F5 BIG-IP® 17.5.0 including APM Security Target

Document Number: CC2024-ASE_ST-002

Document Version: 17.63

Date: March 5, 2026

Prepared By:

Saffire Systems

Carmel, IN 46032

Prepared For:

F5, 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 17.5.0 including APM Security Target
Version:	17.63
Publication Date:	March 5, 2026
Sponsor:	F5, Inc.
Developer:	F5, Inc.
ST Author:	Michelle Ruppel, Saffire Systems

1.2 TOE Identification

The TOE claiming conformance to this ST is identified as *BIG-IP Version 17.5.0 including APM* (Build Hotfix-BIGIP-17.5.0.0.189.15-ENG, also referred to as 17.5) running on any of the devices identified in Section 1.2.1 and any of the hypervisors identified in Section 1.2.2.

BIG-IP runs in three different environments:

- A physical Network Device (pND) running directly on the iSeries or VIPRION hardware
- A virtual network device (vND) running as a tenant on F5OS running on rSeries or VELOS hardware
- A vND running in a virtual machine on a third-party hypervisor, such as VMWare, Hyper-V, and KVM

The TOE is deployed with the following license modes/modules:

- Application Delivery Controller deployment
    - Appliance Mode
    - Traffic Management Operating System (TMOS) modules
    - Traffic Management Microkernel (TMM) module
    - Access Policy Manager (APM) module
    - Local Traffic Manager (LTM) module

1.2.1 F5 Devices

BIG-IP runs in two different environments on F5 devices. BIG-IP operates as a physical Network Device (pND) running directly on the iSeries or VIPRION hardware. BIG-IP operates as a virtual Network Device (vND) running as a tenant on F5OS on the rSeries or VELOS hardware.

The following provides an explanation of table columns in the F5 devices tables 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-APM-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 "XXB" in the SKUs below denotes that the SKU is applicable to a range of platforms or models. "XXB" is part of the actual SKU and not a placeholder.

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.

1.2.1.1 Physical Network Devices

When BIG-IP is running directly on the iSeries or VIPRION hardware, the TOE consists of any of the hardware appliances listed in Table 1 installed with LTM+APM with appliance mode software.

SKU	Part #	Model Series
iSeries		
F5-BIG-LTM-I15600 F5-ADD-BIG-APMI156XXB F5-ADD-BIG-MODE	500-0001-08	i15000
F5-BIG-LTM-I15800 F5-ADD-BIGAPMI158XXB F5-ADD-BIG-MODE	500-0001-08	i15000
F5-BIG-LTM-I15820-DF F5-ADD-BIGAPMI158XXB F5-ADD-BIG-MODE	500-0043-00	I15000-DF
VIPRION		
F5-VPR-LTM-C2400-AC F5-VPR-LTM-B2250 F5-ADD-VPRAPM-C2400B F5-ADD-BIG-MODE	400-0028-11 400-0039-03	C2400 B2250

SKU	Part #	Model Series
F5-VPR-LTM-C4480-AC F5-VPR-LTM-B4450 F5-ADD-VPRAPM-C4400B F5-ADD-BIG-MODE	400-0033-04 400-0053-10	C4480 B4450

Table 1: Supported Physical Network Devices

Each of the hardware platforms includes a third party proprietary cryptographic acceleration card. All iSeries hardware platforms and the VIPRION B4450 include the Intel Coleto Creek (8955). The VIPRION B2250 model includes the Cavium Nitrox (CN3540-500-C20).

1.2.1.2 Virtual Network Devices

When BIG-IP is running as a tenant on F5OS running on rSeries or VELOS hardware, the TOE runs on any of the hardware appliances listed in Table 2 installed with LTM+APM with appliance mode software.

SKU	Part #	Model Series
rSeries		
F5-BIG-LTM-R4600 F5-ADD-BIG-APMR4xxxB F5-ADD-BIG-MODE	200-0417-03	R4000
F5-BIG-LTM-R4800 F5-ADD-BIG-APMR4xxxB F5-ADD-BIG-MODE	200-0417-03	R4000
F5-BIG-LTM-R5600 F5-ADD-BIG-APMR5xxxB F5-ADD-BIG-MODE	200-0411-05	R5000
F5-BIG-LTM-R5800 F5-ADD-BIG-APMR5xxxB F5-ADD-BIG-MODE	200-0411-05	R5000
F5-BIG-LTM-R5900 F5-ADD-BIG-APMR5xxxB F5-ADD-BIG-MODE	200-0411-05	R5000

SKU	**Part #**	**Model Series**
F5-BIG-LTM-R5920-DF F5-ADD-BIG-APMR5xxxB F5-ADD-BIG-MODE	200-0421-03	R5000
F5-BIG-LTM-R10600 F5-ADD-BIGAPMR106xxB F5-ADD-BIG-MODE	200-0413-05	R10000
F5-BIG-LTM-R10800 F5-ADD-BIGAPMR108xxB F5-ADD-BIG-MODE	200-0413-05	R10000
F5-BIG-LTM-R10900 F5-ADD-BIGAPMR109xxB F5-ADD-BIG-MODE	200-0413-05	R10000
F5-BIG-LTM-R10900-DF F5-ADD-BIGAPMR109xxB F5-ADD-BIG-MODE	200-0420-02	R10000
F5-BIG-APM-R12600DSB	200-0424-03	r12000
F5-BIG-APM-R12800DSB	200-0424-03	r12000
F5-BIG-APM-R12900DSB	200-0424-03	r12000
VELOS		
F5-VEL-LTM-BX110 F5-VEL-CX410-AC LIC-PKG-VEL-ADD-APM-LTD	400-0086-04 400-0096-00	BX110 CX410
F5-VEL-LTM-BX520 F5-VEL-CX410-AC LIC-PKG-VEL-ADD-APM-LTD	400-0093-03 400-0096-00	BX520 CX410

SKU	Part #	Model Series
F5-VEL-LTM-BX520 F5-VEL-CX1610-AC LIC-PKG-VEL-ADD-APM-LTD	400-0093-03 400-0091-05	BX520 CX1610

Table 2: Supported Virtual Network Devices

Each of the hardware platforms includes a third party proprietary cryptographic acceleration card. The rSeries and VELOS models include Intel QAT.

Note that BIG-IP on the rSeries and VELOS hardware platforms does not run directly on the hardware but on a platform layer – F5OS-A 1.5.3 for all rSeries models except the r12000, F5OS-A 1.8.0 for all of the rSeries models, and F5OS-C 1.8.1 for VELOS. The hardware appliances and platform layer are part of the Virtualization System (VS) included in the TOE environment.

The VELOS hardware platforms consist of one or two chassis controllers and one or more blades. The hypervisor runs on the chassis controller(s) and the tenants run on the blades.

1.2.2 Hypervisors

The TOE was also tested running on the hypervisors listed in Table 3.

Hypervisor	Platform	Processor
VMWare ESXi 8.0.3 (Build: 24414501)	Dell PowerEdge R650	Intel(R) Xeon(R) Gold 6330N CPU @ 2.20GHz
Hyper-V version 10.0.20348.1 on Windows Server 2022 Standard	Dell PowerEdge R450	Intel(R) Xeon(r) Silver 4309Y CPU @ 2.80GHz
KVM: qemu-system-x86 Version: 1:6.2+dfsg-2ubuntu6.6 on Ubuntu 22.04.2 LTS	Dell PowerEdge R450	Intel(R) Xeon(r) Silver 4309Y CPU @ 2.80GHz

Table 3: Supported Hypervisor Systems

The BIG-IP software is included in the TOE boundary. The hardware and hypervisor are part of the Virtualization System (VS) included in the TOE environment.

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 networking 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

APM	Access Policy Manager
CC	Common Criteria
CMI	Central Management Infrastructure
CRL	Certificate Revocation List
CRLDP	Certificate Revocation List Distribution Point
GUI	Graphical User Interface
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
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
VE	Virtual Edition

1.4 TOE Type

The TOE type is a Networking Device. When running on iSeries and VIPRION devices, the TOE is a physical Network Device. When running on hypervisors or on F5OS on rSeries or VELOS devices, the TOE is a virtual Network Device.

1.5 TOE Overview

The TOE is a standalone TOE consisting of a single component (i.e., it is not a distributed TOE). When running on iSeries and VIPRION devices, the TOE consists of the BIG-IP software and the iSeries or VIPRION hardware devices. When running on hypervisors or on F5OS on rSeries or VELOS devices, the TOE is represented by the virtual network device alone and does not include the virtualization system.

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 Controller, which includes the Local Traffic Manager (LTM) and Access Policy Manager (APM) modules, provides network traffic management capabilities.

BIG-IP products run on appliance or blade hardware, or hardware and platform layer, provided by F5 listed in Section 1.2.1 or on one of the hypervisors listed in Section 1.2.2. When running on a third-party hypervisor, there may be only one guest virtual machine running on the hypervisor and only one instance of BIG-IP for each hardware platform.

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.
- **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" or “TMUI”), 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.

1.6 TOE Description

1.6.1 Introduction

Typical BIG-IP network environments include an internal network, administrator network, external network, server pools, and redundant BIG-IP systems. In this typical example,

- Internet connections are mediated by BIG-IP to provide access to certain resources located in an organization's internal server pool, for example to a web-based e-commerce system presenting a storefront to consumers
- Users in the organization's Intranet 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 to administer the TOE
- The TOE is optionally set up in a redundant failover configuration, with heartbeat monitoring and reporting via a data link between the two instances

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.

The APM terminates TLS-based VPN connections from remote clients. Internal server resources are made available to these remote users by offering web-based access for remote users, forwarding certain application protocols (such as remote desktop protocol (RDP)), and providing transparent VPN tunneling. The APM subsystem relies upon the Active Directory and/or LDAP external authentication providers to provide authentication decisions; local authentication is not performed for APM.

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. The cryptographic functionality implemented in the TOE is provided by OpenSSL.

The TOE is divided into the following subsystems:

- F5 Device Hardware,
- F5 platform layer for rSeries and VELOS devices,

- Hardware for hypervisor deployments,
- Hypervisor for hypervisor deployments,
- Traffic Management Operating System (TMOS),
- Traffic Management Micro-kernel (TMM),
- Access Policy Manager (APM), and
- Local Traffic Manager (LTM).

F5's TMOS is a Linux-based operating system customized for performance and to execute on the TOE 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 APM module terminates TLS-based VPN connections from remote clients although these features are not included in the evaluated configuration.

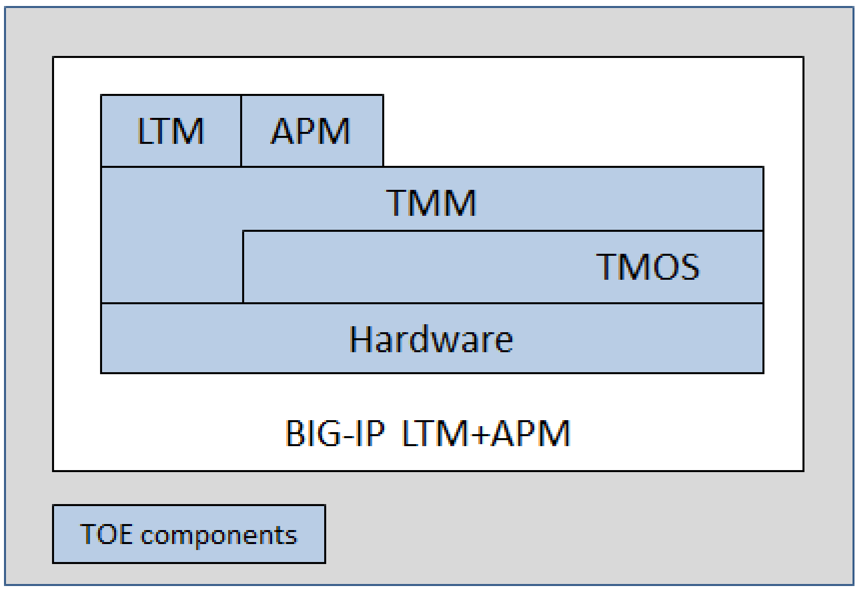

Figure 1: BIG-IP Subsystems for F5 iSeries and VIPRION Devices in Application Delivery Controller Deployments

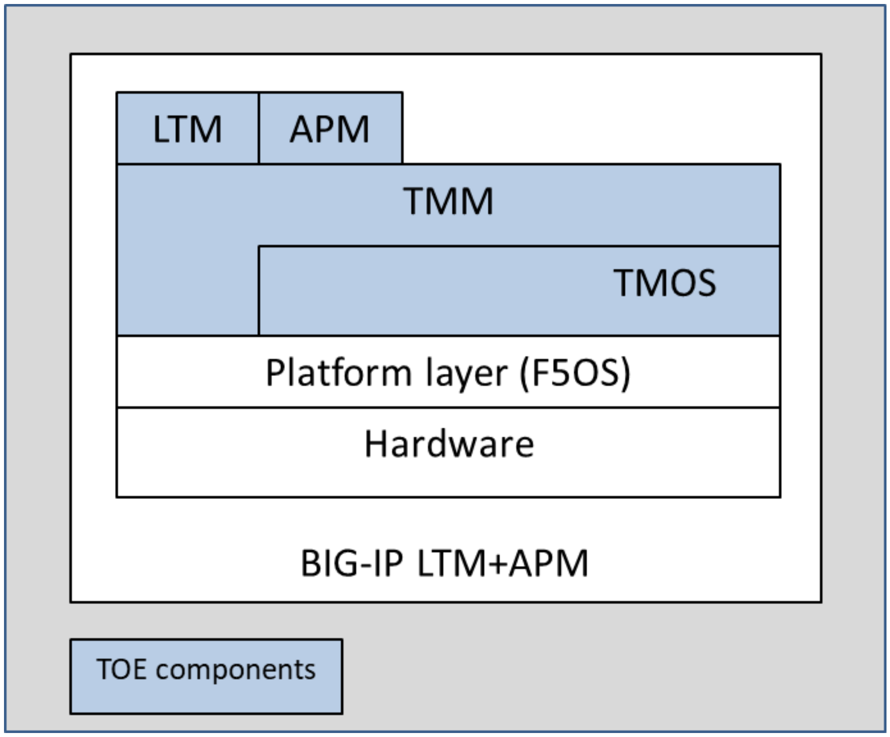

Figure 2: BIG-IP Subsystems for F5 rSeries and VELOS Devices in Application Delivery Controller Deployments

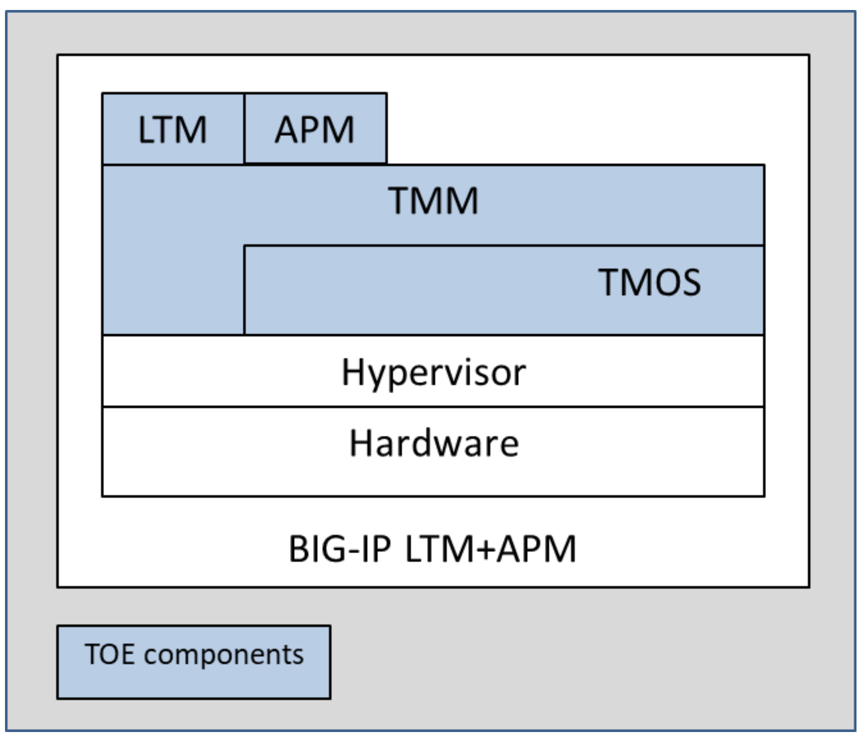

Figure 3: BIG-IP Subsystems for Hypervisors in Application Delivery Controller Deployments

TMOS is a Linux operating system that runs directly on device hardware, directly on the platform layer, or directly on the supported hypervisor. 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" or "TMUI"), 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 APM 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 or hypervisor. 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 or hypervisor provides more than one CPU, TMM runs multi-threaded (one thread per CPU). In this case, disaggregators in the kernel 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 the Application Delivery Controller Deployments are shown in Figure 4, Figure 5, and Figure 6. These plug-in modules include:

- Local Traffic Manager (LTM): authentication of HTTP (based on Apache) traffic and advanced traffic forwarding directives
- Access Policy Manager (APM): TLS-based client connectivity.

A diagram depicting aspects of the TOE's architecture and the boundaries of the TOE are provided in Figure 4, Figure 5, and Figure 6.

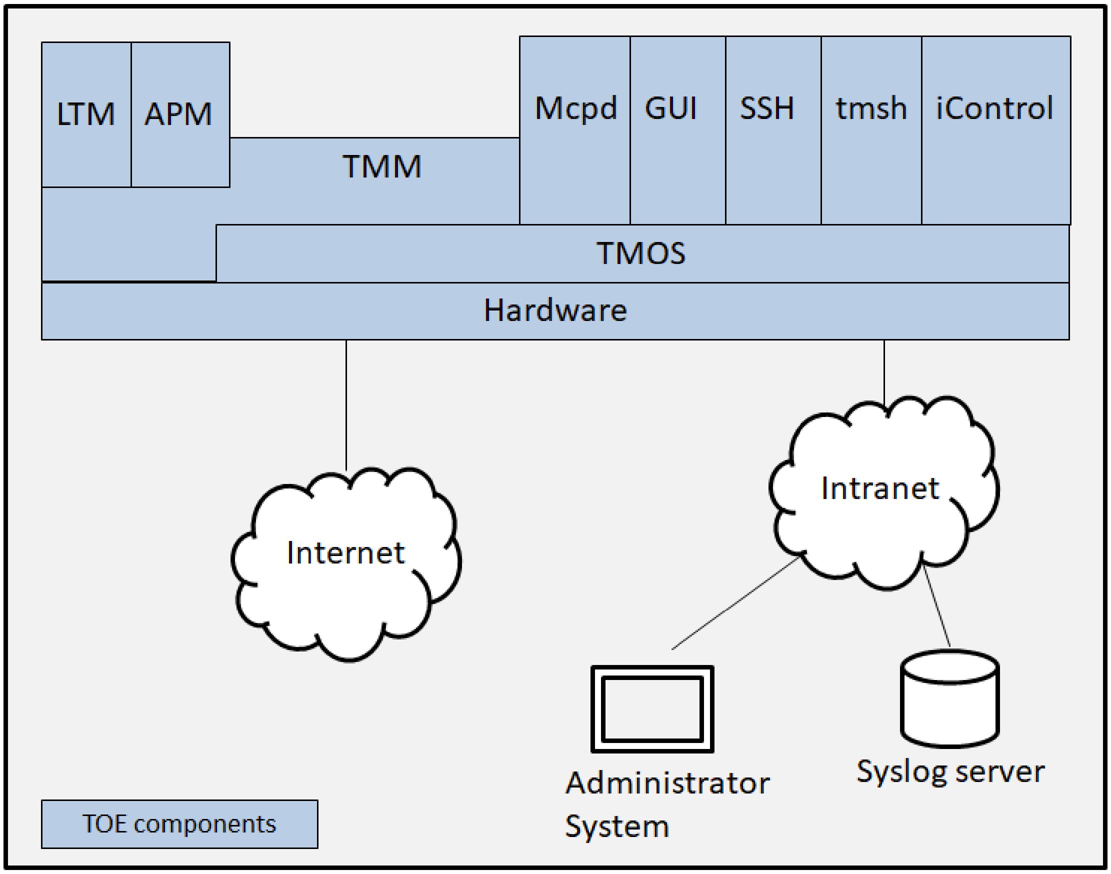

Figure 4: Architectural aspects of BIG-IP – F5 iSeries and VIPRION Devices in Application Delivery Controller Deployments

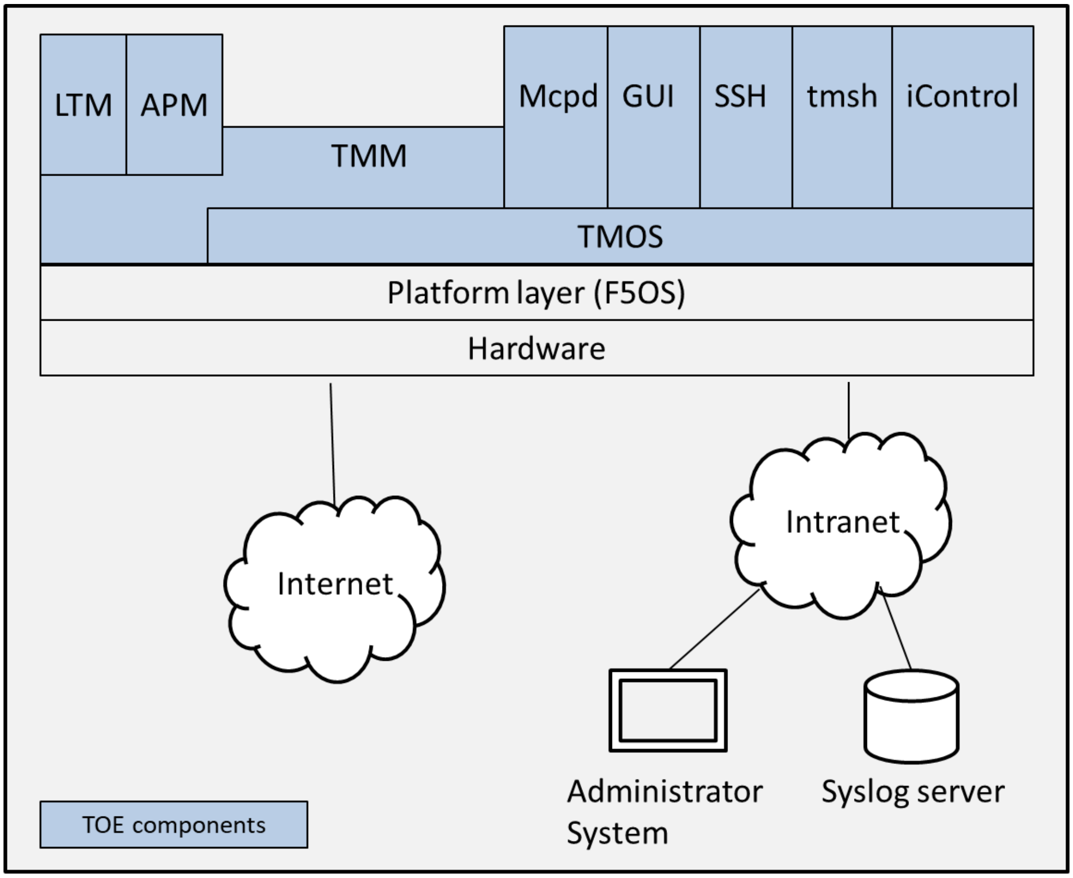

Figure 5: Architectural aspects of BIG-IP – F5 rSeries and VELOS Devices in Application Delivery Controller Deployments

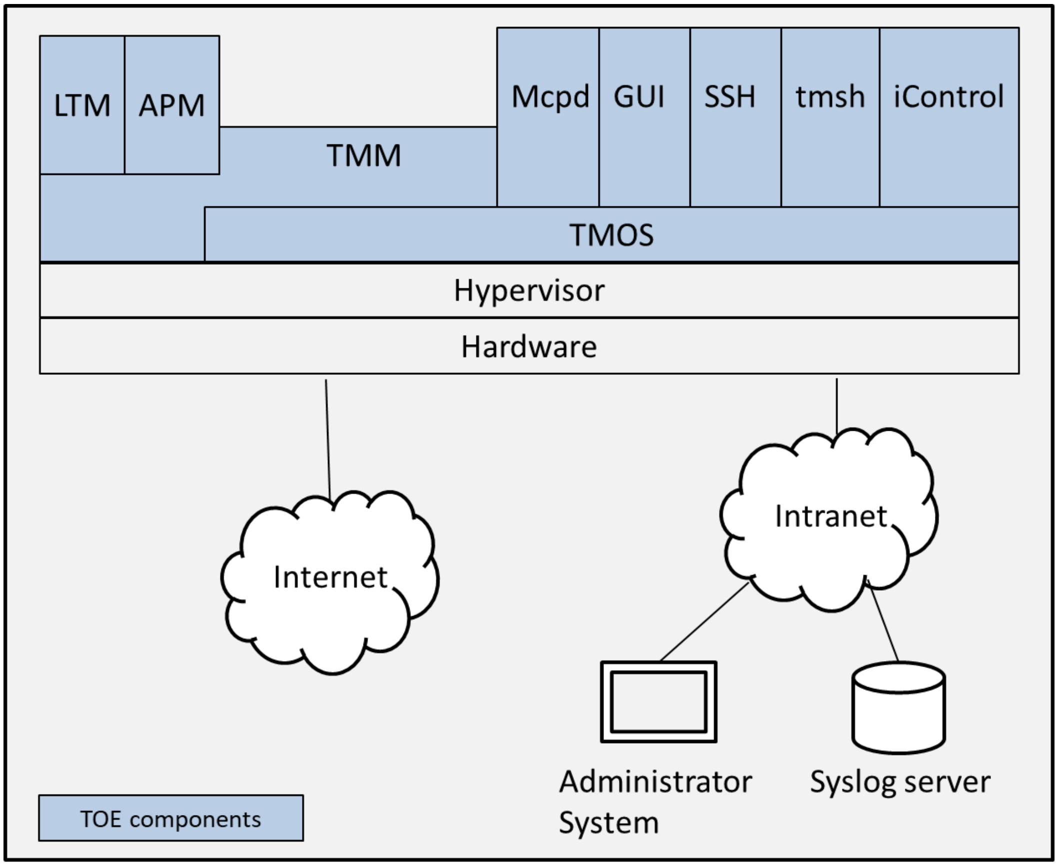

Figure 6: Architectural aspects of BIG-IP – Hypervisor in Application Delivery Controller Deployments

1.6.3 Physical Boundaries

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

1.6.3.1 Physical boundaries

BIG-IP version 17.5 is the software component of the TOE. When BIG-IP version 17.5 is running on one of the F5 iSeries and VIPRION devices identified in Section 1.2.1.1, the TOE is a physical Network Device and includes hardware and software physical components as identified in Section 1.2.1.1.

When BIG-IP version 17.5 runs on one of the F5 rSeries and VELOS devices or on a hypervisor, the TOE is a virtual Network Device and includes only software. When BIG-IP version 17.5 runs on one of the F5 rSeries and VELOS devices identified in Section 1.2.1.2, the Virtualization System includes hardware and platform layer components as identified in Section 1.2.1.2. When BIG-IP version 17.5 runs on one of the hypervisors identified in Section 1.2.2, the Virtualization System includes hypervisor and hardware components as identified in Section 1.2.2.

The evaluated configuration of *BIG-IP Version 17.5.0 including APM* represents a licensing option with the following F5 modules present and operational:

- Application Delivery Controller deployment
    - Appliance Mode
    - Traffic Management Operating System (TMOS) modules

- Traffic Management Microkernel (TMM) module
- Access Policy Manager (APM) module
- Local Traffic Manager (LTM) module

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:

Certification-specific References
K000149875: Common Criteria Certification for BIG-IP 17.5.0
BIG-IP Common Criteria Evaluation Configuration Guide BIG-IP Release 17.5.0
General BIG-IP References
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 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 API Reference
iControl REST API User Guide
F5 Secure Vault: Administration
K08859735: Overview of the FTP profile (14.x – 17.x)
K12042624: Restricting access to the Configuration utility using client certificates (13.x – 16.x)
K13092: Overview of securing access to the BIG-IP system
K13123: Managing BIG-IP product hotfixes (11.x – 17.x)
K13302: Configuring the BIG-IP system to use an SSL chain certificate (11.x – 16.x)
K13454: Configuring SSH public key authentication on BIP-IP systems (11.x – 17.x)
K14620: Managing SSL Certificates for BIG-IP systems using the Configuration utility
K14783: Overview of the Client SSL profile (11.x – 17.x)
K14806: Overview of the Server SSL profile (11.x – 17.x)

K15462: Managing SSL certificates for BIG-IP systems using tmsh
K15497: Configuring a secure password policy for the BIG-IP system (11.x – 17.x)
K15664: Overview of BIG-IP device certificates (11.x – 16.x)
K42531434: Replacing the Configuration utility's self-signed SSL certificate with a CA-signed SSL certificate
K48615077: BIG-IP Daemons (15.x – 17.x)
K5532: Configuring the level of information logged for Traffic Management-related 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
K8021: Configuring the BIG-IP LTM system to allow outbound FTP sessions
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
K14120: Define advanced NTP configurations on the BIG-IP system (11.x - 17.x)
K34556595: Managing login failures for a local user account on the BIG-IP system
Hotfix-BIGIP-17.5.0.0.189.15-ENG.readme
Traffic Management Shell (tmsh) Reference Guide (versions 17.0.0 and 12.0.0[1])
Platform-specific References – Hardware
Platform Guide: i2000/i4000 Series
Platform Guide: i5000/i7000/i10000/i11000 Series
Platform Guide: i15000 Series
Platform Guide: r2000/r4000 Series
Platform Guide: r5000/r10000 Series
Platform Guide: VELOS CX Series
Platform Guide: VIPRION® 2200
Platform Guide: VIPRION® 4400 Series
Platform-specific References – VE
F5 BIG-IP Virtual Edition in Linux KVM
F5 BIG-IP Virtual Edition in Microsoft Hyper-V
F5 BIG-IP Virtual Edition in VMware ESXi
BIG-IP Virtual Edition Supported Platforms
K14810: Overview of BIG-IP VE license and throughput limits
K14946: Overview of BIG-IP VE image sizes
Update BIG-IP VE
KVM – BIG-IP Virtual Edition Users Guide

[1] The tmsh reference guide version 17.x zipfile contains the pages for each of the tmsh commands. The 12.0.0 pdf contains additional general information that is still valid in 17.x but not reproduced in the 17.x zipfile.

Linux KVM – BIG-IP VE Setup
Microsoft Hyper-V – BIG-IP Virtual Edition Setup
Microsoft Hyper-V – BIG-IP Virtual Edition Users Guide
Vmware ESXi – BIG-IP Virtual Edition Setup
Vmware ESXi – BIG-IP VE Users Guide

Table 4: Guidance Documentation

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
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

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. (applicable to F5 devices)
  - Remote (i.e., SSH) access to the Lights Out / Always On Management[2] capabilities of the system is disabled. (applicable to F5 devices)
  - TLS v1.1

[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.

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 older 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.

For TLS sessions, the TOE implements certificate validation using the OpenSSL crypto library. Time synchronization with an NTP server uses SHA-1 message digests to verify the integrity of the NTP packets.

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

For F5 devices, 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. For F5 devices, 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. For F5 devices, the TOE provides a total of four entropy sources. For hypervisors, the TOE provides a total of two 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
- Session keys for SSH and TLS sessions

1.6.4.3 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.

1.6.4.4 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.

1.6.4.5 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. Time is provided by a local real-time clock managed by either the Security Administrator setting the time or synchronizing with an NTP server.

1.6.4.6 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.

1.6.4.7 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.

1.6.4.7.1 Generic network traffic

The BIG-IP 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.

1.6.4.7.2 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.

1.6.4.7.3 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, AES-CBC-256, AES-CTR-128, AES-CTR-256, aes-128-gcm@openssh.com, and aes-256-gcm@openssh.com
- The SSH public-key authentication algorithms are limited to ecdsa-sha2-nistp256 and ecdsa-sha2-nistp384
- The transport data integrity algorithm is limited to HMAC-SHA2-256
- The SSH protocol key exchange mechanism is limited to ecdh-sha2-nistp256 and ecdh-sha2-nistp384.

1.6.4.7.4 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.

1.6.5 Delivery

1.6.5.1 F5 Devices

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, Inc. to manufacture, test, and deliver the BIG-IP product through an on-site assessment, process evaluation and F5, Inc. Supplier Approval Program.

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.

1.6.5.2 *Hypervisors*

The customer is responsible for acquiring and installing the hardware and hypervisor and ensuring that the hardware and hypervisor on which the BIG-IP software will run are free of tampering. Refer to the manufacturers of those components for details.

The BIG-IP software is available for customers to download from the F5 website by product version number. The software is provided in an image file format, and the image file protected via digital signature. Product ISOs and image files for product updates are also available on the F5 website.

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 claims exact conformance to the following Protection Profiles:

- collaborative Protection Profile for Network Devices (NDcPP), Version 3.0e, 06-December-2023

The ST is compliant with the following NDcPP technical decisions:

NIAP TD	Applicability
TD0990 - NIT Technical Decision: CTR_DRBG in FCS_RBG_EXT.1.2	Applicable
TD0923 - NIT Technical Decision: Auditable event for FAU_STG_EXT.1 in FAU_GEN.1.2	Applicable
TD0921 - NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction of Assignment	Applicable
TD0900 - NIT Technical Decision: Clarification to Local Administrator Access in FIA_UIA_EXT.1.3	Applicable
TD0899 - NIT Technical Decision: Correction of Renegotiation Test for TLS 1.2	Applicable
TD0886 - Clarification to FAU_STG_EXT.1 Test 6	Applicable.
TD0880 - NIT Decision: Removal of Duplicate Selection in FMT_SMF.1.1	Applicable.
TD0879 - NIT Decision: Correction of Chapter Headings in CPP_ND_V3.0E	Applicable.
TD0868 - NIT Decision: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8	Not applicable. The TOE does not claim FCS_IPSEC_EXT.1.
TD0836 - NIT Decision: Redundant Requirements in FPT_TST_EXT.1	Applicable.

The ST claims exact conformance to the following packages:

- Functional Package for Secure Shell (SSH) (PKG_SSH), Version 1.0, 2021-05-13

The ST is compliant with the following SSH technical decisions:

NIAP TD	Applicability
TD0967 - Allowance of Kex-strict in PKG_SSH_V1.0	Applicable
TD0909 - Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0	Applicable.
TD0777 - Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	Applicable.
TD0732 - FCS_SSHS_EXT.1.3 Test 2 Update	Applicable.
TD0695 - Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package	Applicable.
TD0682 - Addressing Ambiguity in FCS_SSHS_EXT.1 Tests	Applicable.

The ST was also evaluated against the individual evaluation activities

- Evaluation Activities for Network Device cPP, Version 3.0e, 06-December-2023
- Evaluation activities found in PKG_SSH.

2.3 Conformance Rationale

The ST claims exact conformance to the NDcPP V3.0e and PKG_SSH V1.0.

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 NDcPP. The threats, organizational security policies and assumptions are repeated here for the convenience of the reader. Refer to the NDcPP 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.

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

- Critical network traffic (administration traffic, authentication traffic, audit traffic, etc.) to/from the TOE
- 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 a Basic attack potential.

3.2 Threats

The threats identified in this section may be addressed by the TOE. 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 Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device 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 Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device 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 Network Devices that do not use standardized secure tunnelling 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 Network Device 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 Network Device 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 Network Device 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 device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the

Administrator or device credentials for use by the attacker. Threat agents may also be able to take advantage of weak administrative passwords to gain privileged access to the device.

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.

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 Administrators consent by accessing the TOE.

3.4 Assumptions

The assumptions defined below are made on the operational environment to ensure that the security functionality defined in this ST can be provided by the TOE.

A.PHYSICAL_PROTECTION

The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device 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 device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs.

A.LIMITED_FUNCTIONALITY

The 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 device should not provide a computing platform for general purpose applications (unrelated to networking functionality).

If a virtual TOE evaluated as a pND, following Case 2 vNDs as specified in Section 1.2, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of a distributed TOE run inside more than one virtual machine (VM) on a single VS. In Case 2 vND, no non-TOE guest VMs are allowed on the platform.

A.NO_THRU_TRAFFIC_PROTECTION

A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the NDcPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).

A.TRUSTED_ADMINISTRATOR

The Security Administrator(s) for the Network 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 device. The Network 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.

For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).

A.REGULAR_UPDATES

The Network 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 Network 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 networking equipment when the equipment is discarded or removed from its operational environment.

A.VS_TRUSTED_ADMINISTRATOR (applies to vNDs only)

The Security Administrators for the VS are assumed to be trusted and to act in the best interest of security for the organization. This includes not interfering with the correct operation of the device. The Network Device is not expected to be capable of defending against a malicious VS Administrator that actively works to bypass or compromise the security of the device.

A.VS_REGULAR_UPDATES (applies to vNDs only)

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

A.VS_ISOLATON (applies to vNDs only)

For vNDs, it is assumed that the VS provides, and is configured to provide sufficient isolation between software running in VMs on the same physical platform. Furthermore, it is assumed that the VS adequately protects itself from software running inside VMs on the same physical platform.

A.VS_CORRECT_CONFIGURATION (applies to vNDs only)

For vNDs, it is assumed that the VS and VMs are correctly configured to support ND functionality implemented in VMs.

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. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS.

OE.NO_THRU_TRAFFIC_PROTECTION

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

OE.TRUSTED_ADMIN

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

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

OE.UPDATES

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

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

OE.VM_CONFIGURATION (applies to vNDs only)

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

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

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

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

5 Extended Components Definition

The extended components used in this ST are taken from the NDcPP and/or PKG_SSH.

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

Security Audit (FAU)

- FAU_STG_EXT.1
- FAU_STG_EXT.3

Cryptographic Support (FCS)

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

Identification and Authentication (FIA)

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

Protection of the TSF (FPT)

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

TOE Access (FTA)

- FTA_SSL_EXT.1

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

Cryptographic Support (FCS)

- FCS_SSH_EXT.1
- FCS_SSHS_EXT.1

6 Security Requirements

The security requirements that are levied on the TOE are specified in this section of the ST. If the security requirement includes "_EXT" at the end of the security requirement name, it is an extended security requirement; otherwise it is taken from CC Part 2. The table below identifies the source of each claimed security functional requirement (SFR). Each of the security requirements is drawn from NDcPP or PKG_SSH .

TOE Security Functional Requirements		Source
FAU_GEN.1	Audit Data Generation	NDcPP
FAU_GEN.2	User Identity Association	NDcPP
FAU_STG.1	Protected Audit Trail Storage	NDcPP
FAU_STG_EXT.1	Protected Audit Event Storage	NDcPP
FAU_STG_EXT.3	Action in case of Possible Audit Data Loss	NDcPP
FCS_CKM.1	Cryptographic Key Generation	NDcPP
FCS_CKM.2	Cryptographic Key Establishment	NDcPP
FCS_CKM.4	Cryptographic Key Destruction	NDcPP
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)	NDcPP
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)	NDcPP
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)	NDcPP
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)	NDcPP
FCS_HTTPS_EXT.1	HTTPS Protocol	NDcPP
FCS_NTP_EXT.1	NTP Protocol	NDcPP
FCS_RBG_EXT.1	Random Bit Generation	NDcPP
FCS_SSH_EXT.1	SSH Protocol	PKG_SSH
FCS_SSHS_EXT.1	SSH Protocol - Server	PKG_SSH
FCS_TLSC_EXT.1	TLS Client Protocol	NDcPP
FCS_TLSC_EXT.2	TLS Client Support for Mutual Authentication	NDcPP
FCS_TLSS_EXT.1[1]-[2]	TLS Server Protocol	NDcPP
FIA_AFL.1	Authentication Failure Management	NDcPP
FIA_PMG_EXT.1	Password Management	NDcPP
FIA_UAU.7	Protected Authentication Feedback	NDcPP
FIA_UIA_EXT.1	User Identification and Authentication	NDcPP
FIA_X509_EXT.1/Rev	X.509 Certificate Validation	NDcPP
FIA_X509_EXT.2	X.509 Certificate Authentication	NDcPP
FIA_X509_EXT.3	X.509 Certificate Requests	NDcPP
FMT_MOF.1/ManualUpdate	Management of Security Functions Behavior/ManualUpdate	NDcPP
FMT_MOF.1/Services	Management of Security Functions Behavior/Services	NDcPP
FMT_MTD.1/CoreData	Management of TSF Data/CoreData	NDcPP
FMT_MTD.1/CryptoKeys	Management of TSF Data/CryptoKeys	NDcPP
FMT_SMF.1	Specification of Management Functions	NDcPP
FMT_SMR.2	Restrictions on Security Roles	NDcPP
FPT_APW_EXT.1	Protection of Administrator Passwords	NDcPP
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)	NDcPP
FPT_STM_EXT.1	Reliable Time Stamps	NDcPP
FPT_TST_EXT.1	TSF Testing	NDcPP

FPT_TUD_EXT.1	Trusted Update	NDcPP
FTA_SSL.3	TSF-initiated Termination	NDcPP
FTA_SSL.4	User-initiated Termination	NDcPP
FTA_SSL_EXT.1	TSF-initiated Session Locking	NDcPP
FTA_TAB.1	Default TOE Access Banners	NDcPP
FTP_ITC.1	Inter-TSF Trusted Channel	NDcPP
FTP_TRP.1/Admin	Trusted Path	NDcPP

Table 5: Security Functional Requirements

6.1 Conventions

The CC defines four operations on security functional requirements. The SFRs claimed in this ST have been drawn from the NDcPP or PKG_SSH. The CC operations already performed in the NDcPP or in the PKG_SSH are reproduced in plain text and not denoted in this ST. Instead, the requirements have been copied from the NDcPP/PKG_SSH and any remaining operations have been completed herein. The NDcPP made refinements and SFR operations defined in the Common Criteria (CC) and the PPs should be consulted to identify those operations. The conventions below define the denotations used in this ST to identify the operations completed in this ST by the ST author.

Assignment made in ST: indicated with *italics text*

<u>Selection made in ST:</u> indicated with <u>underlined text</u>

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

deletions indicated with ~~strikethrough text~~

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 - Mandatory

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 Administrator account shall be logged if individual 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 Administrator account shall be logged),
- *starting and stopping services*;

d) Specifically defined auditable events listed in **Table 6**.

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 6**.

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	Configuration of local audit settings.	Identity of account making changes to the audit configuration.
FAU_STG_EXT.3	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/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session	Reason for failure.
FCS_NTP_EXT.1	• Configuration of a new time server • Removal of configured time server.	Identity if new/removed time server
FCS_RBG_EXT.1	None.	None.
FCS_SSH_EXT.1	Failure to establish SSH connection	Reason for failure and Non-TOE endpoint of attempted connection (IP Address) Application Note: TD0777 applies to this Auditable Event definition.
FCS_SSH_EXT.1	Establishment of SSH connection,	Non-TOE endpoint of connection (IP Address)
FCS_SSH_EXT.1	Termination of SSH connection	Non-TOE endpoint of connection (IP Address)
FCS_SSH_EXT.1	Dropping of packet(s) outside defined size limits	Packet size
FCS_SSHS_EXT.1	None.	None.
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_TLSC_EXT.2	None.	None.
FCS_TLSS_EXT.1[1]-[2]	Failure to establish a TLS Session	Reason for failure.
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_UAU.7	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_X509_EXT.1/Rev	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store.	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MOF.1/Services	None.	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	None	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_SKP_EXT.1	None.	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 NDcPP**.)	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).
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	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_SSL_EXT.1 (if "terminate the session" is selected)	The termination of a local session by the session lock.	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	• None • None • Reason for failure
FTP_TRP.1/Admin	• Initiation of the trusted path. • Termination of the trusted path. • Failure of the trusted path functions.	• None • None • Reason for failure

Table 6: Security Functional Requirements and Auditable Events

6.2.1.2 FAU_GEN.2 User Identity Association - Mandatory

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 - Optional

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 - Mandatory

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. In addition

- The TOE shall consist of a single standalone component that stores audit data locally.

FAU_STG_EXT.1.3 The TSF shall maintain a log file of audit records in the event that an interruption of communication with the remote audit server occurs.

FAU_STG_EXT.1.4 The TSF shall be able to store persistent audit records locally with a minimum storage size of *7 GB of audit storage*.

FAU_STG_EXT.1.5 The TSF shall overwrite previous audit records according to the following rule: *log files are numbered and the older log file(s) is deleted* when the local storage space for audit data is full.

FAU_STG_EXT.1.6 The TSF shall provide the following mechanisms for administrative access to locally stored audit records ability to view locally.

6.2.1.5 FAU_STG_EXT.3 Action in case of possible audit data loss - Optional

FAU_STG_EXT.3.1 The TSF shall generate a warning to inform the Administrator before the audit trail exceeds the local audit trail storage capacity.

6.2.2 Cryptographic Support (FCS)

6.2.2.1 FCS_CKM.1 Cryptographic Key Generation - Mandatory

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, 3072, 4096 bits* that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1;
- ECC schemes using 'NIST curves' P-256, P-384 that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4, or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2, or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.;
- FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and RFC 7919.

ST Application Note: TD0921 applies to the SFR definition.

6.2.2.2 FCS_CKM.2 Cryptographic Key Establishment - Mandatory

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

- RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.2";
- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";
- FFC Schemes using "safe-prime" groups that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and groups listed in RFC 7919.

6.2.2.3 FCS_CKM.4 Cryptographic Key Destruction - Mandatory

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 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 the EEPROM on F5 Devices**, logically addresses the storage location of the key and performs a single-pass overwrite consisting of *random data*;

- instructs a part of the TSF to destroy the abstraction that represents the key

that meets the following: No Standard.

6.2.2.4 FCS_COP.1/DataEncryption Cryptographic operation (AES Data Encryption/Decryption) - Mandatory

FCS_COP.1.1/DataEncryption The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in CBC, CTR, 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, CTR 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) - Mandatory

FCS_COP.1.1/SigGen The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm

- RSA Digital Signature Algorithm,
- Elliptic Curve Digital Signature Algorithm

and cryptographic key sizes

- For RSA: *modulus 2048 bits or greater,*
- For ECDSA: *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 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 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 implementing P-256, P-384 curves that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST Recommended curves”; or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves; or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.

ST Application Note: TD0921 applies to the SFR definition.

6.2.2.6 FCS_COP.1/Hash Cryptographic operation (Hash Algorithm) - Mandatory

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 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) - Mandatory

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 - Selection

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.

6.2.2.9 FCS_NTP_EXT.1 NTP Protocol - Selection

FCS_NTP_EXT.1.1 The TSF shall use only the following NTP version(s) NTP v4 (RFC 5905).

FCS_NTP_EXT.1.2 The TSF shall update its system time using

• Authentication using SHA1 as the message digest algorithm(s).

FCS_NTP_EXT.1.3 The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

FCS_NTP_EXT.1.4 The TSF shall support configuration of at least three (3) NTP time sources in the Operational Environment.

6.2.2.10 FCS_RBG_EXT.1 Random Bit Generation - Mandatory

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* platform-based noise source **for F5 devices except 10000 series,** *one* platform-based noise source **for F5 10000 series devices** 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".

ST Application Note: TD0990 applies to the SFR definition.

6.2.2.11 FCS_SSH_EXT.1 SSH Protocol – Selection

FCS_SSH_EXT.1.1 The TOE shall implement SSH acting as a server in accordance with that complies with RFCs 4251, 4252, 4253, 4254, 5656, 6668, 8332 and no other standard.

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods:

- "password" (RFC 4252),
- "publickey" (RFC 4252):
  - ecdsa-sha2-nistp256 (RFC 5656),
  - ecdsa-sha2-nistp384 (RFC 5656),

and no other methods.

FCS_SSH_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_SSH_EXT.1.4 The TSF shall protect data in transit from unauthorised disclosure using the following mechanisms:

aes128-ctr (RFC 4344),
aes256-ctr (RFC 4344),
aes128-cbc (RFC 4253),
aes256-cbc (RFC 4253),
aes128-gcm@openssh.com (RFC 5647),
aes256-gcm@openssh.com (RFC 5647)
and no other mechanisms.

FCS_SSH_EXT.1.5 The TSF shall protect data in transit from modification, deletion, and insertion using:

- hmac-sha2-256 (RFC 6668),

and no other mechanisms.

FCS_SSH_EXT.1.6 The TSF shall establish a shared secret with its peer using:

- ecdh-sha2-nistp256 (RFC 5656),
- ecdh-sha2-nistp384 (RFC 5656),

and no other mechanisms.

FCS_SSH_EXT.1.7 The TSF shall use SSH KDF as defined in

- RFC 5656 (Section 4)

to derive the following cryptographic keys from a shared secret: session keys.

FCS_SSH_EXT.1.8 The TSF shall ensure that

- a rekey of the session keys,

occurs when any of the following thresholds are met: one hour connection time no more than one gigabyte of transmitted data, or no more than one gigabyte of received data.

ST Application Note: TD0909 applies to the SFR definition.

6.2.2.12 FCS_SSHS_EXT.1 SSH Protocol - Server – Selection

FCS_SSHS_EXT.1.1 The TSF shall authenticate itself to its peer (SSH Client) using:

- ecdsa-sha2-nistp256 (RFC 5656),
- ecdsa-sha2-nistp384 (RFC 5656).

6.2.2.13 FCS_TLSC_EXT.1 TLS Client Protocol - Selection

FCS_TLSC_EXT.1.1 The **data plane of the** TSF shall implement TLS 1.3 (RFC 8446), TLS 1.2 (RFC 5246) supporting the following ciphersuites:

- Supported ciphersuites for TLS 1.2 from List 1:
  - 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 8422

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 8422
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 8422
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 8422
- 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

- Select supported ciphersuites for TLS 1.3 from List 2:
  - TLS_AES_128_GCM_SHA256
  - TLS_AES_256_GCM_SHA384

and no other ciphersuites.

FCS_TLSC_EXT.1.2 The **data plane of the** TSF shall verify that the presented identifier matches IPv4 address in the CN or in the SAN.

FCS_TLSC_EXT.1.3 The **data plane of the** TSF shall not establish a trusted channel if the server certificate is invalid

- without any administrator override mechanism.

FCS_TLSC_EXT.1.4 The **data plane of the** TSF shall present the Supported Groups Extension with the following curves/groups: secp256r1, secp384r1, ffdhe2048, ffdhe3072, ffdhe4096 and no other curves/groups in the Client Hello.

FCS_TLSC_EXT.1.5 The **data plane of the** TSF shall

- present the signature_algorithms extension with support for the following algorithms:
  - rsa_pkcs1 with sha256(0x0401),
  - rsa_pkcs1with sha384(0x0501),
  - rsa_pkcs1 with sha512(0x0601),
  - ecdsa_secp256r1 with sha256(0x0403),

- ecdsa_secp384r1 with sha384(0x0503),
- rsa_pss_rsae with sha256(0x0804),
- rsa_pss_rsae with sha384(0x0805),
- rsa_pss_rsae with sha512(0x0806),

and no other algorithms.

FCS_TLSC_EXT.1.6 The **data plane of the** TSF provides the ability to configure the list of supported ciphersuites as defined in FCS_TLSC_EXT.1.1.

FCS_TLSC_EXT.1.7 The **data plane of the** TSF shall prohibit the use of the following extensions:

- Early data extension
- Post-handshake client authentication according to RFC 8446, Section 4.2.6.

FCS_TLSC_EXT.1.8 The **data plane of the** TSF shall not use PSKs.

FCS_TLSC_EXT.1.9 The **data plane of the** TSF shall support TLS 1.2 secure renegotiation through use of the "renegotiation_info" TLS extension in accordance with RFC 5746, reject TLS 1.3 renegotiation attempts.

6.2.2.14 FCS_TLSC_EXT.2 TLS Client support for mutual authentication - Optional

FCS_TLSC_EXT.2.1 The **data plane of the** TSF shall support TLS communication with mutual authentication using X.509v3 certificates.

6.2.2.15 FCS_TLSS_EXT.1[1] TLS Server Protocol (Data Plane Server) - Selection

FCS_TLSS_EXT.1.1[1] The **data plane of the** TSF shall implement TLS 1.3 (RFC 8446), TLS 1.2 (RFC 5246) and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

- Select supported ciphersuites for TLS 1.2 from List 1:
  - 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 8422
  - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 8422
  - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 8422
  - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 8422
  - 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
- Select supported ciphersuites for TLS 1.3 from List 2:
    - TLS_AES_128_GCM_SHA256
    - TLS_AES_256_GCM_SHA384

and no other ciphersuites.

FCS_TLSS_EXT.1.2[1] The **data plane of the** TSF shall authenticate itself using X.509 certificate(s) using RSA with key size 2048, 3072, 4096 bits; ECDSA over NIST curves secp256r1, secp384r1 and no other curves.

FCS_TLSS_EXT.1.3[1] The **data plane of the** TSF shall perform key exchange using:

- RSA key establishment with key size 2048, 3072, 4096 bits;
- EC Diffie-Hellman key agreement over NIST curves secp256r1, secp384r1 and no other curves;
- Diffie-Hellman parameters ffdhe2048, ffdhe3072, ffdhe4096.

FCS_TLSS_EXT.1.4[1] The **data plane of the** TSF shall support no session resumption, session resumption based on session tickets according to RFC 5077 (TLS 1.2), session resumption according to RFC 8446 (TLS 1.3).

FCS_TLSS_EXT.1.5[1] The **data plane of the** TSF provides the ability to configure the list of supported ciphersuites as defined in FCS_TLSS_EXT.1.1**[1]**.

FCS_TLSS_EXT.1.6[1] The **data plane of the** TSF shall prohibit the use of the following extensions:

- Early data extension

FCS_TLSS_EXT.1.7[1] The **data plane of the** TSF shall not use PSKs.

FCS_TLSS_EXT.1.8[1] The **data plane of the** TSF shall support secure renegotiation in accordance with RFC 5746 by always including the "renegotiation_info" TLS extension in TLS 1.2 ServerHello messages, reject TLS 1.3 renegotiation attempts.

6.2.2.16 FCS_TLSS_EXT.1[2] TLS Server Protocol (Control Plane Server) - Selection

FCS_TLSS_EXT.1.1[2] 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:

- Select supported ciphersuites for TLS 1.2 from List 1:
    - 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

and no other ciphersuites.

FCS_TLSS_EXT.1.2[2] The **control plane of the** TSF shall authenticate itself using X.509 certificate(s) using RSA with key size 2048, 3072, 4096 bits; ECDSA over NIST curves secp256r1, secp384r1 and no other curves.

FCS_TLSS_EXT.1.3[2] The **control plane of the** TSF shall perform key exchange using:

- RSA key establishment with key size 2048, 3072, 4096 bits;
- EC Diffie-Hellman key agreement over NIST curves secp256r1, secp384r1 and no other curves.

FCS_TLSS_EXT.1.4[2] The **control plane of the** TSF shall support no session resumption, session resumption based on session tickets according to RFC 5077 (TLS 1.2).

FCS_TLSS_EXT.1.5[2] The **control plane of the** TSF provides the ability to configure the list of supported ciphersuites as defined in FCS_TLSS_EXT.1.1**[2]**.

FCS_TLSS_EXT.1.6[2] The **control plane of the** TSF shall prohibit the use of the following extensions:

- Early data extension

FCS_TLSS_EXT.1.7[2] The **control plane of the** TSF shall not use PSKs.

FCS_TLSS_EXT.1.8[2] The **control plane of the** TSF shall support secure renegotiation in accordance with RFC 5746 by always including the "renegotiation_info" TLS extension in TLS 1.2 ServerHello messages.

6.2.3 Identification and Authentication (FIA)

6.2.3.1 FIA_AFL.1 Authentication Failure Management - Selection

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 *the lockout reset action* is taken by an Administrator; prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed.

6.2.3.2 FIA_PMG_EXT.1 Password Management - Selection

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

a. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")",“`”, “~”, “-“, “+”, “=”, “[“, “]”, “{“, “}”, “;”, “’”, “:”, “””, “, “, “.”, “/”, “<”, “>”, “”, “|”, “\”;
b. Minimum password length shall be configurable to between *15* and *255* characters.

6.2.3.3 FIA_UAU.7 Protected Authentication Feedback - Selection

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.

6.2.3.4 FIA_UIA_EXT.1 User Identification and Authentication - Mandatory

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.

FIA_UIA_EXT.1.3 The TSF shall provide the following remote authentication mechanisms Web GUI password, SSH password, SSH public key and no other mechanism. The TSF shall provide the following local authentication mechanisms password-based.

FIA_UIA_EXT.1.4 The TSF shall authenticate any administrative user’s claimed identity according to each authentication mechanism specified in FIA_UIA_EXT.1.3.

ST Application Note: TD0900 applies to the SFR definition.

6.2.3.5 FIA_X509_EXT.1/Rev X.509 Certificate Validation - Selection

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

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

6.2.3.6 FIA_X509_EXT.2 X.509 Certificate Authentication - Selection

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

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall <u>accept the certificate</u>.

6.2.3.7 FIA_X509_EXT.3 X.509 Certificate Requests - Selection

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

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

6.2.4 Security Management (FMT)

6.2.4.1 FMT_MOF.1/ManualUpdate Management of security functions behavior - Mandatory

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

6.2.4.2 FMT_MOF.1/Services Management of security functions behavior - Selection

FMT_MOF.1.1/Services The TSF shall restrict the ability to start and stop services to Security Administrators.

6.2.4.3 FMT_MTD.1/CoreData Management of TSF Data - Mandatory

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

6.2.4.4 FMT_MTD.1/CryptoKeys Management of TSF Data - Selection

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

6.2.4.5 FMT_SMF.1 Specification of Management Functions - Mandatory

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

- Ability to administer the TOE remotely;
- Ability to configure the access banner;
- Ability to configure the remote session inactivity time before session termination;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- Ability to start and stop services;
- Ability to configure local audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full; changes to local audit storage size);
- Ability to manage the cryptographic keys;
- Ability to configure the cryptographic functionality;
- Ability to configure thresholds for SSH rekeying;
- Ability to set the time which is used for time-stamps;
- Ability to configure NTP;
- Ability to administer the TOE locally;
- Ability to configure the local session inactivity time before session termination or locking;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- Ability to manage the trusted public keys database.

ST Application Note: TD0880 applies to the SFR definition.

6.2.4.6 FMT_SMR.2 Restrictions on security roles - Mandatory

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 remotely

are satisfied.

6.2.5 Protection of TSF (FPT)

6.2.5.1 FPT_APW_EXT.1 Protection of Administrator Passwords - Selection

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

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

6.2.5.2 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys) - Mandatory

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

6.2.5.3 FPT_STM_EXT.1 Reliable Time Stamps - Mandatory

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, synchronise time with an NTP server.

6.2.5.4 FPT_TST_EXT.1 TSF Testing - Mandatory

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests:

- During initial start-up (on power on) to verify the integrity of the TOE firmware and software;
- Prior to providing any cryptographic service and *at power-up* to verify correct operation of cryptographic implementation necessary to fulfil the TSF;
- on-demand, at the conditions *periodically during normal operation* self-tests *TOE Software Integrity Verification*.
- Start-up self-tests *BIOS Power-On Self-Test (POST) on F5 devices*.

to demonstrate the correct operation of the TSF.

ST Application Note: TD0836 applies to the SFR definition.

FPT_TST_EXT.1.2 The TSF shall respond to all failures by *halting the TOE for all self-test failures except TOE software integrity test errors executed during normal operation result in reporting the errors*.

6.2.5.5 FPT_TUD_EXT.1 Trusted Update - Mandatory

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 prior to installing those updates.

6.2.6 TOE Access (FTA)

6.2.6.1 FTA_SSL.3 TSF-initiated Termination - Mandatory

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

6.2.6.2 FTA_SSL.4 User-initiated Termination - Mandatory

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

6.2.6.3 FTA_SSL_EXT.1 TSF-initiated Session Locking - Selection

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions,

- terminate the session

after a Security Administrator-specified time period of inactivity.

6.2.6.4 FTA_TAB.1 Default TOE Access Banners - Mandatory

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.

6.2.7 Trusted path/channels (FTP)

6.2.7.1 FTP_ITC.1 Inter-TSF trusted channel - Mandatory

FTP_ITC.1.1 The TSF shall be capable of using TLS to provide a trusted communication channel between itself and 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 disclosure and detection of modification of the channel data.

FTP_ITC.1.2 The TSF shall permitthe TSF, 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*.

6.2.7.2 FTP_TRP.1/Admin Trusted Path - Mandatory

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 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 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.

6.3 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 NDcPP.

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

Assurance Class	Assurance Component ID	Assurance Component Name
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 – conformance
AVA: Vulnerability assessment	AVA_VAN.1	Vulnerability survey

Table 7: Security Assurance Requirements

ASE_TSS.1 is refined by the NDcPP as noted below:

> ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR. **In the case of entropy analysis, the TSS is used in conjunction with required supplementary information on Entropy.**

In addition, the TOE will provide the evidence necessary for the evaluators to perform the evaluation activities defined in the Evaluation Activities for Network Device cPP document.

6.4 Security Requirements Rationale

This Security Target makes no modifications or additions to the NDcPP security problem definition, security objectives, or security assurance requirements. The security functionality requirements claimed in this ST include:

- all of the required/mandatory SFRs from the NDcPP,
- selected optional SFRs from the NDcPP,
- mandatory selection-based SFRs from the NDcPP,
- all of the required/mandatory SFRs from the PKG_SSH,
- mandatory selection-based SFRs from the PKG_SSH,

There are no additional SFRs or SARS included in this ST. Operations performed on the SFRs comply the corresponding Application Notes in the NDcPP and/or PKG_SSH.

6.4.1 Security Functional Requirement Dependencies

All of the security functional requirements claimed in this Security Target are taken directly from the NDcPP version 3.0e or PKG_SSH v1.0. All operations on the SFRs have been completed correctly. Therefore, the dependency rationale used by the NDcPP version 3.0e is considered applicable and acceptable since the NDcPP has been validated and 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 is a standalone TOE storing audit data locally and transmitting audit data to external syslog hosts.

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 8 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 8: Audit Logs and Their Content

The TOE includes within each audit record the information required by FAU_GEN.1.2 and specified in Table 6.

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 even 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 older syslog file(s) 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 80% 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_EXT.3.

7.2 Cryptographic Support

The TOE utilizes 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.

The TOE also uses the SHA-1 message digest algorithm for authenticating NTP servers.

7.2.1 Key Generation and Establishment

The asymmetric 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, 4096	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 created 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 Revision 3	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.
FFC	FFC NIST SP 800-56A Revision 3 and groups listed in RFC 7919	Safe-prime groups: ffdhe2048, ffdhe3072, ffdhe4096	For data plane TLS key exchange.

Table 9: Key generation in the TOE

The TOE also generates TLS session keys and SSH session keys while setting up the communication session.

The TOE offers administrative interfaces for creating a private key and certificate signing request (CSR). See Section 7.3.2 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 Primary Key and a Unit Key to protect sensitive configuration attributes. The Primary 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 Primary Key using 128-bit AES encryption. On F5 devices, 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 Primary Key. On virtual edition hypervisors, the Unit Key (a key-encrypting-key) is a symmetric key stored in a hidden file in the file system that is associated with the device and is used to protect the Primary Key. The Primary Key is encrypted with the Unit Key using 256-bit AES encryption. If the Unit Key is replaced, the old Unit Key is cleared by overwriting it with random data and then the new Unit Key is written. The Primary Key can be replaced using the tmsh command "`modify sys crypto master-key`".

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
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.

Application	Key type	Storage Location	Volatile/ Non-volatile	Zeroized when?	Description
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 10: 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, CTR, GCM modes)	128 256	payload encryption	AES as specified by ISO 18033-3 CBC as specified in ISO 10116 CTR as specified in ISO 10116 GCM as specified in ISO 19772	FCS_COP.1/DataEncryption
RSA	Modulus of 2048, 3072, 4096	certificate-based authentication, key exchange	FIPS PUB 186-4 Section 5.5 using RSASSA-PKCS1v1_5, ISO/IEC 9796-2	FCS_COP.1/SigGen

Algorithm	Key length (bits)	Purpose	Reference	SFR
ECDSA	256, 384 bits 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 / NTP authentication	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
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

Algorithm	Key length (bits)	Purpose	Reference	SFR
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 software 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 11: 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. On F5 devices, 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. On hypervisors, the jitterentropy-engine is the second entropy source.

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 is not used to implement any of the security functions claimed in the TSF. The administrative guidance will instruct the user to not use the SSH Client on 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, 8332.

SSH connections to the TOE's command line interface are protected using SSH version 2, using the cryptographic algorithms identified in Table 12.

Usage	Algorithm
Transport encryption algorithm	AES CBC and CTR mode with 128 and 256 bit-sizes keys aes128-gcm@openssh.com and aes256-gcm@openssh.com
Transport data integrity protection hashing algorithm	HMAC-SHA2-256
Public key authentication algorithms	ecdsa-sha2-nistp256, ecdsa-sha2-nistp384
Key exchange (hard-coded)	ecdh-sha2-nistp256 and ecdh-sha2-nistp384

Table 12: SSH Algorithms

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, diffie-hellman-group1-sha1 key exchange is intentionally disabled in the SSH implementation.

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_SSH_EXT.1 and FCS_SSHS_EXT.1.

7.2.6 TLS Protocol

The TOE implements both the TLS server and TLS client protocol. In the evaluated configuration, the TLS server protocol is implemented in both the data plane and the control plane, and the TLS client protocol is implemented only in the data plane.

TLS 1.3 is implemented by the data plane; not the control plane. The only ciphersuites supported by TLS 1.3 are TLS_AES_128_GCM_SHA256 and TLS_AES_256_GCM_SHA384.

Table 13 summarizes the ciphersuites supported by the evaluated configuration for TLS 1.2 connections for both the control plane and the data plane.

Cipher	Data Plane Client	Data Plane Server	Control Plane Server
TLS_RSA_WITH_AES_128_CBC_SHA	TLS v1.2	TLS v1.2	N/A
TLS_RSA_WITH_AES_256_CBC_SHA	TLS v1.2	TLS v1.2	N/A
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	TLS v1.2	TLS v1.2	N/A
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	TLS v1.2	TLS v1.2	N/A
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	TLS v1.2	TLS v1.2	N/A

Cipher	Data Plane Client	Data Plane Server	Control Plane Server
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	TLS v1.2	TLS v1.2	N/A
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS v1.2	TLS v1.2	N/A
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS v1.2	TLS v1.2	N/A
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

Table 13: TLS v1.2 Ciphersuites

All other proposed ciphersuites are rejected. The TOE provides a mechanism for configuring the list of supported ciphersuites for both the TLS server and TLS client.

The TLS implementations do not permit out-of-band provisioning of pre-shared keys (PSKs) in the evaluated configuration.

The TLS 1.2 implementation supports secure renegotiation through use of the “renegotiation_info” TLS extension in accordance with RFC 5746. The TLS 1.3 implementation rejects secure renegotiation attempts

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

7.2.6.1 TLS Client

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 data plane of the TOE is used to communicate with the external audit server.

For BIG-IP acting as TLS client, the TOE checks Common Name (CN) and DNS name. The CN or SAN in the certificate is compared by requiring an exact string match of the authenticate name against the IPv4 address in the certificate. The reference identifiers do not need to be converted by the TOE to perform this comparison.

The BIG-IP TLS client supports ECDH in the Client Hello by default. This can optionally be disabled by removing the corresponding ciphersuites, although individual curves cannot be configured.

The TLS client implementation indicates which signature/hash algorithm pairs it can support in digital signatures in handshake messages (and in certificates since no signature_algorithms_cert is present) using the signature_algorithms extension. The BIG-IP TLS client supports the following algorithms in the signature_algorithms extension:

- rsa_pkcs1 with sha256(0x0401),
- rsa_pkcs1with sha384(0x0501),

- rsa_pkcs1 with sha512(0x0601),
- ecdsa_secp256r1 with sha256(0x0403),
- ecdsa_secp384r1 with sha384(0x0503),
- rsa_pss_rsae with sha256(0x0804),
- rsa_pss_rsae with sha384(0x0805),
- rsa_pss_rsae with sha512(0x0806),

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

This functionality implements FCS_TLSC_EXT.1, FCS_TLSC_EXT.2.

7.2.6.2 TLS Server

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. Both the data and control plane support TLS 1.2 sessions. The data plane also supports TLS 1.3. The TLS server implementation in the TOE is configured to deny SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, and TLS 1.1 session requests.

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

The TLS 1.2 server supports session resumption based on session tickets according to RFC 5077. These session tickets adhere to the structural format described in Section 4 of RFC 5077. These session tickets are encrypted using the AES with CBC mode symmetric algorithm with 128 bit key length as defined in FCS_COP.1/DataEncryption. The TLS 1.3 server supports session resumption as defined in RFC 8446 Section 2.2. In addition, both the TLS 1.2 and TLS 1.3 servers support no session resumption.

Session establishment creates a session ID. When a new context is started and a session ID is offered, the session ID is verified to be acceptable to allow session resumption by checking the validity of the session ID in the session ID table, the age of the session ID, the cipher suite offered in the session ID, configuration settings of the session ID, and the Server Name Indication (SNI). Any failure in these validation steps listed below would trigger a full handshake.

Multiple contexts are supported for session resumption. A session can be constructed in one context and resumed in another context. The context which constructs the session ID during full handshake is the owner of that session ID and also validates the session ID and session state. Contexts which resume a session request that the originating context session owner validate the session ID and session state. If the originating context session validation response does not validate the session, a full handshake is triggered. Contexts validate sessions by requesting that the originating owner of a session validate a session before resumption can continue. If a session is not validated, a full handshake is triggered.

The TLS 1.2 and TLS 1.3 implementations authentication themselves using X.509 certificates using RSA with key size 2048, 3072, and 4096 bits or ECDSA over NIST curves secp256r1 and secp384r1.

When acting as a TLS server, BIG-IP generates key establishment parameters using RSA with key size 2048, 3072, and 4096 bits, and ECDH parameters over NIST curves secp256r1 and secp384r1. The TLS server key exchange message parameters (ECDH) are as defined / required by RFC 5246 Section 7.4.3 for TLS 1.2, and RFC 4492. For example, its classic ECDH using named curves with predefined parameters. The TOE does not support DHE_RSA ciphersuites, so server key exchange messages are not sent. For the data plane TLS 1.3 implementation the server key exchange parameters are as defined / required by RFC 8446 Section 7.4.2.

This functionality implements FCS_TLSS_EXT.1[1]-[2].

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.

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.

7.2.8 NTP Protocol

Administrators can configure BIG-IP to use NTP v4 to synchronize the real-time clock with an external NTP time source. The authenticity of the timestamps is ensured by verifying the integrity of the NTP packets using the SHA-1 message digest algorithm. BIG-IP does not update the system time from broadcast and/or multicast addresses. It is recommended that administrators configure at least 3 NTP time sources.

This functionality implements FCS_NTP_EXT.1.

7.3 Identification and Authentication

All management interfaces support the ability for administrative users (i.e., all users authorized to access the TOE's administrative interfaces) to be identified by a user name and authenticated by an individual password associated with that user account. 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 7.6.

For SSH public key authentication, the administrative user creates the key pair and sends the public key to BIG-IP for storage in the local user's authorized_keys file. The SSH server sends an encrypted challenge request to the SSH client requesting a connection using the user's shared public key. The SSH client uses the private key to decrypt the challenge request and responds to the SSH server. If the SSH client response is cryptographically correct, the client is granted access.

This functionality implements FIA_UIA_EXT.1.

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.

7.3.1 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 can be defined by the Security Agministrator. The default minimum password length value is 15; the valid range is from 15 to 255. This policy is enforced when users change their own passwords.

The TSF allows passwords to contain the following special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, “`”, “~”, “-“, “+”, “=”, “[“, “]”, “{“, “}”, “;”, “’”, “:”, “””, “ “, “.”, “/”, “<”, “>”, “”, “|”, “\”.

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 addition, an Administrator can issue a lockout reset command to reset the user’s failed authentication attempt counter, allowing the user’s lockout time to be reset immediately.

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.

7.3.2 Certificate Validation

The TOE implements certificate validation using the OpenSSL crypto library. The TLS client in the data plane supports TLS with mutual authentication for trusted communication channels with the external audit server.

The TOE supports validation of X.509 digital certificates using a certificate revocation list (CRL) as specified in [RFC 5280] 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 TSF cannot establish a connection to determine the validity of a certificate, 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.

7.4 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

7.7. 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 information on roles can be found in Section 7.4.1.)

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,
- start/stop services,
- manage TSF data, configure the login access banner,
- configure session inactivity timeout,
- manage cryptographic keys,
- configure cryptographic functionality,
- configure the RekeyLimit which defines how much data can be transmitted within an SSH connection before rekeying,
- configuration of trusted updates,
- set the time period for rejecting logins from an Administrator who has reached the maximum number of unsuccessful authentication attempts,
- configure NTP, 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.

The Security Administrator is able to start and stop the following services using the “`bigstart <stop, start, restart> <service>`” command or the following tmsh command “`tmsh <stop, start,`

`restart> /sys service <service>`”. The list of services that can be started and stopped are found in https://support.f5.com/csp/article/K67197865.

This functionality implements FMT_MOF.1/Services, FMT_SMF.1.

7.4.1 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 14 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 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. In addition, the Application Editor has full access to the APM-related configuration objects in BIG-IP. In particular, this includes the following authorizations with regard to management capabilities offered by the Configuration utility: • Config Utility (basic network and licensing configuration) - No access • Traffic Summary - Read-only • Reports (reporting on TOE users) - No access • Performance - Read-only • Statistics - Read-only • Local Traffic feature - Read-only access for Virtual Servers, Profiles, iRules, SNATs, and SSL Certificates; Modification (but not creation or deletion) of Nodes, Pools, Pool Members, and Monitors; Enabling/Disabling Nodes and Monitors • Access Profiles - Modification (but not Creation/Deletion) and activation of access policies with the exception of the "Max Concurrent Users" field • AAA Servers - Full access • ACLs - Full access • VLAN Based Routing - Read-only access for VLAN, Self-IP, and VLAN Gateways; Creation/Modification/Deletion of VLAN Selection Agents • Client IP Allocation - Full access • Network Access Resources - Full access • Customization - Full access • Advanced Customization - No access

Role	Associated rights
	• Session Variable Management - Creation/Modification/Deletion of Variable Assignment Agent; Creation/Modification (but not Deletion) of session variables • End User Security - Full access • Network features - No access to ARP configuration; Read-only access to all other features • System features - Read-only access; can change password for users in Application Editor role
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.

Table 14: BIG-IP User Roles

The Security Administrator role as defined in FMT_SMR.2 is provided by a combination of the BIG-IP user roles defined in Table 14, except for the Guest and No Access roles. Administrative users may be assigned different combinations of BIG-IP user roles, so some administrative users may not have the role(s) necessary to perform all of the administrative functions.

The "`tmsh sys crypto key`" command can be used by the Security Administrator to manage cryptographic keys on the TOE. The Security Administrator can manage the SSH key pairs, TLS key pairs, and pre-shared keys. The Security Administrator is able to perform the following operations on the keys:

- Importing of SSH public keys
- Generating SSL keys
- Changing keys
- Deleting keys
- Installing keys

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

7.5 Protection of the TSF

7.5.1 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.

7.5.2 Self-tests

The following self-tests are implemented by the TOE:

- On F5 devices. 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-eicheck.py utility) is run at power on, reboot, and once daily to check the integrity of the RPMs. This self-test can also be run on demand at any time.
- The cryptographic algorithm self-tests provided by OpenSSL are run at power on and reboot (during OpenSSL initialization).

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-eicheck.py utility provides HMAC integrity testing. 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. In addition, the HMAC integrity test is executed during every reboot 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.

7.5.3 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.

On F5 devices, 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

On hypervisors, the digital signatures of updates provided by F5 must be verified by the Security Administrator prior to being installed. The digital signatures of the image files and the product ISO update and image files are located on the F5 customer download website. The administrative guidance instructs the customer to:

- Download the image files or product ISO / image files and digital signatures (Note: access to the F5 customer website requires a trusted protocol such as HTTPS)
- Verify the digital signatures of the downloaded files
- Install the update

A signature file exists for each ISO, software change update or image file provided by F5. The content of the signature file includes two digital signatures of the ISO image file: one SHA256 digest and one SHA384 digest. The private and public keys are generated using the OpenSSL utility. The signing key is a 2048 bit or 3072 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 or the SHA384 hash of software archives using 3072-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.

7.5.4 Time Source

The TOE provides reliable time stamps for its own use, in particular in audit records and other time-sensitive security functionality. For F5 devices, the TOE hardware includes a hardware-based clock and the TOE's operating system makes the real-time clock available through a mcpd-maintained time stamp.

Administrators can manually set the time or configure BIG-IP to use NTP to synchronize the time with an external NTP time source. When manually setting the time, Administrators set the hardware-based clock on F5 devices and set the BIG-IP system clock on hypervisors. Refer to Section 7.2.8 for more information on NTP.

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 FCS_NTP_EXT.1, FPT_STM_EXT.1.

7.6 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 local and remote administrative user sessions (Console, Configuration Utility or tmsh) 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 of the administrative interface in use and therefore close an authenticated session).

This functionality implements FTA_SSL.4.

7.7 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 7.3.2, 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 1.2, using the ciphersuites identified in FCS_TLSS_EXT.1[2].
- 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.

