTTP). The actual resources that BIG-IP forwards the traffic to are referred to as nodes, represented by their IP address. Nodes can be grouped into pools, for example for the purpose of load balancing. (A client sends an HTTP request to BIG-IP's virtual server address, and BIG-IP will then select a node from the pool associated with the virtual server to forward the request to.) Virtual servers are a management tool used to simplify the configuration of filtering and processing incoming network requests.

In order to determine the treatment of different types of traffic, such as requiring client authentication or inspection of HTTP code at the application layer, administrators can assign profiles to virtual servers. Profiles contain detailed instructions on how the different traffic management-related security functions of the TOE are applied to matching traffic.

This functionality implements FMT_SMF.1.

7.5.1 Security Roles

Access of individual users to the web-based Configuration utility, tmsh, iControl API, and iControl REST API is restricted based on pre-defined roles. These roles define which type of objects a user has access to and which type of tasks he or she can perform. The role definitions cannot be changed by TOE administrators.

Table 10 contains the definition of user roles.

The TOE allows security administrators to define the type of terminal access that individual users have, i.e. whether they have access to the tmsh via SSH or not. The TOE can be administered either locally or remotely. Administering the TOE locally entails connecting a device to the management port on the BIG-IP via an Ethernet cable

The tasks that users can perform on objects, depending on their role, are grouped into hierarchical access levels:

- write: create, modify, enable and disable, and delete an object
- update: modify, enable, and disable an object
- enable/disable: enable and disable an object
- read: view an object

In addition to roles, the TOE implements the concept of partitions for restricting access to objects. Configuration objects that deal with the individual traffic management functions offered by the TOE are stored in partitions (either the Common partition, or administrator-defined partitions. Objects (including users, server pools, etc.) can be created in different partitions by administrators, and users can be assigned a partition they have access to ("partition access"). As a result, users will only have the type of access defined by their assigned role to objects in the partition that is defined by their partition access. (With certain exceptions documented in the tables below.) It is possible to assign a user access to "all" partitions, in which case the user will have access to objects in all partitions as defined by their role (referred to in the guidance documentation as "universal access").

Note: The fact that a user account is created in a specific partition does not mean that the user will automatically have access to other objects in that partition.

The TOE comes with a pre-defined "Common" partition, which cannot be deleted. New objects created by users are either placed in the user's partition, or - if the user has access to all partitions - are placed in the Common partition unless the user explicitly chooses otherwise. The pre-defined "admin" user with the Administrator role is located in the Common partition.

Even users who are located in a partition other than Common have certain access to objects in the Common partition, as follows:

- Administrator always has access to all objects defined in the TOE.
- User Managers have write access to user account objects in the Common partition, except those with the Administrator role assigned to them.
- Resource Administrators, Managers, Certificate Managers, Application Editors, Operators, and Guests have read access to all objects in the Common partition.

Role	Associated rights
Administrator	This role grants users complete access to all partitioned and non-partitioned objects on the system, manage remote user accounts and change their own passwords. This includes trusted updates and the management of all security functions and TSF data.
Resource Administrator	This role grants users complete access to all partitioned and non-partitioned objects on the system, except user account objects. Additionally, users with this role can change their own passwords. This includes management of all security functions and TSF data, including remote users, remote roles, but not other user management

Role	Associated rights
	functions.
User Manager	Users with the User Manager role that have access to all partitions can create, modify, delete, and view all user accounts except those that are assigned the Administrator role, or the User Manager role with different partition access. However, User Managers cannot manage user roles for remote user accounts. Users with the User Manager role that have access only to a single partition can create, modify, delete, and view only those user accounts that are in that partition and that have access to that partition only. User accounts with the User Manager role can change their own passwords.
Manager	This role grants users permission to create, modify, and delete virtual servers, nodes, pools, pool members, custom profiles, and custom monitors. Users in this role can view all objects on the system and change their own passwords.
Certificate Manager	This role grants users permission to manage device certificates and keys, as well as perform Federal Information Processing Standard (FIPS) operations.
Application Editor	This role grants users permission to modify nodes, pools, pool members, monitors and change their own passwords. These users can view all objects on the system.
Operator	This role grants users permission to enable or disable nodes and pool members. These users can view all objects.
Auditor	This role grants users permission to view all configuration data on the system, including logs and archives. Users with this role cannot create, modify, or delete any data, nor can they view TLS keys or user passwords.
Guest	This role grants users permission to view all objects on the system in their partition and Common partition.
No Access	This role prevents users from accessing the system.
Firewall Manager	This user has access only to the firewall software panel, and performs tasks associated only with security.

Table 10: BIG-IP User Roles

The Security Administrator role as defined in FMT_SMR.2 is considered to include each of the roles defined in Table 10, except for the Guest and No Access roles.

This functionality implements FMT_MOF.1/Services, FMT_MOF.1/ManualUpdate, FMT_MTD.1/CoreData, FMT_MTD.1/AdminAct, FMT_SMF.1, FMT_SMR.2.

7.6 Protection of the TSF

7.6.1 Protection of Sensitive Data

The TOE protects passwords used for the authentication of administrative users as follows:

- In storage for local user authentication, the TOE's administrative interfaces do not offer any interface to retrieve user passwords or configuration files.

- In transit between remote users and the TOE, the TOE implements SSH and TLS to protect the communication.

Pre-shared keys (such as credentials for remote servers), symmetric keys, and private keys are stored in the TOE's configuration files. The TOE does not offer an interface to retrieve the contents of its configuration files. Passwords are stored in a salted hashed format.

This functionality implements FPT_APW_EXT.1 and FPT_SKP_EXT.1.

7.6.2 Self-tests

The following self-tests are implemented by the TOE:

- The BIOS Power-On Self-Test POST test is only run at power on
- The OpenSSL integrity tests are run at power on and reboot (during OpenSSL initialization) for OpenSSL.
- The software integrity check (i.e., sys-icheck utility) is run at power on and reboot to check the integrity of the RPMs. This self-test can be run at any time.
- The cryptographic algorithm self-tests provided by OpenSSL are run at power on and.

The BIOS POST is a diagnostic program that checks the basic components required for the hardware to operate, tests the memory, and checks the disk controller, the attached disks, and the network controllers. The BIOS POST tests cannot be run on demand.

The fipscheck utility is a standard Open Source utility for verifying the integrity of OpenSSL during initialization.

The sys-icheck utility provides software integrity testing by comparing the current state of files in the system to a database created at install time and modified only through authorized system update mechanisms. When a discrepancy is detected, the utility reports that discrepancy. The utility can be run at any time during system operation, and will just report errors. However, once the system is placed into the Common Criteria configuration it is enabled to run at each boot, and will halt the boot if errors are found.

The TOE will execute self-tests at power-on to test the cryptographic algorithms and random number generation using known answer tests for each of the algorithms. If a power-on test fails, the boot process will halt.

The self-tests implemented by the TOE which are described in this section cover all aspects of the TSF are therefore and are sufficient for demonstrating that the TSF is operating correctly in the intended environment.

This functionality implements FPT_TST_EXT.1/PowerOn and FPT_TST_EXT.1/OnDemand.

7.6.3 Update Verification

While the evaluated configuration of the TOE is limited to the specific version and patch level of BIG-IP covered in this ST, the TOE nevertheless provides functionality that supports administrators in verifying the integrity and authenticity of updates provided by F5. The Configuration Utility or tmsh can be used to query the TOE version.

The TOE is able to validate digital signatures of updates provided by F5; F5 places the ISO files (updates) and signature files on their website. The administrative guidance instructs the customer to:

- Download the ISO and digital signature file
- Verify the ISO using that file

- Install the update

A signature file exists for each software change update provided by F5. The content of the signature file is a digital signature of a SHA256 digest of the ISO image file. The private and public keys are generated using the OpenSSL utility. The signing key is a 2048 bit RSA private key that is stored at F5 CM and only available for official F5 builds. The public key is included in the TMOS filesystem and is available on the F5 official site adjacent to the software archives. Note: The update verification implementation does not utilize certificates; only digital signatures.

The BIG-IP verifies the SHA256 hash of software archives, using 2048-bit RSA digital signature algorithm. If the signature is verified, the software update is installed. If the signature does not verify, the software update installation fails / aborts. The administrative guidance will instruct the customer to download the update again or contact F5 support.

This functionality implements FPT_TUD_EXT.1.

7.6.4 Time Source

The TOE provides reliable time stamps for its own use, in particular in audit records and other time-sensitive security functionality. The TOE is an appliance that includes a hardware-based clock and the TOE's operating system makes the real-time clock available through a mcpd-maintained time stamp. Administrators have the ability to set the hardware-based clock.

The security functions that rely on this time stamp in the evaluated configuration include:

- generation of audit records
- session locking for administrative users
- timeouts for remote sessions
- certificate validation / revocation

This functionality implements FPT_STM_EXT.1.

7.7 TOE Access

For interactive user authentication at the web-based Configuration utility via HTTPS and the command line tmsh via SSH or the serial port console, BIG-IP implements the display of administrator-defined banners to users.

This functionality implements FTA_TAB.1.

The TOE terminates remote administrative user (Configuration Utility or tmsh) sessions after an administrator-defined period of inactivity. Users can also actively terminate their sessions (log out).

This functionality implements FTA_SSL_EXT.1, FTA_SSL.3.

Lastly, administrators are able to actively terminate these sessions (i.e., to log out and therefore close an authenticated session).

This functionality implements FTA_SSL.4.

7.8 Trusted Path/Channels

The TOE acts as the TLS client when communicating with audit servers for the protection of audit records sent from the TOE to an external audit server. As described in Section 7.4.2, the TOE is configured to require a mutually authenticated connection with the external audit server. The external audit server provides a certificate to the TOE during establishment of the TLS connection in order to authenticate the external audit server.

This functionality implements FTP_ITC.1.

Network administrators connect to the TOE remotely via a dedicated network interface to administer the TOE. Administrators are authenticated locally by user name and password; remote authentication (via LDAP or AD) is not supported by the TOE. The TOE implements the following trusted paths, which are logically distinct from other communication paths and provide assured identification of both end points, as well as protecting the transmitted data from disclosure and providing detection of modification of the transmitted data:

- TLS Connections to the TOE via the web-based Configuration utility, iControl API and the iControl REST API are protected by TLS. TLS sessions are limited to TLS versions 1.1 and 1.2, using the cipher suites identified in FCS_TLSS_EXT.1[3]-[4].
- SSH Connections to the TOE's command line interface are protected using SSH version 2 as defined in FCS_SSHS_EXT.1. Additionally, the SSH implementation has hard-coded ecdh-sha2-nistp256 and ecdh-sha2-nistp384 key exchange; diffie-hellman-group1-sha1 key exchange is intentionally disabled.

This functionality implements FTP_TRP.1/Admin.

7.9 Firewall

7.9.1 Secure Initialization

The TOE initialization process is designed to ensure that security protections are in place before external users can access the TOE or request that it process data. The following is the initialization sequence:

- At power on (or system reset), the CPU begins executing system firmware. The details are specific to each platform, but will include either legacy BIOS or UEFI-compatible firmware. System firmware discovers and initializes the CPU(s), trains and configures memory controllers, trains system buses (such as QPI and PCIe) interconnecting hardware components, discovers and configures PCIe/PCI bridges, configures and locks down system management mode (SMM), initializes hardware such as storage and USB controllers that may be used during booting, selects a boot device, then loads the boot loader. Devices that are trained, such as those connected to DDR3, QPI or PCIe buses, must be functioning nominally to complete the process. A malfunctioning device that fails training may be silently skipped, in which case it will not be available to the system. Catastrophic failures cause initialization to fail completely, causing a system hang. The remaining devices are ignored. All systems provide a means to monitor the progress of the system firmware. This may include LEDs on the system main board, or a numerical code displayed on the VGA. If a failure causes a system hang, this code will reveal where in the process initialization stopped.
- The primary boot device holds the bootloader code. System firmware loads and executes the bootloader.
- The bootloader reads its configuration file, displays a menu on the VGA console (rarely connected), and over a terminal connected to the first serial communications port. If no user input is detected, the bootloader selects the default entry from its configuration file. This specifies the files containing the Linux kernel and initial ramdisk (initrd). Once loaded into memory, the bootloader launches the kernel.
- At this point, the root of the file system is the initrd in system memory. Linux no longer has access to the boot file system. Linux loads drivers from initrd that support the primary mass storage device. The primary file system is mounted read-only, checked for consistency, then write-enabled. A pivot root operation makes this the root file system and the initrd image in memory is released.

- The kernel launches /sbin/init (the “init” process). Init’s configuration is read from /etc/inittab
    - /etc/rc.d/rc.sysinit is executed. Scripts in /etc/sysconfig/sysinit directory are executed in order. In general, these scripts initialize hardware (additional disks and system hardware – but NOT Ethernet interfaces); the hardware is scanned and attached.
    - The increase_entropy daemon is started, which detects that the /dev/random entropy pool is running low, queries the cryptographic accelerator hardware for more entropy and stirs it into the pool.
- Linux enters runlevel 3.
    - Mcpd starts if the configuration is valid. It instantiates all data objects that process traffic (such as flow control rules and filters) from configuration data before entering the running state. This guarantees that those filters are in place before any traffic packets can be processed.
    - Management NIC is started
    - Audit daemon is started
    - OpenSSH started
    - Apache web server is started
    - OpenSSL initialized and established on configured services
    - If vCMP is licensed, provisioned, and configured, the vCMP daemon is started. This daemon is responsible for all guest operations, including deployment, and for establishing the internal connection to route traffic to the configured guests. Traffic will not flow across that connection until TMM itself is started.
- TMM is started; once in the running state, it begins processing data plane traffic.
    a. Traffic may now flow on the internal and external VLANs under the control of the information flow control filters already in place to process the traffic.
    b. If traffic that requires authentication contacts the TOE, and the TOE cannot connect to the remote authentication server (for example, if the external authentication server itself has not yet completed initialization), the traffic will be denied.

This functionality implements FFW_RUL_EXT.1.

7.9.1.1 Packet Filter / Stateful Firewall

When a network packet is received by the TOE, it is sent to the information flow control filters for processing before transmission. For the TCP, UDP, and ICMP protocols, the information flow control filters match network packets to existing network sessions if possible. For new network sessions, the network packets are compared against a set of network policy rules.

7.9.1.1.1 Rule-based Filtering

The TOE implements packet filtering functionality that, in the evaluated configuration, can be configured via the tmsh. The products guides contain guidance on how to create the firewall rules, but BIG-IP does not provide a specific pre-defined set of rules. The rules are created and configured by an administrator.

Administrator-defined rules are used to implement traffic filtering based on attributes including:

- source and destination IP addresses (per [RFC791] (IPv4) and [RFC2460] (IPv6))
- the transport layer protocol used (in particular, TCP or UDP)

- source and destination ports (per [RFC793] (TCP) and [RFC768] (UDP))
- ICMP message type and code (per [RFC792] (ICMPv4) and [RFC4443] (ICMPv6))
- distinct interface (external port or virtual port)
- ICMP source & destination address, type, and code

Rules can be associated with individual interfaces (VLANs, virtual IPs) or can be specified globally (i.e., they will be applied to all interfaces). Virtual IP addresses, together with a defined service (such as HTTP), are also referred to as virtual servers. They constitute BIG-IP's internal representation of traffic management objects that can be associated with certain handling and filtering instructions. In other words, virtual IPs can be used in traffic filtering rules to represent the destination address in network traffic packets that are subject to filtering.

In practice, the TOE allows administrators to specify rules for other attributes of IP-based traffic at the Internet and transport layers as well, without the evaluation making specific claims on this.

Rules will be matched in the order specified by administrators. Individual rules can either lead to a denial of the traffic, or permission of the session. In addition, administrators can specify (per rule) that a log entry will be created when network packets match the rule.

For the TCP, UDP, ICMP protocols, the TOE's rule enforcement considers the state of a network session when deciding whether to forward a network packet or deny it. The TOE maintains a session database for TCP, UDP, and ICMP sessions. For example, if network packets during session establishment were permitted based on existing rulesets, then subsequent packets for the same session (matching source and destination IP addresses and ports, sequence numbers, and flags) will be permitted without further evaluation, as long as the session is still active (i.e., matches an entry in TMM's state table). UDP packets that match source and destination addresses and port of a previously permitted packet will be accepted within the limits of defined time-outs. ICMP network packets that match the source and destination addresses, type, and code of a previously permitted packet will be accepted within the limits of defined time-outs. Administrators can define time-outs for inactive connections by setting the corresponding configuration parameters. If a connection does not show activity past that amount of time, it will be dropped. Network packets that do not match the security attributes for the existing TCP, UDP, or ICMP sessions as described above will be subject to the firewall rules associated with the applicable interface.

For all protocols, when a session is terminated by either end of the communication, the session information is removed from the session database. In addition, all sessions have a timeout value after which, the session information is removed from the session database. Timed out sessions, must establish a new session and be processed by the information flow control filters again in order to continue transmitting. The session removal is effectively immediately, once the session information is removed from the session database.

TCP SYN flood attack mitigation, which limits the number of simultaneous TCP SYN queue entries, is implemented using TCP SYN cookies. When a predetermined, configurable threshold is reached for new or untrusted connections, the SYNCheck feature is activated, initiating the use of TCP SYN cookies for subsequent TCP connections resulting in no connection being created on the initial TCP SYN packet. BIG-IP discards the TCP SYN queue entry after sending a SYN-ACK to alleviate the SYN queue. If the client is legitimate and responds with an ACK, the "cookie" value that is sent to the client allows BIG-IP to reconstruct the SYN and therefore continue to establish the TCP session. Since the SYN queue does not fill once the threshold has been reached this effectively prevents TCP SYN ACK denial of service. BIG-IP does not have a concept of embryonic (“half-open”) TCP sessions.

SYN cookies are implemented by the TOE's SYN Check feature. Administrators can specify a threshold (in absolute numbers) of active TCP connections for the system. Once this threshold is reached, the TOE will start using SYN cookies for TCP connection requests, i.e., use a proprietary algorithm to calculate individual sequence numbers for use in SYN/ACK responses to clients, instead of storing the connection

requests in memory. If an ACK response is received, the TOE will calculate the original sequence value, and whether it matches the sequence number included in the SYN/ACK response. Only if this is the case, and the response has not exceeded a specific time limit, the connection will be accepted.

The TOE is also capable of generating dynamic rule sets for the FTP protocol which requires more than one connection. For example, if an FTP control session is established based on an administrator-defined rule that permits that traffic, the TOE will create and enforce a dynamic rule that permits traffic matching the data connection defined in that control session as specified in [RFC959]. The filtering rules can be configured to audit the outcome of the enforcement of dynamic rules for FTP.

Administrators can further refine the traffic filtering behavior of the TSF as follows:

1) Administrators can specify that ARP packets are always accepted.
2) Administrators can define types of ICMP packets that are always accepted.
3) Administrators can specify that traffic originating from certain MAC addresses, IP addresses, or VLANs is always accepted.
4) Administrators can specify packet evaluation rules using keywords

Network packets that do not match an explicit accept rule are denied. If any component fails during the processing of a network packet, that packet is denied.

This functionality implements FFW_RUL_EXT.1 and FFW_RUL_EXT.2.

7.9.1.1.2 Static Filtering

In addition, network packets with certain attributes (that typically represent malicious traffic and have no common application in other contexts) are rejected by the TOE regardless of administrator-defined rules. Administrators are able to configure whether log entries are created for these conditions.

This includes:

- packets which are invalid fragments (invalid fragments for each protocol are as defined by the applicable RFC for that protocol)
- fragmented IP packets which cannot be re-assembled completely
- packets where the source address of the network packet is defined as being on a broadcast network
- packets where the source address of the network packet is defined as being on a multicast network
- packets where the source address of the network packet is defined as being a loopback address
- 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 [RFC5735] for IPv4
- 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 [RFC3513] for IPv6
- packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified
- packets where the source address of the network packet is equal to the address of the network interface where the network packet was received
- packets where the source or destination address of the network packet is a link-local address
- 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 as determined by the IP address

assigned to the network interface on which the packet was received. The IP address of the network packet must be in the subnet of the IP address assigned to the network interface on which the packet was received.

7.9.1.1.3 RFC Conformance

Interoperability and compliance testing were performed on the protocols as specified in the following RFCs.

- RFC 792 (ICMPv4)
- RFC 4443 (ICMPv6)
- RFC 791 (IPv4)
- RFC 2460 (IPv6)
- RFC 793 (TCP)
- RFC 768 (UDP)

The following methods were used to perform interoperability and compliance testing.

- F5-written test cases for compliance and interoperability scenarios, covering UDP, ICMPv4 and ICMPv6, TCP, HTTP, HTTPS, SSH, SSL, and TLS
- Ixia's IxANVL (Automated Network Validation Library) Protocol Compliance Test Suite, covering TCP
- TAHI Project's Test and Verification for IPv6 Test Suite, covering IPv6 and ICMPv6

