BIG-IP 11.5.1 HF 10 ADF-Base Security Target

Version:	**1.7**
Status:	**Release**
Last Update:	**2017-02-06**

Trademarks

The following terms are trademarks of F5:Networks, Inc.

- BIG-IP
- Application Delivery Firewall
- Local Traffic Manager
- Access Policy Manager
- Application Security Manager
- TMOS Platform

Other company, product, and service names may be trademarks or service marks of others.

Legal Notice

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

This document may be reproduced or distributed in any form without prior permission provided the copyright notice is retained on all copies. Modified versions of this document may be freely distributed provided that they are clearly identified as such, and this copyright is included intact.

Revision History

Revision	Date	Author(s)	Changes to Previous Revision
1.0	2015-12-08	Staffan Persson	Finalized version
1.1	2015-12-11	Staffan Persson	Clarified cipher specs and documentation list.
1.2	2016-02-07	Staffan Persson	Addressed certifier comments and clarified the administrative roles and their rights. Also addressed some other issues identified in the Security Target.
1.3	2016-02-08	Staffan Persson	Changes to application notes and crypto tables, and fixed some typos.
1.4	2016-03-09	Staffan Persson	Moved FPT_STM.1 from the TSF to the TOE environment.
1.5	2016-04-28	Staffan Persson	Removed TLS 1.0.
1.6	2017-01-09	Staffan Persson	Updated based on evaluator comments.
1.7	2017-02-06	Staffan Persson	Updated based on more comments.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	BIG-IP 11.5.1 HF 10 ADF-Base Security Target
Version:	1.7
Status:	Release
Date:	2017-02-06
Sponsor:	F5 Networks, Inc.
Developer:	F5 Networks, Inc.
Keywords:	Security Target, Common Criteria, F5, Application Delivery Controller, Firewall, Networking

1.2 TOE Identification

The TOE is BIG-IP ADF-Base Version 11.5.1 HF10.

1.3 TOE Type

The TOE type is Networking Device. In particular the TOE is a firewall with stateful traffic filtering. The TOE is the base configuration of a product from the BIG-IP product family, called Application Delivery Controllers that contains the core security functionality. The TOE is designed with the following Protection Profiles in mind: Protection Profile for Network Devices v1.1 (NDPP), as well as the Network Device Protection Profile (NDPP) Extended Package Stateful Traffic Filter Firewall (FWPP).

1.4 TOE Overview

The TOE is part of the BIG-IP product, which is an appliance containing both hardware and software. The TOE is an Application Delivery Controller, which is an Application Delivery Firewall (ADF-Base) that provides network traffic management and firewall capabilities.

The TOE is software only and sits on top of F5's Traffic Management Operating System (TMOS) that runs on hardware provided by F5.

A summary of the security functionality offered by the TOE follows. Section 1.5.5 (TOE boundaries) provides further information on the scope of the TOE and evaluated configurations, as well as on the supported hardware platforms.

1.4.1 Basic security functionality offered by the TOE

The TOE's provides the following security functionality:

- Firewall: The TOE offers basic firewall functionality, including stateful packet inspection and network address translation, and logic to mitigate denial-of-service attacks.
- Failover functionality: The TOE is configured as two redundant systems that synchronize their configuration data. The TOE will detect failures that may occur in hard- or software of one system and fail over to the redundant system while maintaining a secure configuration.

- Auditing capabilities: BIG-IP implements syslog audit capabilities by generating audit records for security-relevant events.
- Authentication: The TOE provides authentication of TLS connections.
- Administration capabilities: A command line interface, a web-based GUI, and a web-based API are provided to administrators for all relevant configuration of security functionality. 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").
- Communications security: The TOE can establish encrypted communication channels with external entities (Traffic TLS) as well as remote management connections with SSH.

1.5 TOE Description

1.5.1 Introduction

Figure 1 provides a schematic example of the TOE's role and location in a networking environment.

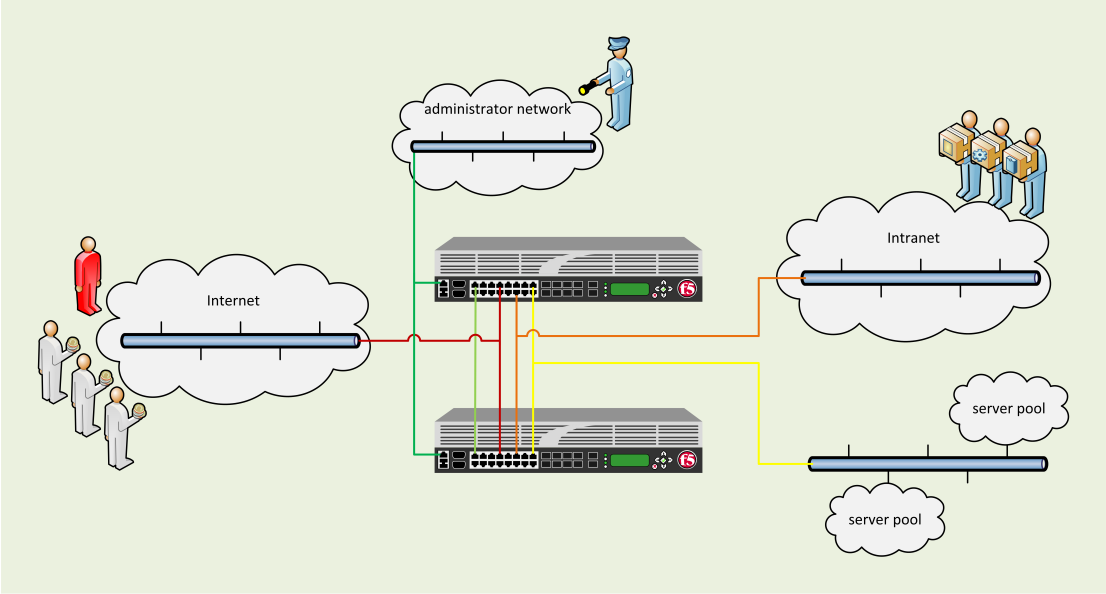

Figure 1: Schematic example of a BIG-IP network environment

The F5 hardware hosting BIG-IP is depicted by the two redundant network devices in the diagram. In this example:

- Internet connections (dark red network connection) are mediated by BIG-IP to provide access to certain resources located in an organization's internal server pool (yellow network connection), for example to a web-based e-commerce system presenting a storefront to consumers
- Users in the organization's Intranet (orange network connection) also access resources in the server pools to interact with the internal server pool
- Network administrators connect to BIG-IP via a dedicated network interface (dark green network connection) to administer the TOE

- The TOE is set up in a redundant failover configuration, with heartbeat monitoring and reporting via a data link between the two instances (light green connections)

1.5.2 Architecture

The TOE is implemented on top of F5's Traffic Management Operating System (TMOS) that runs on hardware provided by F5. TMOS is a Linux operating system that runs on appliance hardware. Neither the hardware, nor the TMOS operating system (with certain exceptions spelled out in this Security Target, i.e. the OpenSSH, OpenSSL, PAM, and syslog implementations provided by the TMOS operating system) are part of the TOE.

At the core of the BIG-IP is the Traffic Management Microkernel (TMM), representing the data plane of the product when compared to traditional network device architectures. It is implemented by a daemon running with root privileges, performing its own memory management, and having direct access to the network hardware. TMM implements a number of sequential filters both for the “client-side” and “server-side” network interfaces served by BIG-IP. The filters implemented in TMM include a TCP, TLS, compression, and HTTP filter, amongst others. If the hardware (TOE environment) provides more than one CPU, TMM runs multi-threaded (one thread per CPU). In this case, disaggregators implemented in hardware or, depending on the underlying appliance, firmware, are responsible for de-multiplexing and multiplexing network traffic for handling by an individual TMM thread. In addition to the actual switch hardware, F5 appliance hardware (TOE environment) 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 implemented in TMM, with additional and more complex logic implemented in a counter-part implemented in a Linux-based daemon (module). The plug-in modules relevant to this evaluation are shown in Figure 2, include:

- Local Traffic Manager (LTM): authentication of HTTP traffic and advanced traffic forwarding directives
- Advanced Firewall Manager (AFM): network filtering as described in the [FWPP].

A diagram depicting aspects of the TOE's architecture, and the boundaries of the TOE, is provided in Figure 2.

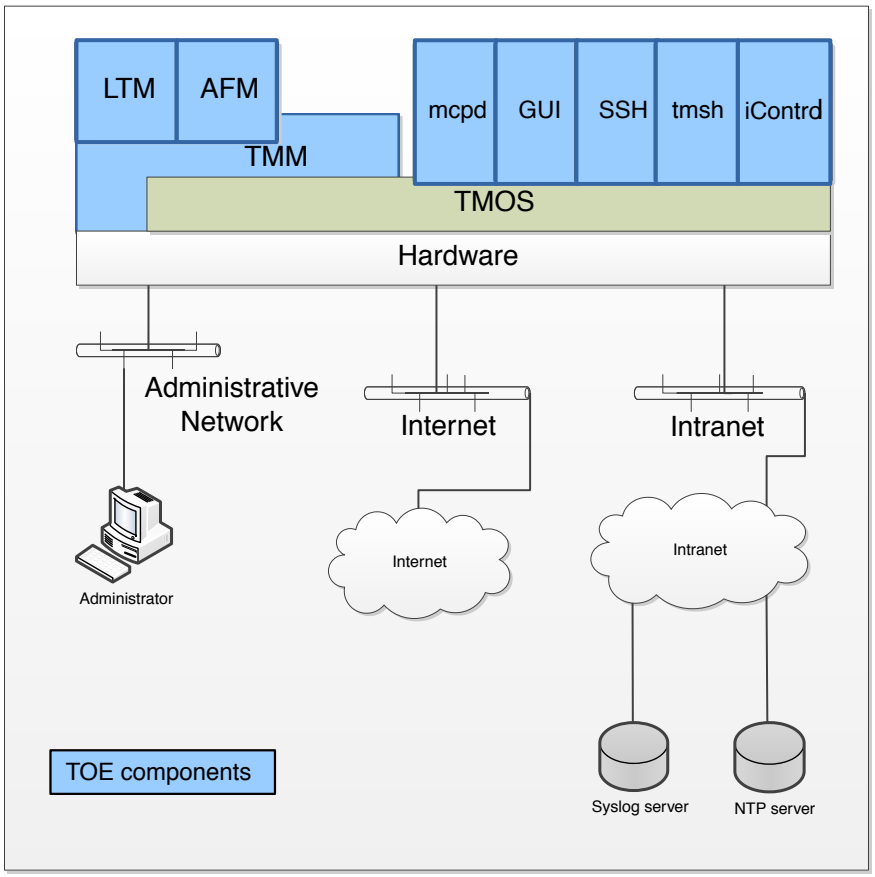

Figure 2: Architectural aspects of BIG-IP

Section 1.5.5 describes the packages, and functions, that are available in different product offerings in more detail.

In addition to data plane functionality implemented in TMM, the TOE comprises a number of software pieces that run on the Linux host in order to provide:

- Auditing functionality, by generating audit events 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.)
- Management functionality, presented to consumers both via a dedicated shell (traffic management shell, "tmsh") that can be reached by administrators via SSH; and via a both a web GUI and a SOAP protocol interface ("iControl API") that can be reached through a network interface. 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 is enforced on all administrative interfaces. Local administration is using password authentication with an internal password repository, relying on a password policy enforced by the TOE. Remote administrators are authenticated by the TOE using certificates that are validated by the TOE.

Not shown in Figure 2 is also a connection to a redundant BIG-IP host for failover purposes. Synchronization information, availability status, etc. is exchanged constantly by the machines via a dedicated network connection between the machines.

Multiple modules in the TOE and its operational environment provide cryptographic services for the TSF. This is further discussed in section 1.5.3.8.

1.5.3 Security functionality

This section provides an overview of the security functions implemented by the TOE.

1.5.3.1 Authentication

Administrators

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.

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

Network traffic

The Local Traffic Manager module (LTM) in *BIG-IP ADF-Base* supports the authentication of network sessions ("traffic authentication") using the following mechanism:

- TLS client certificate authentication (certificates, user groups, and roles)

Remote clients can authenticate the TOE by validating server certificates presented by the TOE.

1.5.3.2 Access control

Administrators

BIG-IP implements role-based access control. The roles 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 administrative users by authorized administrators. The list of roles is provided in Table 13.

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 to restrict administrative access of individual administrators to configuration objects that belong to the partition that has been assigned to the administrative user.

1.5.3.3 Stateful Firewall

BIG-IP ADF-Base implements a full-featured stateful firewall for Level 3 / Level 4 network traffic.

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

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

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

1.5.3.4 Synchronization and failover

BIG-IP supports the creation of redundant system configurations by clustering multiple BIG-IP devices, i.e., same Part Number, and configuring failover relationships. The evaluated configuration, in particular, comprises two BIG-IP systems configured in an active/standby failover configuration, which synchronizes configuration data between the devices, provides state and persistence monitoring, and allows the standby device to failover if either the active device sends a corresponding request, or the standby device detects missing heartbeats from the active device, continuing 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.

1.5.3.5 Auditing

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, and the authentication of traffic.

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

1.5.3.6 TSF 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 a dedicated connection that allows administration of most aspects of the TSF.
- traffic management shell (tmsh)
  tmsh is a command line interface available via SSH. It allows administration of all aspects of the TSF.
- iControl API
  The iControl API is a SOAP/XML based interface that allows programmatic access to the TSF configuration via a dedicated connection.

Additionally, management users can write "iRules®"; scripts used to direct and manipulate the way that the BIG-IP system manages application traffic. iRules® are written in Tools Command Language (Tcl), an industry-standard programming language. The iRules scripts would support additional application level traffic filtering beyond the functionality which is part of the evaluated configuration.

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 system.

Other security features associated with management interfaces include:

- session time-outs for Configuration utility and tmsh sessions

1.5.3.7 Communications security

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

Generic network traffic

BIG-IP ADF-Base's LTM allows the termination of 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.

Certificate validation is performed for connection to external servers. Certificate revocation checks are using locally kept CRLs.

Administrative traffic

The TOE secures administrative traffic (i.e., administrators connecting to the TOE in order to configure and maintain it). Remote access to the traffic management shell (tmsh) is secured via SSH.

Note: *Access to the web-based Configuration utility and iControl API is provided via a dedicated connection.*

OpenSSH

The TOE SSH implementation is based on OpenSSH Version openssh-4.3p2; 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-256
- The transport mechanism uses SSH_RSA as public key algorithm (ssh_rsa)
- The transport data integrity algorithm is limited to hmac-sha1
- The SSH protocol key exchange mechanism is limited to diffie-hellman-group14-sha1

Remote logging

The TOE offers the establishment of TLS sessions with external log hosts for protection of audit records in transfer, using the mechanism described in the previous section.

1.5.3.8 Cryptography

The cryptography in the BIG-IP rely on cryptographic mechanisms for their effective implementation. The cryptographic mechanisms are described in the TOE Summary Specification

1.5.4 Operational environment support

The TOE relies on its operational environment to support some of the TOE security functions, as well as to provide a certain degree of protection of the TOE itself.

1.5.4.1 Physical environment

Protection expected from the TOE's physical operational environment includes:

- The device hosting the TOE needs to be protected from unauthorized physical access.
- If the TOE is used to terminate TLS sessions and forward web traffic unprotected, then the networks used to forward this traffic to its final destination need to be protected appropriately.

1.5.4.2 Runtime environment

The TOE relies on its runtime environment, i.e. the Linux operating system and hardware hosting the TOE, for a number of functions:

- Provision of entropy by the Cavium hardware.
- If present and configured to be used, protection of certain cryptographic key material by a FIPS 140-2 validated cryptographic module. Note that this module was not subject to evaluation.
- Enforcement of filtering decisions by the switch fabric and HSB.

1.5.4.3 Network environment

In addition to the support provided by the system hosting the TOE, dependencies on external systems exist. Aspects that an organization using the TOE need to be considerate of in order to ensure the effectiveness of the TOE include:

- Time servers need to provide an accurate time to the TOE.
- Log servers need to be able to handle the amount of syslog traffic generated by the TOE, and preserve a proper context.

1.5.4.4 Virtualized environment

The TOE can be deployed on a vCMP system as a guest. This involves defining a guest on the underlying host system, including the assignment of CPU cores and chassis slots, as well as an IP address for the management port. The host also manages an overall list of available VLANs that can be assigned to the guest to operate on.

The TOE in this case is the guest running in the virtualized environment, implementing the same functionality as described in this Security Target for stand-alone appliances. Any host functionality and virtualization technology is not subject to this evaluation, and a cooperative environment on the vCMP platform is assumed.

1.5.5 TOE boundaries

The following sections describe the physical and logical boundaries of the TOE.

1.5.5.1 Physical boundaries

The TOE is software only as identified in Section 1.2

The evaluated configuration of *BIG-IP ADF-Base* represents a licensing option with the following modules present and operational.

- Traffic Management Microkernel (TMM)
- Local Traffic Manager (LTM), and

- Advanced Firewall Manager (AFM).

The TOE is available via electronic download from F5's website. Mechanisms to allow consumers to validate the authenticity and integrity of the download are provided.

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

- Guidance Supplement: AGD_PRE and AGD_OPE, BIG-IP ADF-Base and ADC-AP Release 11.5.1
- BIG-IP Local Traffic Manager: Concepts, Version 11.5.1
- BIG-IP Local Traffic Manager: Implementations 11.5.1
- BIG-IP Redundant Systems Configuration Guide, Version 11.0
- BIG-IP TMOS: Concepts, Version 11.5
- BIG-IP TMOS: Implementations, Version 11.5.1
- Traffic Management Shell (tmsh) Reference Guide, Version 11.5.1
- BIG-IP Network Firewall: Policies and Implementations, Version 11.5.1
- BIG-IP TMOS: IP Routing Administration, Version 11.5.1
- External Monitoring of BIG-IP Systems: Implementations, Version 11.5
- BIG-IP Data Center Firewall Configuration Guide, Version 11.2
- BIG-IP Device Service Clustering: Administration, Version 11.5
- vCMP for Appliance Models: Administration, Version 11.5

Physical boundaries between the TOE and its runtime environment are described in section 1.5.2. In particular:

- The base Linux operating system (TMOS), other than particular applications implementing TSF, is considered part of the runtime environment. Those parts implementing TSF and considered part of the TOE, are as follows:
  - openSSL
  - openSSH
  - PAM
  - SYSLOG
- Hardware is considered part of the runtime environment. This also includes the bitstream for the HSB. The following TOE and hardware appliance model combinations are covered by this evaluation:

SKU	Model	VCMP?	Software
F5-BIG-LTM-10200V F5-ADD-BIG-AFM-10000 F5-ADD-BIG-MODE	10000	Y	LTM+AFM w/ AppMode
F5-BIG-LTM-10200V-F F5-ADD-BIG-AFM-10000 F5-ADD-BIG-MODE	10000	Y	LTM+AFM w/ AppMode
F5-BIG-LTM-10200V-S F5-ADD-BIG-AFM-10000 F5-ADD-BIG-MODE	10000	Y	LTM+AFM w/ AppMode
F5-VPR-LTM-C4480-AC	B4300	Y	LTM+AFM w/ AppMode

SKU	Model	VCMP?	Software
F5-VPR-LTM-B4300 F5-ADD-VPR-AFM-C4400 F5-ADD-VPR-VCMP-4480 F5-ADD-BIG-MODE			
F5-VPR-LTM-C4480-DCN F5-VPR-LTM-B4300N F5-ADD-VPR-AFM-C4400 F5-ADD-VPR-VCMP-4480 F5-ADD-BIG-MODE	B4300N	Y	LTM+AFM w/ AppMode
F5-BIG-LTM-5200V F5-ADD-BIG-AFM-5000 F5-ADD-BIG-MODE	5000	Y	LTM+AFM w/ AppMode
F5-BIG-LTM-7200V F5-ADD-BIG-AFM-7000 F5-ADD-BIG-MODE	7000	Y	LTM+AFM w/ AppMode
F5-BIG-LTM-7200V-S F5-ADD-BIG-AFM-7000 F5-ADD-BIG-MODE	7000	Y	LTM+AFM w/ AppMode
F5-BIG-LTM-7200V-F F5-ADD-BIG-AFM-7000 F5-ADD-BIG-MODE	7000	Y	LTM+AFM w/ AppMode
F5-BIG-LTM-10000S F5-ADD-BIG-AFM-10000 F5-ADD-BIG-MODE	10000	Y	LTM+AFM w/ AppMode
F5-VPR-LTM-C4480-AC F5-VPR-LTM-B4300 F5-ADD-VPR-AFM-C4400 F5-ADD-BIG-MODE	B4300	N	LTM+AFM w/ AppMode
F5-VPR-LTM-C4480-DCN F5-VPR-LTM-B4300N F5-ADD-VPR-AFM-C4400 F5-ADD-BIG-MODE	B4300N	N	LTM+AFM w/ AppMode
F5-BIG-LTM-5000S F5-ADD-BIG-AFM-5000 F5-ADD-BIG-MODE	5000	N	LTM+AFM w/ AppMode
F5-BIG-LTM-7000S F5-ADD-BIG-AFM-7000 F5-ADD-BIG-MODE	7000	N	LTM+AFM w/ AppMode

Table 1: Supported Hardware Models

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

- Client software (e.g., the BIG-IP Client for TLS VPN connections, endpoint inspection software executed on clients) is not part of the TOE.
- NTP and audit servers.

1.5.5.2 Logical boundaries / evaluated configuration

The security functions provided by the TOE are described above. This section discusses specific configurations that apply to the evaluated configuration, and provides an overview of functionality that - while present in BIG-IP - is not considered security functionality subject to this evaluation.

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

- Appliance mode is licensed. This results, amongst other effects, in root access to the underlying system being disabled, and Always-On Management not being able to access the host.
- A physical network port is dedicated on each device for the exchange of management traffic with the mirrored device (configuration synchronization, failover monitoring).
- Dynamic routing is excluded from the evaluated configuration.
- Disabled interfaces:
    - Shells other than tmsh are disabled. For example, bash and other user-serviceable shells are excluded.
    - Management of the TOE via SNMP is disabled.
    - Management of the TOE via the appliance's LCD display is disabled.
    - Remote (i.e., SSH) access to the Lights Out / Always On Management capabilities of the system is disabled.
    - Serial port console

No security claims have been made on the following functionality of BIG-IP. These functions can be used in the evaluated configuration, but they are not part of the security functionality that has been subject to evaluation:

- Client software for establishing connections to BIG-IP.
- Cookies: LTM uses cookies to support a number of traffic management state information. Cryptographic properties of these cookies (randomness of unique identifiers, encryption of cookie parameters by BIG-IP) have not been assessed in this evaluation.
- Certificate generation for use in traffic TLS. The certificate generation mechanism in BIG-IP is primarily provided for administrative and testing purposes. Production environments are expected to operate their own Public Key Infrastructure and supply certificates for the TOE's use in public-facing functionality.
- Advanced filtering capabilities in LTM. BIG-IP offers advanced firewalling and filtering capabilities.
- Antivirus and Database security services (external providers). Features are not enabled unless configured.
- Policy Builder: provides suggestions to be inserted or removed into/from security policy

1.5.6 Security Policy Model

The security policy model for BIG-IP is defined by the security functional requirements in section 6.1. This section summarizes the subjects and objects participating in the individual policies defined in the SFRs.

1.5.6.1 Administrator Access Control Policy

The following subjects and objects are involved in defining the administrative access to the TOE, i.e. authentication of administrative users and access control for configuration objects that these users can manipulate.

Subjects, and their security attributes:

- administrative users
  - user name
  - password
  - role
  - partition access - a user's partition (or "all")
  - terminal access - whether the user can configure the TOE via the tmsh
  - locked - whether the user's account is locked
  - password expiration counter
  - number of consecutive failed authentication attempts

Objects, and their security attributes:

- configuration objects
  - object type
  - partition

1.5.6.2 TSF and user data

TSF data:

- information representing the subjects and their security attributes identified in the policies above
- security attributes of objects and information identified in the policies above
- network packet header fields defined in FFW_RUL_EXT.1.
- audit setings and records
- cryptographic keys, certificates, settings, and other critical security parameters
- failover settings and status of peer systems in the failover configuration
- service settings
- partition settings
- firewall rules
- TOE update data
- configuration objects[1]
- timeout settings
- Identification and Authentication settings
- TOE component status
- traffic authentication settings
- TOE state data

[1] Configuration objects refer to the objects that define or store multiple types of TSF data used to configure the system with one operation

- system time

User data:

- network traffic mediated by the TSF

2 CC Conformance Claim

This Security Target is CC Part 2 extended and CC Part 3 conformant, with a claimed Evaluation Assurance Level 4 (EAL4), augmented by ALC_FLR.3.

Common Criteria [CC] version 3.1 revision 4 is the basis for this conformance claim.

The ST has been developed by using applicable NIAP PPs. However, the ST does not claim conformance to any of them.

The relevant NIAP PPs that have been used are [NDPP] Protection Profile for Network Devices, version 1.1 of 2012-06-08; and [FWPP] Network Device Protection Profile Extended Package Stateful Traffic Filter Firewall, version 1.0 of 2011-12-19

The ST modified the wording of several threats defined in [FWPP]. The intention and scope of these threats is identical to the threats in FWPP, however, wording was changed to be more explicit in the definitions of threat agents and the assets to be protected by the TOE. The original threat names were kept to allow easy referencing.

3 Security Problem Definition

3.1 Threat Environment

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

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

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

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

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

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

3.1.1 Threats countered by the TOE

T.ADMIN_ERROR

An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.

T.NETWORK_ACCESS

Unauthorized access may be achieved to services on a protected network from outside that network, or alternately services outside a protected network from inside the protected network.

T.NETWORK_DISCLOSURE

Sensitive information on a protected network might be disclosed resulting from ingress- or egress-based actions.

T.NETWORK_DOS

Attacks against services inside a protected network, or indirectly by virtue of access to malicious agents from within a protected network, might lead to denial of services otherwise available within a protected network.

T.NETWORK_MISUSE

Access to services made available by a protected network might be used counter to Operational Environment policies.

T.PUBLIC_NETWORKS

An attacker might be able to observe organizational data exchanged between the TOE and another trusted IT product through a public network.

T.RESOURCE_EXHAUSTION

An attacker causes network traffic to be mediated or otherwise handled by the TOE that exceeds the amount of traffic the TSF can reliably handle, causing unavailability of the TSF to legitimate users. (In particular, denying authorized administrators the possibility to administrate the TOE.)

T.TSF_FAILURE

Security mechanisms of the TOE may fail, leading to a compromise of the TSF.

T.UNDETECTED_ACTIONS

Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.

T.UNAUTHORIZED_ACCESS

A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.

T.UNAUTHORIZED_UPDATE

A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.

T.USER_DATA_REUSE

User data may be inadvertently sent to a destination not intended by the original sender.

3.2 Assumptions

3.2.1 Environment of use of the TOE

3.2.1.1 Physical

A.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.

3.2.1.2 Personnel

A.TRUSTED_ADMIN

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

A.TRAINED_ADMIN

TOE Administrators are carefully trained to follow and apply all administrator guidance.

3.2.1.3 Connectivity

A.CONNECTIONS

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

A.LOGSERVER

It is assumed that the environment is able to receive, store and protect the audit records generated by the TOE and provides means for the audit analysis, including time correlation.

A.MGMTNET

The management networks used for managing the TOE, synchronizing the fail-over system and connecting the TOE to NTP servers are private, separate physical networks that are protected from unauthorized physical and logical access.

A.NO_GENERAL_PURPOSE

It is assumed that 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.

A.PEERTRUST

Systems that are configured in a device group to synchronize configuration data between each other for a potential failover must be trustworthy. That means that they are all under the same administration as the TOE, identically configured and that the same assumptions can be made about them as for the TOE.

A.KEYS

It is assumed that digital certificates, certificate revocation lists (CRLs) used for certificate validation and private and public keys used for SSH client authentication generated externally, meeting the corresponding standards and providing sufficient security strength through the use of appropriate key lengths and message digest algorithms. It is also assumed that Administrators verify the integrity and authenticity of digital certificates and key material before importing them into the TOE, and verifying that certificates are signed using strong hash algorithms.

A.TIME

It is assumed that a reliable time is provided by the TOE environment to the TOE.

A.LDAP

It is assumed that a reliable LDAP service is provided by the TOE environment to the TOE for the provision of X.509 certificates.

3.3 Organizational Security Policies

P.ACCESS_BANNER

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

P.FAILOVER

The TOE shall provide failover functionality between redundant devices, maintaining a secure state during failover.

P.LTM-TRAFFICMGMT

The LTM module shall provide authentication of HTTP traffic, and HTTPS proxy functionality.

4 Security Objectives

The following sections define the security objectives for the TOE and for the TOE's operational environment.

4.1 Objectives for the TOE

O.ADDRESS_FILTERING

The TOE will provide the means to filter and log network packets based on source and destination addresses.

O.DISPLAY_BANNER

The TOE will display an advisory warning regarding use of the TOE.

O.FAILOVER

The TOE will provide failover capabilities between redundant configurations.

O.LTM-TRAFFICMGMT

The TOE will provide a Local Traffic Management (LTM) module that can authenticate HTTP traffic, and proxy HTTPS communications by terminating them and forwarding unencrypted traffic to internal web servers.

O.PORT_FILTERING

The TOE will provide the means to filter and log network packets based on source and destination transport layer ports.

O.PROTECTED_COMMUNICATIONS

The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.

O.RELATED_CONNECTION_FILTERING

For specific protocols, the TOE will dynamically permit a network packet flow in response to a connection permitted by the ruleset.

O.RESIDUAL_INFORMATION_CLEARING

The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.

O.RESOURCE_AVAILABILITY

The TOE shall provide mechanisms that mitigate user attempts to exhaust TOE resources (e.g., persistent storage).

O.SESSION_LOCK

The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.

O.STATEFUL_INSPECTION

The TOE will determine if a network packet belongs to an allowed established connection before applying the ruleset.

O.SYSTEM_MONITORING

The TOE will provide the capability to generate audit data and send those data to an external IT entity.

O.TOE_ADMINISTRATION

The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators.

O.TSF_SELF_TEST

The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

O.VERIFIABLE_UPDATES

The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered.

4.2 Objectives for the Operational Environment

OE.CONNECTIONS

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

OE.LOGSERVER

The TOE environment must be able to receive, store and protect the audit records generated by the TOE and provides means for the audit analysis, including time correlation.

OE.MGMTNET

The management networks used for managing the TOE, synchronizing the fail-over system and connecting the TOE to NTP servers are private, separate physical networks that are protected from unauthorized physical and logical access.

OE.NO_GENERAL_PURPOSE

There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.

OE.PEERTRUST

Systems that are configured in a device group to synchronize configuration data between each other for a potential failover must be trustworthy. That means that they are all under the same administration as the TOE, identically configured and that the same assumptions can be made about them as for the TOE.

OE.KEYS

Digital certificates, certificate revocation lists (CRLs) used for certificate validation and private and public keys used for SSH client authentication generated externally, meet the corresponding standards and provide sufficient security strength through the use of appropriate key lengths and message digest algorithms. Administrators must verify the integrity and authenticity of digital certificates and key material before importing them into the TOE, and verify that certificates are signed using strong hash algorithms.

OE.TIME

Reliable time stamp is provided by the TOE environment to the TOE.

OE.LDAP

Reliable LDAP service is provided by the TOE environment to the TOE for the provision of X.509 certificates.

OE.PHYSICAL

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

OE.TRUSTED_ADMIN

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

OE.TRAINED_ADMIN

TOE Administrators are carefully trained to follow and apply all administrator guidance.

4.3 Security Objectives Rationale

4.3.1 Coverage

The following table provides a mapping of TOE objectives to threats and policies, showing that each objective counters or enforces at least one threat or policy, respectively.

Objective	Threats / OSPs
O.ADDRESS_FILTERING	T.NETWORK_ACCESS T.NETWORK_DISCLOSURE T.NETWORK_DOS T.NETWORK_MISUSE
O.DISPLAY_BANNER	P.ACCESS_BANNER
O.FAILOVER	P.FAILOVER
O.LTM-TRAFFICMGMT	P.LTM-TRAFFICMGMT

Objective	Threats / OSPs
O.PORT_FILTERING	T.NETWORK_ACCESS T.NETWORK_DISCLOSURE T.NETWORK_DOS T.NETWORK_MISUSE
O.PROTECTED_COMMUNICATIONS	T.PUBLIC_NETWORKS T.UNAUTHORIZED_ACCESS
O.RELATED_CONNECTION_FILTERING	T.NETWORK_ACCESS
O.RESIDUAL_INFORMATION_CLEARING	T.USER_DATA_REUSE
O.RESOURCE_AVAILABILITY	T.RESOURCE_EXHAUSTION
O.SESSION_LOCK	T.UNAUTHORIZED_ACCESS
O.STATEFUL_INSPECTION	T.NETWORK_DOS
O.SYSTEM_MONITORING	T.ADMIN_ERROR T.NETWORK_MISUSE T.UNDETECTED_ACTIONS
O.TOE_ADMINISTRATION	T.UNAUTHORIZED_ACCESS
O.TSF_SELF_TEST	T.TSF_FAILURE
O.VERIFIABLE_UPDATES	T.UNAUTHORIZED_UPDATE

Table 2: Mapping of security objectives to threats and policies

The following table provides a mapping of the objectives for the Operational Environment to assumptions, threats and policies, showing that each objective holds, counters or enforces at least one assumption, threat or policy, respectively.

Objective	Assumptions / Threats / OSPs
OE.CONNECTIONS	A.CONNECTIONS
OE.LOGSERVER	A.LOGSERVER
OE.MGMTNET	A.MGMTNET
OE.NO_GENERAL_PURPOSE	A.NO_GENERAL_PURPOSE
OE.PEERTRUST	A.PEERTRUST
OE.KEYS	A.KEYS
OE.TIME	A.TIME
OE.LDAP	A.LDAP
OE.PHYSICAL	A.PHYSICAL

Objective	Assumptions / Threats / OSPs
OE.TRUSTED_ADMIN	A.TRUSTED_ADMIN T.ADMIN_ERROR
OE.TRAINED_ADMIN	A.TRAINED_ADMIN T.ADMIN_ERROR

Table 3: Mapping of security objectives for the Operational Environment to assumptions, threats and policies

4.3.2 Sufficiency

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat, when achieved, actually contributes to the removal, diminishing or mitigation of that threat.

Threat	Rationale for security objectives
T.ADMIN_ERROR	OE.TRAINED_ADMIN asks for administrators that are trained to follow all guidance documentation, OE.TRUSTED_ADMIN asks for administrators that can be trusted to follow all guidance documentation in the first place, while O.SYSTEM_MONITORING requires the TOE to implement accountability mechanisms that would allow the review of administrator actions, contributing to the detection of incorrect configurations.
T.NETWORK_ACCESS	This threat is mitigated by allowing an organization to specify filter rules that will be applied by the TOE to traffic based on IP addresses (O.ADDRESS_FILTERING), ports (O.PORT_FILTERING), and protocol-specific relationships between connections (such as the separate control and payload traffic for an FTP connection) (O.RELATED_CONNECTION_FILTERING).
T.NETWORK_DISCLOSURE	O.ADDRESS_FILTERING and O.PORT_FILTERING, respectively, require a network device to provide traffic filtering based on rules that include IP-addresses and ports. This allows an organization to reduce the threat of information disclosure through traffic mediated by the device.
T.NETWORK_DOS	In order to mitigate the threat of DOS attacks, O.STATEFUL_INSPECTION requires to augment the filtering based on addresses (O.ADDRESS_FILTERING) and ports (O.PORT_FILTERING) to include stateful filtering for relevant protocols.
T.NETWORK_MISUSE	The threat of misuse is mitigated by providing administrators with a means to generate log entries (O.SYSTEM_MONITORING) for traffic that matches filter rules per O.ADDRESS_FILTERING and O.PORT_FILTERING.
T.PUBLIC_NETWORKS	The TSF counters this threat by providing a trusted communication channel between itself and a remote BIG-IP instance that allows the forwarding of network traffic between those two, as covered by O.PROTECTED_COMMUNICATIONS.

Threat	Rationale for security objectives
T.RESOURCE_EXHAUSTION	The objective O.RESOURCE_AVAILABILITY expects that the TSF will be able to counter user attempts to exhaust TOE resources.
T.TSF_FAILURE	The need for an implementation of self-tests is defined in O.TSF_SELF_TEST, countering the threat of failed security mechanisms leading to compromise.
T.UNDETECTED_ACTIONS	O.SYSTEM_MONITORING defines the objective for auditing mechanisms to be implemented in the TOE.
T.UNAUTHORIZED_ACCESS	While O.TOE_ADMINISTRATION requires access control mechanisms to be in place for administration of the TOE, O.SESSION_LOCK and O.PROTECTED_COMMUNICATIONS contribute to countering this threat by requiring mitigation of session hijacking in particular, and the implementation of protected communication channels in general.
T.UNAUTHORIZED_UPDATE	O.VERIFIABLE_UPDATES requires the implementation of TSF mechanisms that contribute to verifying the integrity of software updates.
T.USER_DATA_REUSE	The threat of the TSF being coerced into disclosing data held in memory from a previous transaction to users not part of that transaction is addressed by O.RESIDUAL_INFORMATION_CLEARING asking for a mechanism to prevent this.

Table 4: Sufficiency of objectives countering threats

The following rationale provides justification that the security objectives for the environment are suitable to cover each individual assumption, that each security objective for the environment that traces back to an assumption about the environment of use of the TOE, when achieved, actually contributes to the environment achieving consistency with the assumption, and that if all security objectives for the environment that trace back to an assumption are achieved, the intended usage is supported.

Assumption	Rationale for security objectives
A.PHYSICAL	This assumption is reflected in OE.PHYSICAL, which contains the same wording.
A.TRUSTED_ADMIN	OE.TRUSTED_ADMIN reflects this assumption.
A.TRAINED_ADMIN	OE.TRAINED_ADMIN reflects this assumption.
A.CONNECTIONS	OE.CONNECTIONS upholds the assumption that the operational environment is configured in a way that allows the TOE to be the single point of enforcement for traffic between distinctive networks.
A.LOGSERVER	OE.LOGSERVER upholds the assumption that the operational environment is able to receive, store and protect the audit records generated by the TOE and provides means for the audit analysis, including time correlation.

Assumption	Rationale for security objectives
A.MGMTNET	OE.MGMTNET upholds the assumption that the operational environment is configured in a way that the management networks are private, separate physical networks that is protected from unauthorized physical and logical access.
A.NO_GENERAL_PURPOSE	A.NO_GENERAL_PURPOSE is implemented by OE.NO_GENERAL_PURPOSE, which contains the same wording.
A.PEERTRUST	A.PEERTRUST is implemented by OE.PEERTRUST, which contains the same wording.
A.KEYS	A.KEYS is implemented by OE.KEYS, which contains the same wording.
A.TIME	A.TIME is implemented by OE.TIME, which contains the same wording.
A.LDAP	A.LDAP is implemented by OE.LDAP, which contains the same wording.

Table 5: Sufficiency of objectives holding assumptions

The following rationale provides justification that the security objectives are suitable to cover each individual organizational security policy (OSP), that each security objective that traces back to an OSP, when achieved, actually contributes to the implementation of the OSP, and that if all security objectives that trace back to an OSP are achieved, the OSP is implemented.

OSP	Rationale for security objectives
P.ACCESS_BANNER	The policy is implemented by O.DISPLAY_BANNER requiring the TOE to implement such a banner.
P.FAILOVER	O.FAILOVER implements the objective for providing failover capabilities between redundant TOE systems.
P.LTM-TRAFFICMGMT	This OSP is enforced by O.LTM-TRAFFICMGMT, which requires the TOE to provide a module implementing the policy's requirements.

Table 6: Sufficiency of objectives enforcing Organizational Security Policies

5 Extended Components Definition

The Security Target draws upon the extended components implicitly defined in [NDPP] and [FWPP]. The extended components from the PPS are defined here for completeness.

5.1 Class FAU: Security audit

5.1.1 Security audit event storage (STG)

Management: FAU_STG_EXT.1

The following actions could be considered for the management functions in FMT:

a) Definition of the audit log destination system.

Audit: FAU_STG_EXT.1

There are no audit events foreseen.

5.1.1.1 FAU_STG_EXT.1 - External Audit Trail Storage

Hierarchical to:	No other components.
Dependencies:	FAU_GEN.1 Audit data generation

FAU_STG_EXT.1.1 The TSF shall be able to [selection: **transmit the generated audit data to an external IT entity, receive and store audit data from an external IT entity**] using a trusted channel implementing the [selection: **IPSec, SSH, TLS, TLS/HTTPS**] protocol.

Rationale

This extended component extends FAU_STG to transmit and store the generated audit data on a remote system via a protected channel.

5.2 Class FCS: Cryptographic support

5.2.1 Cryptographic key management (CKM)

Management: FCS_CKM_EXT.4

There are no management activities foreseen.

Audit: FCS_CKM_EXT.4

There are no audit events foreseen.

5.2.1.1 FCS_CKM_EXT.4 - Cryptographic Key Zeroization

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

FCS_CKM_EXT.4.1 The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

Rationale

This is a specific implementation of FCS_CKM.4 that requires explicit deletion via overwriting with zeros.

The zeroization applies to each intermediate storage area for plaintext key/cryptographic critical security parameter (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/cryptographic critical security parameter to another location.

5.2.2 Explicit HTTPS specification (HTTPS)

Management: FCS_HTTPS_EXT.1

There are no management activities foreseen.

Audit: FCS_HTTPS_EXT.1

There are no audit events foreseen.

5.2.2.1 FCS_HTTPS_EXT.1 - Explicit HTTPS specification

Hierarchical to: No other components.

Dependencies: FCS_TLS_EXT.1 Flexible TLS

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 as specified in FCS_TLS_EXT.1.

Rationale

This component is used to explicitly require a specific HTTPS implementation.

5.2.3 Random Bit Generation (RBG)

Management: FCS_RBG_EXT.1

There are no management activities foreseen.

Audit: FCS_RBG_EXT.1

There are no audit events foreseen.

5.2.3.1 FCS_RBG_EXT.1 - Random Bit Generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [selection, choose one of: **NIST Special Publication 800-90 using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES), Dual_EC_DRBG (any)], FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using**

AES] seeded by an entropy source that accumulated entropy from one or both of [selection: **a software-based noise source, a TSF-hardware-based noise source**] .

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of [selection, choose one of: **128 bits, 256 bits**] of entropy at least equal to the greatest bit length of the keys and authorization factors that it will generate.

Rationale

This component defines explicit requirements for the random number generator used in the TOE.

5.2.4 Explicit SSH specification (SSH)

Management: FCS_SSH_EXT.1

There are no management activities foreseen.

Audit: FCS_SSH_EXT.1

There are no audit events foreseen.

5.2.4.1 FCS_SSH_EXT.1 - Explicit SSH specification

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_SSH_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, and 4254.

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

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: **number of bytes**] bytes in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, [selection: **AEAD_AES_128_GCM, AEAD_AES_256_GCM, no other algorithms**].

FCS_SSH_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses SSH_RSA and [selection: **PGP-SIGN-RSA, PGP-SIGN-DSS, no other public key algorithms**]as its public key algorithm(s).

FCS_SSH_EXT.1.6 The TSF shall ensure that data integrity algorithms used in SSH transport connection is [selection: **hmac-sha1, hmac-sha1-96, hmac-md5, hmac-md5-96**].

FCS_SSH_EXT.1.7 The TSF shall ensure that diffie-hellman-group14-sha1 is the only allowed key exchange method used for the SSH protocol.

Rationale

This component is used to explicitly require a specific SSH implementation.

5.2.5 Explicit TLS specification (TLS)

Management: FCS_TLS_EXT.1

The following actions could be considered for the management functions in FMT:

a) Selection of TLS versions and/or ciphersuites, if available to administrators.

Audit: FCS_TLS_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Failure to establish a TLS session, the reason for failure, and the non-TOE endpoint of connection (IP address).

b) Basic: Establishment/termination of a TLS session, and the non-TOE endpoint of connection (IP address).

5.2.5.1 FCS_TLS_EXT.1 - Flexible TLS

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols: [selection: **TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)**].

FCS_TLS_EXT.1.2 The TSF shall support the following ciphersuites: [assignment: **TLS cipher suites defined in RFC standards**].

Rationale

This SFR specifies the minimal requirements for TLS. It is a more flexible implementation of the NDPP-defined component FCS_TLS_EXT.1. BIG-IP modules implement TLS tunnel functionality for a large number of infrastructure varieties that consumers may need to support, and hence, require more flexibility when it comes to protocol versions and cipher suites.

5.3 Class FFW: Firewall Rules

This class provides one family specifically concerned with the specification firewall filtering rules. The FWPP-defined class FFW_RUL_EXT provides explicit specification of firewall rules for stateful inspection.

5.3.1 Stateful Traffic Filtering (RUL)

Management: FFW_RUL_EXT.1

The following actions could be considered for the management functions in FMT:

a) Specification of the rules.

Audit: FFW_RUL_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Log all packets rejected by the rules

5.3.1.1 FFW_RUL_EXT.1 - Stateful Traffic Filtering

Hierarchical to: No other components.

Dependencies: No dependencies.

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

FFW_RUL_EXT.1.2 The TSF shall process the following network traffic protocols:

a) Internet Control Message Protocol version 4 (ICMPv4)

b) Internet Control Message Protocol version 6 (ICMPv6)

c) Internet Protocol (IPv4)

d) Internet Protocol version 6 (IPv6)

e) Transmission Control Protocol (TCP)

f) User Datagram Protocol (UDP)

and be capable of inspecting network packet header fields defined by the following RFCs to the extent mandated in the other elements of this SFR

a) [RFC792] (ICMPv4)

b) [RFC4443] (ICMPv6)

c) [RFC791] (IPv4)

d) [RFC2460] (IPv6)

e) [RFC793] (TCP)

f) [RFC768] (UDP).

FFW_RUL_EXT.1.3 The TSF shall allow the definition of Stateful Traffic Filtering rules using the following network protocol fields:

a) ICMPv4
   1. Type
   2. Code

b) ICMPv6
   1. Type
   2. Code

c) IPv4
   1. Source address
   2. Destination address
   3. Protocol

d) IPv6
   1. Source address
   2. Destination address
   3. Next Header

e) TCP
   1. Source Port
   2. Destination Port

f) UDP
   1. Source Port
   2. Destination Port

and distinct interface.

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

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

FFW_RUL_EXT.1.6 The TSF shall:

a) accept a network packet without further processing of Stateful Traffic Filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [selection: **ICMP, no other protocols**] based on the following network packet attributes:
   1: TCP: source and destination addresses, source and destination ports, sequence number, Flags;
   2: UDP: source and destination addresses, source and destination ports;
   3: [selection: **ICMP: source and destination addresses, [selection: type, code, [assignment: list of matching attributes]], no other protocols**].

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

FFW_RUL_EXT.1.7 The TSF shall be able to process the following network protocols:

1. FTP
2. [selection: **H.323, [assignment: other supported protocols], no other protocols**]

to dynamically define rules or establish sessions allowing network traffic of the following types:

a) FTP: TCP data sessions in accordance with the FTP protocol as specified in [RFC959],

b) [selection: **[assignment: list of additionally supported protocols and the types of network traffic to be allowed based on those protocols], none**]

FFW_RUL_EXT.1.8 The TSF shall enforce the following default Stateful Traffic Filtering rules on all network traffic:

1. The TSF shall reject and be capable of logging packets which are invalid fragments;
2. The TSF shall reject and be capable of logging fragmented IP packets which cannot be re-assembled completely;
3. The TSF shall reject and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;

4. The TSF shall reject and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received;
5. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a broadcast network;
6. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a multicast network;
7. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
8. The TSF shall reject and be capable of logging network packets where the source address of the network packet is a multicast;
9. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is a link-local address;
10. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as being an address “reserved for future use” as specified in [RFC5735] for IPv4;
11. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as an “unspecified address” or an address “reserved for future definition and use” as specified in [RFC4291] for IPv6;
12. The TSF shall reject and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and
13. [**[assignment: other default rules enforced by the TOE], no other rules**].

FFW_RUL_EXT.1.9 When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall process the applicable Stateful Traffic Filtering rules (as determined in accordance with FFW_RUL_EXT.1.5) in the following order: administrator-defined.

FFW_RUL_EXT.1.10 When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall deny packet flow if a matching rule is not identified.

Rationale

The FWPP-defined class FFW provides explicit specification of firewall rules. FFW_RUL_EXT.1 describes requirements for stateful packet filtering.

5.4 Class FIA: Identification and Authentication

5.4.1 Password Management (PMG)

Management: FIA_PMG_EXT.1

The following actions could be considered for the management functions in FMT:

a) Specification of password composition.

Audit: FIA_PMG_EXT.1

There are no audit events foreseen.

5.4.1.1 FIA_PMG_EXT.1 - Password Management

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_PMG_EXT.1 The TSF shall provide the following password management capabilities for administrative passwords:

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

Rationale

The NDPP-defined family PMG provides a password quality specification.

5.4.2 User Identification and Authentication (UAU)

Management: FIA_UAU_EXT.2

The following actions could be considered for the management functions in FMT:

a) Specification of password composition.

Audit: FIA_UAU_EXT.2

There are no audit events foreseen.

5.4.2.1 FIA_UAU_EXT.2 - Password-based Authentication Mechanism

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [selection: **[assignment: other authentication mechanism(s)], none**] to perform administrative user authentication.

Rationale

The NDPP-defined component FIA_UAU_EXT.2 provides the specification of authentication mechanisms while mandating at least password based authentication.

5.4.3 User Identification and Authentication (UIA)

Management: FIA_UIA_EXT.1

The following actions could be considered for the management functions in FMT:

a) Specification of the banner for FTPA_TAB.1

Audit: FIA_UIA_EXT.1

There are no audit events foreseen.

5.4.3.1 FIA_UIA_EXT.1 - User Identification and Authentication

Hierarchical to: No other components.

Dependencies: FTA_TAB.1 Default TOE access banners

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:

a) Display the warning banner in accordance with FTA_TAB.1;

b) [selection: **no other actions, [assignment: list of services, actions performed by the TSF in response to non-TOE requests]**].

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.

Rationale

The NDPP-defined component FIA_UIU_EXT.1 describes the TOE behavior before authentication.

5.5 Class FPT: Protection of the TSF

5.5.1 Protection of Administrator Passwords (APW)

Management: FPT_APW_EXT.1

There are no management activities foreseen.

Audit: FPT_APW_EXT.1

There are no audit events foreseen.

5.5.1.1 FPT_APW_EXT.1 - Protection of Administrator Passwords

Hierarchical to: No other components.

Dependencies: No dependencies.

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

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

Rationale

The NDPP-defined component FPT_APW explicitly defines requirements for protected password storage.

5.5.2 Protection of TSF Data (for reading of all symmetric keys) (SKP)

Management: FPT_SKP_EXT.1

There are no management activities foreseen.

Audit: FPT_SKP_EXT.1

There are no audit events foreseen.

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

Hierarchical to: No other components.

Dependencies: No dependencies.

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

Rationale

The NDPP-defined component FPT_SKP explicitly defines requirements for protection of symmetric keys.

5.5.3 TSF Testing (TST)

Management: FPT_TST_EXT.1

There are no management activities foreseen.

Audit: FPT_TST_EXT.1

There are no audit events foreseen.

5.5.3.1 FPT_TST_EXT.1 - TSF Testing

Hierarchical to: No other components.

Dependencies: No dependencies.

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

Rationale

The NDPP-defined component FPT_TST_EXT explicitly defines requirements for self test of the TSF.

5.5.4 Trusted Update (TUD)

Management: FPT_TUD_EXT.1

There are no management activities foreseen.

Audit: FPT_TUD_EXT.1

There are no audit events foreseen.

5.5.4.1 FPT_TUD_EXT.1 - Trusted Update

Hierarchical to: No other components.

Dependencies: No dependencies.

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

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

FPT_TUD_EXT.1.3 The TSF shall provide a means to verify firmware/software updates to the TOE using a [selection: **digital signature mechanism, published hash**] prior to installing those updates.

Rationale

The NDPP-defined component FPT_TUD explicitly defines requirements for update verification.

6 Security Requirements

6.1 TOE Security Functional Requirements

This ST identifies assignments and selections in **bold typeface.** Refinements are identified in strike-through and ~~italy~~*italics*. Iterations performed by the ST author are identified by adding a dash (-) and IDENTIFIER to the SFR reference, while iterations derived from the PPs continue to carry a sequential number in parentheses (2) added to the SFR reference.

The following table shows the SFRs for the TOE, and the operations performed on the components according to CC part 2: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
Security audit	FAU_GEN.1 Audit Data Generation		CC Part 2	No	Yes	Yes	Yes
	FAU_GEN.2 User Identity Association		CC Part 2	No	No	No	No
	FAU_STG_EXT.1 External Audit Trail Storage		ECD	No	No	No	Yes
Cryptographic support	FCS_CKM.1 Cryptographic Key Generation (SSH host key)		CC Part 2	Yes	No	Yes	No
	FCS_CKM.1-RSA Cryptographic Key Generation (for asymmetric keys)	FCS_CKM.1	CC Part 2	Yes	Yes	Yes	Yes
	FCS_CKM_EXT.4 Cryptographic Key Zeroization		ECD	No	No	No	No
	FCS_COP.1(1) Cryptographic Operation (for cryptographic signature)	FCS_COP.1	CC Part 2	Yes	No	Yes	No
	FCS_COP.1(2) Cryptographic Operation (for cryptographic hashing)	FCS_COP.1	CC Part 2	Yes	Yes	Yes	No
	FCS_COP.1(3) Cryptographic Operation (for keyed-hash message authentication)	FCS_COP.1	CC Part 2	Yes	Yes	Yes	No
	FCS_HTTPS_EXT.1 Explicit: HTTPS		ECD	No	Yes	No	No
	FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)		ECD	No	Yes	No	Yes
	FCS_SSH_EXT.1 Explicit: SSH		ECD	No	Yes	Yes	Yes
	FCS_TLS_EXT.1 Traffic TLS		ECD	No	Yes	Yes	Yes
User data protection	FDP_ACC.1 Subset access control		CC Part 2	No	No	Yes	No

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FDP_ACF.1 Security attribute based access control		CC Part 2	No	No	Yes	No
	FDP_ITC.1 Import of user data without security attributes		CC Part 2	No	No	Yes	No
	FDP_RIP.2 Full Residual Information Protection		CC Part 2	No	No	No	Yes
	FDP_UCT.1 Inter-TSF user data confidentiality transfer protection		CC Part 2	No	No	Yes	Yes
	FDP_UIT.1 Inter-TSF user data integrity transfer protection		CC Part 2	No	No	Yes	Yes
Firewall rules	FFW_RUL_EXT.1 Stateful Traffic Filtering		ECD	No	Yes	No	Yes
Identification and authentication	FIA_AFL.1 Authentication failure handling		CC Part 2	No	No	Yes	Yes
	FIA_ATD.1 User attribute definition		CC Part 2	No	Yes	Yes	No
	FIA_PMG_EXT.1 Password Management		ECD	No	Yes	Yes	Yes
	FIA_UAU_EXT.2 Password-based Authentication Mechanism		ECD	No	No	No	Yes
	FIA_UAU.5 Traffic authentication mechanisms		CC Part 2	No	Yes	Yes	No
	FIA_UAU.7 Protected authentication feedback		CC Part 2	No	Yes	Yes	No
	FIA_UIA_EXT.1 User Identification and Authentication		ECD	No	No	No	Yes
Security management	FMT_MSA.1 Management of security attributes		CC Part 2	No	No	Yes	Yes
	FMT_MSA.3 Static attribute initialisation		CC Part 2	No	No	Yes	Yes
	FMT_MTD.1 Management of TSF data (for general TSF data)		CC Part 2	No	No	Yes	Yes
	FMT_SMF.1 Specification of Management Functions		CC Part 2	No	No	Yes	No
	FMT_SMR.1 Security roles		CC Part 2	No	No	Yes	No

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
Protection of the TSF	FPT_APW_EXT.1 Protection of Administrator Passwords		ECD	No	No	No	No
	FPT_FLS.1 Failure with preservation of secure state		CC Part 2	No	No	Yes	No
	FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)		ECD	No	No	No	No
	FPT_TST_EXT.1 TSF testing		ECD	No	No	No	No
	FPT_TUD_EXT.1 Extended: Trusted Update		ECD	No	No	No	Yes
Resource utilisation	FRU_RSA.1 Maximum Quotas		CC Part 2	No	No	Yes	Yes
TOE access	FTA_SSL.3 TSF-initiated Termination		CC Part 2	No	Yes	Yes	No
	FTA_SSL.4 User-initiated termination		CC Part 2	No	Yes	No	No
	FTA_TAB.1 Default TOE Access Banners		CC Part 2	No	Yes	No	No
Trusted path/channel of the TSF	FTP_ITC.1 Inter-TSF trusted channel		CC Part 2	No	No	Yes	Yes
	FTP_TRP.1 Trusted Path		CC Part 2	No	Yes	Yes	Yes

Table 7: SFRs for the TOE

6.1.1 Security audit

6.1.1.1 Audit Data Generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the **not specified** level of audit; and

c) **All administrative actions;**

d) **Specifically defined auditable events listed in Table 8**.

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 ~~(if applicable)~~, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **information specified in column three of Table 8**.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	
FAU_GEN.2	None.	
FAU_STG_EXT.1	None.	
FCS_CKM.1	None.	
FCS_CKM.1-RSA	None.	
FCS_CKM_EXT.4	None.	
FCS_COP.1(1)	None.	
FCS_COP.1(2)	None.	
FCS_COP.1(3)	None.	
FCS_HTTPS_EXT.1	Failure to establish a HTTPS session. Establishment/Termination of a HTTPS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_RBG_EXT.1	None.	
FCS_SSH_EXT.1	Failure to establish an SSH session. Establishment/Termination of an SSH session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_TLS_EXT.1	Failure to establish a TLS session. Establishment/Termination of a TLS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FDP_ACC.1	*None.*	
FDP_ACF.1	*All requests to perform an operation on an object covered by the SFP.*	*No additional information.*
FDP_RIP.2	None.	
FDP_UCT.1	Any use of data exchange mechanisms (truseted channel/path).	Identification of the user of data exchange mechanism.
FDP_UIT.1	Any use of data exchange mechanisms (truseted channel/path).	Identification of the user of data exchange mechanism.
FFW_RUL_EXT.1	Application of rules configured with the ‘log’ operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE Interface
	Indication of packets dropped due to too much network traffic	TOE interface that is unable to process packets

Requirement	Auditable Events	Additional Audit Record Contents
FIA_AFL.1	*The reaching of the threshold for the unsuccessful authentication attempts and the actions (e.g. disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of a terminal).*	*No additional information.*
FIA_ATD.1	*None.*	
FIA_PMG_EXT.1	None.	
FIA_UAU_EXT.2	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.5	*None.*	
FIA_UAU.7	None.	
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FMT_MSA.1	*None.*	
FMT_MSA.3	*None.*	
FMT_MTD.1	None.	
FMT_SMF.1	~~None.~~*Use of the management functions.*	*No additional information.*
FMT_SMR.1	None.	
FPT_APW_EXT.1	None.	
FPT_FLS.1	*Failure of the TSF.*	*No additional information.*
FPT_SKP_EXT.1	None.	
FPT_TST_EXT.1	None.	
FPT_TUD_EXT.1	Initiation of update.	No additional information.
FRU_RSA.1	*None.*	
FTA_SSL.3	The termination of a interactive session by the session locking mechanism.	No additional information.
FTA_SSL.4	The termination of an interactive session.	No additional information.
FTA_TAB.1	None.	
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.

Requirement	Auditable Events	Additional Audit Record Contents
FTP_TRP.1	Initiation of the trusted ~~channel~~*path functions*. Termination of the trusted ~~channel~~*path functions*. Failures of the trusted path functions.	Identification of the claimed user identity.

Table 8: Auditable Events

ST Application Note: *Changes / additions to the auditable events and audit record contents derived from NDPP and FWPP have been marked in italics.*

6.1.1.2 User Identity Association (FAU_GEN.2)

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

6.1.1.3 External Audit Trail Storage (FAU_STG_EXT.1)

FAU_STG_EXT.1.1 The TSF shall be able to **transmit the generated audit data to an external IT entity** using a trusted channel implementing the **TLS** protocol.

6.1.2 Cryptographic support

6.1.2.1 Cryptographic Key Generation (SSH host key) (FCS_CKM.1)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **RSA key generation using probable primes with a specified random number generator FCS_RBG_EXT.1** and specified cryptographic key sizes **1024-bit or higher RSA key sizes** that meet the following: **FIPS 186-2, A.2.1 for Miller Rabin probabalistic primality tests**.

ST Application Note: *The key is generated when the SSH server is first started.*

6.1.2.2 Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1-RSA)

FCS_CKM.1.1-RSA The TSF shall generate *asymmetric* cryptographic keys *used for key establishment* in accordance with ~~a specified cryptographic key generation algorithm~~

a) NIST Special Publication 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” for finite field-based key establishment schemes;

b) NIST Special Publication 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” for elliptic curve-based key establishment schemes and implementing “NIST curves” P-256 (as defined in FIPS PUB 186-3, “Digital Signature Standard”)

c) NIST Special Publication 800-56B, “Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography” for RSA-based key establishment schemes

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

Application Note: *Item a applies only to SSH, and items b and c only to TLS. Note that for item c, the RSA key generation is done as part of the TOE environment.*

6.1.2.3 Cryptographic Key Zeroization (FCS_CKM_EXT.4)

FCS_CKM_EXT.4.1 The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

NDPP Application Note: "*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.*"

NDPP Application Note: *The zeroization indicated above applies to each intermediate storage area for plaintext key/cryptographic critical security parameter (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/cryptographic critical security parameter to another location.*

6.1.2.4 Cryptographic Operation (for cryptographic signature) (FCS_COP.1(1))

FCS_COP.1.1(1) The TSF shall perform **cryptographic signature services** in accordance with a specified cryptographic algorithm

a) RSA Digital Signature Algorithm (rDSA)

b) Elliptic Curve Digital Signature Algorithm (ECDSA)

and cryptographic key sizes

a) (modulus) of 2048 bits or greater,

b) 256 bits or greater]

that meet the following:

a) RSA: FIPS PUB 186-3, “Digital Signature Standard”

b) ECDSA NIST curve P-256 (as defined in FIPS PUB 186-3, “Digital Signature Standard”).

6.1.2.5 Cryptographic Operation (for cryptographic hashing) (FCS_COP.1(2))

FCS_COP.1.1(2) The TSF shall perform **cryptographic hashing services** in accordance with a specified cryptographic algorithm **SHA-1, SHA-256, SHA-384** and ~~cryptographic key~~*messages digest* sizes **160, 256, 384 bits** that meet the following: **FIPS Pub 180-3, "Secure Hash Standard" [FIPSPUB180-3]**.

6.1.2.6 Cryptographic Operation (for keyed-hash message authentication) (FCS_COP.1(3))

FCS_COP.1.1(3) 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 size **160, 256, 384 bits**, and *message digest sizes 160 bits* that meet the following: **FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code" [FIPSPUB198-1], and FIPS Pub 180-3, "Secure Hash Standard" [FIPSPUB180-3]**.

ST Application Note: *This SFR applies to both the hashes generated in TLS (cf.* [RFC4346] *section 6.3) and in SSH (cf.* [RFC4253] *section 6.4).*

6.1.2.7 Explicit: HTTPS (FCS_HTTPS_EXT.1)

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with [RFC2818].

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS as specified in FCS_TLS_EXT.1.

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

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with **NIST Special Publication ~~800-90~~[NIST800-90A] using CTR_DRBG (AES)** seeded by an entropy source that accumulated entropy from **a software-based noise source**.

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of **256 bits** of entropy at least equal to the greatest bit length of the keys and authorization factors that it will generate.

ST Application Note: *The TOE actually uses entropy that is originally generated by hardware in a crypto module provided by the underlying operational environment. However, it is acquired from that module via a software interface and further processed by TSF software, making the selection performed more accurate than selecting "a TSF-hardware-based noise source" as offered by NDPP.*

ST Application Note:

This application note rephrases the RNG requirement using AIS20 [AIS_20]*.*

FCS_RNG.1.1 The TSF shall provide a deterministic random number generator using the CTR_DRBG with AES 256 bit core specified in [NIST800-90A] *that implements:*

a) DRG2.1: If initialized with a random seed using /dev/random as random source, the internal state of the RNG shall have a minimum entropy of 48 bits.

b) DRG2.2: The DRNG provides forward secrecy.

c) DRG2.3: The DRNG provides backward secrecy.

FCS_RNG.1.2 The TSF shall provide random numbers that meet:

*a) DRG.2.4: The RNG initialized with a random seed holding 96 bits of entropy generates output for which 2**25 strings of length 80 bits are mutually different with probability of less than 1-2**-10.*

b) DRG.2.5: The test suite A and no other test suite cannot distinguish the random numbers from output sequences of ideal RNGs.

6.1.2.9 Explicit: SSH (FCS_SSH_EXT.1)

FCS_SSH_EXT.1.1 The TSF shall implement the SSH protocol that complies with ~~RFCs 4251, 4252, 4253, and 4254~~[RFC4251], [RFC4252], [RFC4253], *and* [RFC4254].

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in [RFC4252]: public-key based, password-based.

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in [RFC4253], packets greater than **(256*1024)** bytes in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: ~~AES-CBC-128,~~ AES-CBC-256 **no other algorithms**.

FCS_SSH_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses SSH_RSA and **no other public key algorithms** as its public key algorithm(s).

FCS_SSH_EXT.1.6 The TSF shall ensure that data integrity algorithms used in SSH transport connection is **hmac-sha1**.

FCS_SSH_EXT.1.7 The TSF shall ensure that diffie-hellman-group14-sha1 is the only allowed key exchange method used for the SSH protocol.

6.1.2.10 Traffic TLS (FCS_TLS_EXT.1)

FCS_TLS_EXT.1.1 The TSF shall implement the following protocols: **TLS 1.1 ([RFC4346]), TLS 1.2 ([RFC5246])**.

ST Application Note: *While TLS 1.0 has been removed from the security claims, the user still can select it, subject to the warnings in the guidance.*

FCS_TLS_EXT.1.2 The TSF shall support the following ciphersuites:

1. **TLS_RSA_WITH_AES_128_CBC_SHA**
2. **TLS_RSA_WITH_AES_256_CBC_SHA**
3. **TLS_RSA_WITH_AES_128_CBC_SHA256**
4. **TLS_RSA_WITH_AES_256_CBC_SHA256**
5. **TLS_RSA_WITH_AES_128_GCM_SHA256**
6. **TLS_RSA_WITH_AES_256_GCM_SHA384**
7. **TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA**
8. **TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA**
9. **TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA**
10. **TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA**
11. **TLS_ECDH_RSA_WITH_AES_128_CBC_SHA**
12. **TLS_ECDH_RSA_WITH_AES_256_CBC_SHA**

13. **TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA**
14. **TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA**
15. **TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256**
16. **TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384**
17. **TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256**
18. **TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384**
19. **TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256**
20. **TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384**
21. **TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256**
22. **TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384**
23. **TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256**
24. **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384**
25. **TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256**
26. **TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384**
27. **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256**
28. **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384**
29. **TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256**
30. **TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384**

6.1.3 User data protection

6.1.3.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the **Administrator Access Control Policy** on

- **a) subjects: administrative users**
- **b) objects: configuration objects**
- **c) operations among subjects and objects:**
  1. **write (create, modify, enable, disable, delete, view)**
  2. **update (modify, enable, disable, view)**
  3. **enable/disable (enable, disable, view)**
  4. **read (view)**

.

6.1.3.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the **Administrator Access Control Policy** to objects based on the following:

- **a) subjects: user with the following attributes:**
  1. **role**
  2. **partition access**
- **b) objects: partition with the following attributes:**
  1. **partition type common or specific**

.

ST Application Note: *Partitions are either common or specific. Access to the common partition is determined by the administrative role only, while access to specific partitions are determined by the partition access right for each specific user.*

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) the operation a subject requests on an object will be performed:

1. if the object is in a partition that matches the subject's partition access; and

2. if the type of operation matches the type of operations granted to the subject by its role for the type of object the operation is requested on (as defined for individual roles in the TSS in table 13).

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:

a) any operations requested by subjects with the Administrator role are always performed

b) any operations requested by subjects with the User Manager role on objects in the Common partition, other than user account objects associated with the Administrator role, are always performed

c) read operations requested by subjects with a role other than No Access on objects in the Common partition, are always performed

.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **none**.

6.1.3.3 Import of user data without security attributes (FDP_ITC.1)

FDP_ITC.1.1 The TSF shall enforce the **Administrator Access Control Policy** when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **no additional importation control rules**.

ST Application Note: *This SFR addresses the import of certificates for use in traffic TLS connections. Only administrative users with the role of Administrator or Certificate Manager can import TLS certificates.*

6.1.3.4 Full Residual Information Protection (FDP_RIP.2)

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the **allocation of the resource to** all objects.

NDPP Application Note: *"Resources" in the context of this requirement are network packets being sent through (as opposed to "to", as is the case when a security administrator connects to the TOE) the TOE. The concern is that once a network packet is sent, the buffer or memory area used by the packet still contains data from that packet, and that if that buffer is re-used, those data might remain and make their way into a new packet.*

6.1.3.5 Inter-TSF user data confidentiality transfer protection (FDP_UCT.1)

FDP_UCT.1.1 The TSF shall enforce the **Administrator Access Control Policy** to **transmit, receive** user data in a manner protected from unauthorised disclosure.

ST Application Note: *This SFR applies to the SSH access.*

6.1.3.6 Inter-TSF user data integrity transfer protection (FDP_UIT.1)

FDP_UIT.1.1 The TSF shall enforce the **Administrator Access Control Policy** to **transmit, receive** user data in a manner protected from **modification, deletion, insertion, replay** errors.

FDP_UIT.1.2 The TSF shall be able to determine on receipt of user data, whether **modification, deletion, insertion, replay** has occurred.

ST Application Note:: *This SFR applies to the SSH access.*

6.1.4 Firewall rules

6.1.4.1 Stateful Traffic Filtering (FFW_RUL_EXT.1)

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

FFW_RUL_EXT.1.2 The TSF shall process the following network traffic protocols:

a) Internet Control Message Protocol version 4 (ICMPv4)

b) Internet Control Message Protocol version 6 (ICMPv6)

c) Internet Protocol (IPv4)

d) Internet Protocol version 6 (IPv6)

e) Transmission Control Protocol (TCP)

f) User Datagram Protocol (UDP)

and be capable of inspecting network packet header fields defined by the following RFCs to the extent mandated in the other elements of this SFR

a) [RFC792] (ICMPv4)

b) [RFC4443] (ICMPv6)

c) [RFC791] (IPv4)

d) [RFC2460] (IPv6)

e) [RFC793] (TCP)

f) [RFC768] (UDP).

FFW_RUL_EXT.1.3 The TSF shall allow the definition of Stateful Traffic Filtering rules using the following network protocol fields:

a) ICMPv4

1. Type
2. Code

b) ICMPv6
1. Type
2. Code

c) IPv4
1. Source address
2. Destination address
3. Protocol

d) IPv6
1. Source address
2. Destination address
3. Next Header

e) TCP
1. Source Port
2. Destination Port

f) UDP
1. Source Port
2. Destination Port

and distinct interface.

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

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

FFW_RUL_EXT.1.6 The TSF shall:

a) accept a network packet without further processing of Stateful Traffic Filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, **ICMP** based on the following network packet attributes:
1. TCP: source and destination addresses, source and destination ports, sequence number, Flags;
2. UDP: source and destination addresses, source and destination ports;
3. **ICMP: source and destination addresses, type, code**.

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

FFW_RUL_EXT.1.7 The TSF shall be able to process the following network protocols:

1. FTP
2. **no other protocols**

to dynamically define rules or establish sessions allowing network traffic of the following types:

a) FTP: TCP data sessions in accordance with the FTP protocol as specified in [RFC959],

b) **none**

FFW_RUL_EXT.1.8 The TSF shall enforce the following default Stateful Traffic Filtering rules on all network traffic:

1. The TSF shall reject and be capable of logging packets which are invalid fragments;
2. The TSF shall reject and be capable of logging fragmented IP packets which cannot be re-assembled completely;
3. The TSF shall reject and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;
4. The TSF shall reject and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received;
5. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a broadcast network;
6. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a multicast network;
7. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
8. The TSF shall reject and be capable of logging network packets where the source address of the network packet is a multicast;
9. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is a link-local address;
10. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as being an address “reserved for future use” as specified in [RFC5735] for IPv4;
11. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as an “unspecified address” or an address “reserved for future definition and use” as specified in ~~RFC 3513~~[RFC4291] for IPv6;
12. The TSF shall reject and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and
13. **no other rules**.

FFW_RUL_EXT.1.9 When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall process the applicable Stateful Traffic Filtering rules (as determined in accordance with FFW_RUL_EXT.1.5) in the following order: administrator-defined.

FFW_RUL_EXT.1.10 When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall deny packet flow if a matching rule is not identified.

6.1.5 Identification and authentication

6.1.5.1 Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when **an administrator configurable positive integer within 1 - 10** unsuccessful authentication attempts occur related to **password-based authentication of an individual administrative user account through any of the administrative interfaces**.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been **met**, the TSF shall **disable the user account**.

ST Application Note: *Only an administrative users with the role of Administrator can specify the number of unsuccessful authentication attempts.*

6.1.5.2 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual *administrative* users:

- **user name**
- **password**
- **role**
- **partition access**
- **terminal access**
- **locked**
- **password expiration date.**

6.1.5.3 Password Management (FIA_PMG_EXT.1)

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 special characters: **any character from the following:**

- **`~!@#$%^&*()-_+=[]{};':",./<>?|**

b) Minimum password length shall be settable by ~~the Security Administrator~~*authorized administrators*, and support passwords of 15 characters or greater;

ST Application Note: *Only an administrative users with the role of Administrator can specify the minimum password length.*

6.1.5.4 Password-based Authentication Mechanism (FIA_UAU_EXT.2)

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

6.1.5.5 Traffic authentication mechanisms (FIA_UAU.5)

FIA_UAU.5.1 The TSF shall provide **the implementation of: TLS (client and server) certificate validation** to support *remote* user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the **rules defined in administrator-specified SSL Client profiles, or SSL Server profiles that have been associated with virtual servers**.

Application Note: *Authentication relies on X.509 certificates that are validated according to* [RFC5280]. *The certificates are provided the TOE environment and access is given by the LDAP server that is part of the TOE environment. See OE.LDAP.*

6.1.5.6 Protected authentication feedback (FIA_UAU.7)

FIA_UAU.7.1 The TSF shall provide only **obscured feedback** to the *administrative* user while the authentication is in progress.

6.1.5.7 User Identification and Authentication (FIA_UIA_EXT.1)

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

a) Display the warning banner in accordance with FTA_TAB.1;

b) **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 on that administrative user.

Application Note: *This requirement applies to users (administrators and external IT entities) of services available from the TOE directly, and not services available by connecting through the TOE.*

6.1.6 Security management

6.1.6.1 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the **Administrator Access Control Policy** to restrict the ability to **change_default , query , modify , delete** the security attributes **for the Administrator Access Control Policy** to **authorized administrators**.

ST Application Note: *Changes to the security attributes is restricted to administrative users with the role Administrator and User Administrator.*

6.1.6.2 Static attribute initialisation (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the **Administrator Access Control Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **authorized administrators** to specify alternative initial values to override the default values when an object or information is created.

6.1.6.3 Management of TSF data (for general TSF data) (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to **manage** the **TSF data** to **the authorized administrators**.

ST Application Note: *TSF data in the context of this SFR is the configuration data of the TOE. The TSF data that can be managed depends on the role of the administrative user. See table 13.*

6.1.6.4 Specification of Management Functions (FMT_SMF.1)

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

a) Ability to administer the TOE locally and remotely;

b) Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;

c) Ability to configure the cryptographic functionality;

d) Configure Firewall rules.

ST Application Note: *The ability to administer the TOE applies to all administrator roles. The ability to update the TOE is limited to administrative users with the role of Administrator, as described in FPT_TUD_EXT.1. The ability to configure the cryptographic functionality is restricted to administrative users with the Administrator role, however the Certificate Manager can also manage certificates. The ability to configure Firewall rules is limited to administrative users with the role of Administrator and Firewall Administrator.*

6.1.6.5 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles: **Authorized Administrator represented by the following**

- **Administrator**
- **Resource Administrator**
- **User Manager**
- **Manager**
- **Certificate Manager**
- **iRule Manager**
- **Application Editor**
- **Operator**
- **Auditor**
- **Guest**
- **Firewall Manager**

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

ST Application Note: *There is an additional role "No Access" that can be assigned to users, with the single purpose of preventing them to access the TOE. Assigning users this role is a way to temporarily disable them from using the TOE, but still maintaining the account.*

6.1.7 Protection of the TSF

6.1.7.1 Protection of Administrator Passwords (FPT_APW_EXT.1)

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

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

NDPP Application Note: *The intent of the requirement is that raw password authentication data are not stored in the clear, and that no user or administrator is able to read the plaintext password through "normal" interfaces. An all-powerful administrator of course could directly read memory to capture a password but is trusted not to do so.*

6.1.7.2 Failure with preservation of secure state (FPT_FLS.1)

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur:

1. **The active instance reports a failover condition to the standby instance, causing the standby instance to become active.**
2. **The standby instance determines that the active instance is not available anymore, causing the standby instance to become active.**

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

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

NDPP Application Note: *The intent of the requirement is that an administrator is unable to read or view the identified keys (stored or ephemeral) through "normal" interfaces. While it is understood that the administrator could directly read memory to view these keys, to do so is not a trivial task and may require substantial work on the part of an administrator. Since the administrator is considered a trusted agent, it is assumed they would not endeavor in such an activity.*

6.1.7.4 TSF testing (FPT_TST_EXT.1)

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

6.1.7.5 Extended: Trusted Update (FPT_TUD_EXT.1)

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

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

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

ST Application Note: *The ability to initiate updates to the TOE is limited to the administrative users with the role of Administrator.*

6.1.8 Resource utilisation

6.1.8.1 Maximum Quotas (FRU_RSA.1)

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: **duration of TCP and UDP connections** that **subjects** can use **simultaneously**.

6.1.9 TOE access

6.1.9.1 TSF-initiated Termination (FTA_SSL.3)

FTA_SSL.3.1 The TSF shall terminate an interactive session after a *authorized administrator configurable* **time interval of session inactivity**.

ST Application Note: *The ability to configure the time interval of session inactivity is limited to the administrative users with the role of Administrator.*

6.1.9.2 User-initiated termination (FTA_SSL.4)

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

6.1.9.3 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1 Before establishing an *administrative* user session the TSF shall display a *authorized administrator-specified* advisory *notice and consent* warning message regarding ~~unauthorized~~use of the TOE.

NDPP Application Note: *This requirement is intended to apply to interactive sessions between a human user and a TOE. IT entities establishing connections or programmatic connections (e.g., remote procedure calls over a network) are not required to be covered by this requirement.*

ST Application Note: *The ability to specify the banner is limited to the administrative users with the role of Administrator.*

6.1.10 Trusted path/channel of the TSF

6.1.10.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification and disclosure.

FTP_ITC.1.2 The TSF shall permit **the TSF, another trusted IT product** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **HTTPS connections to web servers and for transmission of syslog records to syslog servers**.

ST Application Note: *This FTP_ITC.1 addresses trusted channels implemented by the TOE for HTTPS connections to web servers and syslog servers. For connection to web servers, the TOE can act both as a client and a server for the trusted channel.*

6.1.10.2 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1 The TSF shall *use SSH for tmsh to* provide a *trusted* communication path between itself and **remote** ~~users~~*administrators* that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **disclosure, and detection of modification of the communicated data**.

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

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

6.2 Security Functional Requirements Rationale

6.2.1 Coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective.

Security functional requirements	Objectives
FAU_GEN.1	O.SYSTEM_MONITORING
FAU_GEN.2	O.SYSTEM_MONITORING
FAU_STG_EXT.1	O.SYSTEM_MONITORING
FCS_CKM.1	O.PROTECTED_COMMUNICATIONS
FCS_CKM.1-RSA	O.PROTECTED_COMMUNICATIONS
FCS_CKM_EXT.4	O.LTM-TRAFFICMGMT, O.PROTECTED_COMMUNICATIONS
FCS_COP.1(1)	O.LTM-TRAFFICMGMT, O.PROTECTED_COMMUNICATIONS
FCS_COP.1(2)	O.LTM-TRAFFICMGMT, O.PROTECTED_COMMUNICATIONS
FCS_COP.1(3)	O.LTM-TRAFFICMGMT, O.PROTECTED_COMMUNICATIONS
FCS_HTTPS_EXT.1	O.PROTECTED_COMMUNICATIONS
FCS_RBG_EXT.1	O.LTM-TRAFFICMGMT, O.PROTECTED_COMMUNICATIONS
FCS_SSH_EXT.1	O.PROTECTED_COMMUNICATIONS
FCS_TLS_EXT.1	O.LTM-TRAFFICMGMT, O.PROTECTED_COMMUNICATIONS
FDP_ACC.1	O.TOE_ADMINISTRATION

Security functional requirements	Objectives
FDP_ACF.1	O.TOE_ADMINISTRATION
FDP_ITC.1	O.LTM-TRAFFICMGMT
FDP_RIP.2	O.RESIDUAL_INFORMATION_CLEARING
FDP_UCT.1	O.PROTECTED_COMMUNICATIONS
FDP_UIT.1	O.PROTECTED_COMMUNICATIONS
FFW_RUL_EXT.1	O.ADDRESS_FILTERING, O.PORT_FILTERING, O.RELATED_CONNECTION_FILTERING, O.STATEFUL_INSPECTION
FIA_AFL.1	O.TOE_ADMINISTRATION
FIA_ATD.1	O.TOE_ADMINISTRATION
FIA_PMG_EXT.1	O.TOE_ADMINISTRATION
FIA_UAU_EXT.2	O.TOE_ADMINISTRATION
FIA_UAU.5	O.LTM-TRAFFICMGMT
FIA_UAU.7	O.TOE_ADMINISTRATION
FIA_UIA_EXT.1	O.TOE_ADMINISTRATION
FMT_MSA.1	O.TOE_ADMINISTRATION
FMT_MSA.3	O.TOE_ADMINISTRATION
FMT_MTD.1	O.TOE_ADMINISTRATION
FMT_SMF.1	O.TOE_ADMINISTRATION
FMT_SMR.1	O.TOE_ADMINISTRATION
FPT_APW_EXT.1	O.TOE_ADMINISTRATION
FPT_FLS.1	O.FAILOVER
FPT_SKP_EXT.1	O.TOE_ADMINISTRATION
FPT_TST_EXT.1	O.TSF_SELF_TEST
FPT_TUD_EXT.1	O.VERIFIABLE_UPDATES
FRU_RSA.1	O.RESOURCE_AVAILABILITY
FTA_SSL.3	O.SESSION_LOCK
FTA_SSL.4	O.SESSION_LOCK
FTA_TAB.1	O.DISPLAY_BANNER

Security functional requirements	Objectives
FTP_ITC.1	O.LTM-TRAFFICMGMT, O.PROTECTED_COMMUNICATIONS
FTP_TRP.1	O.PROTECTED_COMMUNICATIONS

Table 9: Mapping of security functional requirements to security objectives

6.2.2 Sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives.

Security objectives	Rationale
O.ADDRESS_FILTERING	FFW_RUL_EXT.1 spells out the requirement to implement filters and logs as desired in O.ADDRESS_FILTERING.
O.DISPLAY_BANNER	The requirement for banners can be found in FTA_TAB.1.
O.FAILOVER	Failover capabilities are required in FPT_FLS.1 (preservation of a secure state if one instance of the TOE fails).
O.LTM-TRAFFICMGMT	The requirement for LTM to authenticate network traffic is reflected in FIA_UAU.5 In order to proxy HTTPS traffic, a requirement to implement TLS is spelled out on a high level in FTP_ITC.1 and implemented in FCS_TLS_EXT.1. Cryptographic support (key management and primitives) for the latter is spelled out in FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_RBG_EXT.1, and FDP_ITC.1.
O.PORT_FILTERING	FFW_RUL_EXT.1 spells out the requirement to implement filters and logs as desired in O.PORT_FILTERING.
O.PROTECTED_COMMUNICATIONS	Requirements for the provision of HTTPS (FCS_HTTPS_EXT.1), SSH (FCS_SSH_EXT.1), and TLS FCS_TLS_EXT.1) implement FTP_ITC.1, FTP_ITC.1, and FTP_TRP.1. Cryptographic support for these protocols is implemented in FCS_CKM.1-RSA, FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), and FCS_RBG_EXT.1. This also includes FDP_UCT.1 and FDP_UIT.1, which apply to administrative interfaces using SSH.
O.RELATED_CONNECTION_FILTERING	FFW_RUL_EXT.1 implements the requirement for support of connections related to the same protocol in O.RELATED_CONNECTION_FILTERING.
O.RESIDUAL_INFORMATION_CLEARING	This objective is addressed in FDP_RIP.2, which requires that the TOE ensures that residual information in memory does not leak into newly created network packets.

Security objectives	Rationale
O.RESOURCE_AVAILABILITY	FRU_RSA.1 spells out resources for which the TOE is required to implement protection against exhaustion in order to uphold its manageability at any time.
O.SESSION_LOCK	This is addressed in SFRs FTA_SSL.3 requiring termination of interactive sessions after a period of inactivity and FTA_SSL.4 by allowing users to actively terminate (end) a session.
O.STATEFUL_INSPECTION	For stateful protocols, FFW_RUL_EXT.1 requires the implementation of stateful packet inspection.
O.SYSTEM_MONITORING	FAU_GEN.1 defines the types of audit records that the TOE must be able to generate. FAU_GEN.2 requires that audit records be associated with user identities, where possible. FAU_STG_EXT.1 requires that the TOE shall be able to send audit records to an external log server. OE.TIME supports the generation of audit records by providing a reliable time stamp.
O.TOE_ADMINISTRATION	Authentication of administrators is defined in FIA_UIA_EXT.1, FIA_UAU_EXT.2, and FIA_UAU.7. This includes authentication failure handling (FIA_AFL.1), password management capabilities (FIA_PMG_EXT.1). User attributes needed for authentication and other security functions are enumerated in FIA_ATD.1. In oder to support the administration of the TSF (FMT_SMF.1), FDP_ACC.1 and FDP_ACF.1 define the "Administrator Access Control Policy" that implements the requirement in FMT_MTD.1, with additional properties defined in FMT_MSA.1, FMT_MSA.3, and FMT_SMR.1. FPT_APW_EXT.1 and FPT_SKP_EXT.1 spell out specific protections for plaintext passwords and cryptographic key material.
O.TSF_SELF_TEST	FPT_TST_EXT.1 spells out the self-test requirements for the TOE.
O.VERIFIABLE_UPDATES	FPT_TUD_EXT.1 requires the TOE to implement verifiable update capabilities.

Table 10: Security objectives for the TOE rationale

6.2.3 Security Requirements Dependency Analysis

Dependencies within the EAL package selected for the security assurance requirements have been considered by the authors of CC Part 3 and are not analyzed here again. The included component on flaw remediation, ALC_FLR.3, has no dependencies on other requirements.

The security functional requirements in this Security Target do not introduce dependencies on any security assurance requirement; neither do the security assurance requirements in this Security Target introduce dependencies on any security functional requirement.

The following table demonstrates the dependencies of SFRs modeled in CC Part 2 and how the SFRs for the TOE resolve those dependencies.

Security functional requirement	Dependencies	Resolution
FAU_GEN.1	FPT_STM.1	The TOE itself does not generate reliable time stamps but relies on time stamps provided by the TOE environment (OE.TIME).
FAU_GEN.2	FAU_GEN.1	FAU_GEN.1
	FIA_UID.1	FIA_UIA_EXT.1
FAU_STG_EXT.1	FAU_GEN.1	FAU_GEN.1
FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1]	The keys are not distributed since they are used locally by the SSH host, rather than relying on stand-alone key distribution mechanisms defined in a dependent SFR because the distribution of keys is inherent to the SSH protocol the TOE is required to support. Therefore, the dependency on FCS_CKM.2 is not necessary. FCS_SSH_EXT.1
	FCS_CKM.4	FCS_CKM_EXT.4
FCS_CKM.1-RSA	[FCS_CKM.2 or FCS_COP.1]	The SFR has been iterated to specify the key distribution method rather than relying on stand-alone key distribution mechanisms defined in a dependent SFR because the distribution of keys is inherent to the protocols the TOE is required to support. Therefore, the dependency on FCS_CKM.2 is not necessary. FCS_COP.1(1)
	FCS_CKM.4	FCS_CKM_EXT.4
FCS_CKM_EXT.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FDP_ITC.1
FCS_COP.1(1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FDP_ITC.1 FCS_CKM.1-RSA
	FCS_CKM.4	FCS_CKM_EXT.4
FCS_COP.1(2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	Hash mechanisms do not require keys.
	FCS_CKM.4	FCS_CKM_EXT.4

Security functional requirement	Dependencies	Resolution
FCS_COP.1(3)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	The TOE does not (necessarily) generate keys used as secret for the HMAC because the sources for the necessary keys are defined by the protocols supported by the TSF; therefore, the dependency on FCS_CKM.1 is not necessary.
	FCS_CKM.4	FCS_CKM_EXT.4
FCS_HTTPS_EXT.1	FCS_TLS_EXT.1	FCS_TLS_EXT.1
FCS_RBG_EXT.1	No dependencies.	
FCS_SSH_EXT.1	No dependencies.	
FCS_TLS_EXT.1	No dependencies.	
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1	FDP_ACC.1
	FMT_MSA.3	FMT_MSA.3
FDP_ITC.1	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1
	FMT_MSA.3	FMT_MSA.3
FDP_RIP.2	No dependencies.	
FDP_UCT.1	[FTP_ITC.1 or FTP_TRP.1]	FTP_TRP.1
	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1
FDP_UIT.1	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1
	[FTP_ITC.1 or FTP_TRP.1]	FTP_TRP.1
FFW_RUL_EXT.1	No dependencies.	
FIA_AFL.1	FIA_UAU.1	FIA_UIA_EXT.1
FIA_ATD.1	No dependencies.	
FIA_PMG_EXT.1	No dependencies.	
FIA_UAU_EXT.2	No dependencies.	
FIA_UAU.5	No dependencies.	
FIA_UAU.7	FIA_UAU.1	FIA_UIA_EXT.1
FIA_UIA_EXT.1	FTA_TAB.1	FTA_TAB.1

Security functional requirement	Dependencies	Resolution
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3	FMT_MSA.1	FMT_MSA.1
	FMT_SMR.1	FMT_SMR.1
FMT_MTD.1	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_SMF.1	No dependencies.	
FMT_SMR.1	FIA_UID.1	FIA_UIA_EXT.1
FPT_APW_EXT.1	No dependencies.	
FPT_FLS.1	No dependencies.	
FPT_SKP_EXT.1	No dependencies.	
FPT_TST_EXT.1	No dependencies.	
FPT_TUD_EXT.1	No dependencies.	
FRU_RSA.1	No dependencies.	
FTA_SSL.3	No dependencies.	
FTA_SSL.4	No dependencies.	
FTA_TAB.1	No dependencies.	
FTP_ITC.1	No dependencies.	
FTP_TRP.1	No dependencies.	

Table 11: TOE SFR dependency analysis

6.2.3.1 Security Requirements Dependency Rationale

Below is the rationale for dependencies that are satisified by other SFRs then the specified by CC Part 2.

1. The dependency on FIA_UID.1 by FAU_GEN.2 is satisfied by FIA_UIA_EXT.1; this is acceptable because FIA_UIA_EXT.1 is a superset of FIA_UID.1; requiring not only identification, but also authentication.

2. The dependency on FCS_CKM.4 by FCS_CKM.1-RSA, FCS_COP.1(1), FCS_COP.1(2), and FCS_COP.1(3) is satisfied by FCS_CKM_EXT.4; this is acceptable because FCS_CKM_EXT.4 specifies the cryptographic keys more specifically (all plaintext secret and private

cryptographic keys and CSPs), the key destruction method (zeroization), and the timing of the destruction (when no longer required). Also, the application note gives additional guidance to the reader.

3. The dependency on FIA_UAU.1 by FIA_AFL.1, FIA_UAU.7 and FMT_SMR.1 is satisfied by FIA_UIA_EXT.1 because FIA_UIA_EXT.1 is a superset of FIA_UID.1; requiring not only authentication but also identification.

6.2.4 Internal consistency and mutual support of SFRs

The sufficiency rationale in section 6.2.2 has already demonstrated how the IT security requirements work together to implement the individual objectives for the TOE and the IT environment. This section will elaborate on the internal consistency and mutual support of the IT security requirements.

Generic requirements particular to network devices include residual information protection for content of traffic that is being mediated (FDP_RIP.2).

Requirements posed by [FWPP] are trivially reflected in FFW_RUL_EXT.1. This is supported by the management capabilities discussed below.

The TOE implements a variety of protocols to provide secure communication channels and trusted paths. Following NDPP, this is defined in a cascade of requirements starting with generic requirements (FTP_ITC.1 and FTP_TRP.1. On a protocol level, this is then spelled out by FCS_HTTPS_EXT.1 for HTTPS, and FCS_SSH_EXT.1 for SSH.

The LTM module of the TOE implements HTTP traffic authentication, as required in FIA_UAU.5. Proxying HTTPS traffic basically requires the termination and establishment of TLS connections, as defined in FCS_TLS_EXT.1 and, on a higher level, FTP_ITC.1.

The cryptographic support for the secure communications protocols discussed above is provided by central cryptographic libraries that are exhaustively described in section 1.5.3.8, including how the individual crypto SFRs work together to implement this support.

Management capabilities for administrators are defined in FMT_SMF.1, FMT_MTD.1, FMT_MSA.1, FMT_MSA.3, and FMT_SMR.1. Administrators are subject to an access control policy (FDP_ACC.1 and FDP_ACF.1), based on authentication of individual administrators (FIA_UIA_EXT.1, FIA_UAU_EXT.2, FIA_UAU.7, FIA_AFL.1, FIA_PMG_EXT.1), and further access restrictions (FPT_APW_EXT.1 and FPT_SKP_EXT.1). (User attributes are defined in FIA_ATD.1.) Access to the administrative interfaces is protected implementing the trusted paths described above, and is subject to display of an advisory (FTA_TAB.1) and proper termination of sessions (FTA_SSL.3, and FTA_SSL.4).

The TOE implements auditing for all security-relevant mechanisms: FAU_GEN.1 defines the types of audit records that the TOE must be able to generate. FAU_GEN.2 requires that audit records be associated with user identities, where possible. FAU_STG_EXT.1 requires that the TOE shall be able to send audit records to an external log server.

Underlying self-protection requirements for the TSF include prevention of exhaustion of administrative resources (FRU_RSA.1), a set of defined self-tests (FPT_TST_EXT.1), and capabilities to verify software updates (FPT_TUD_EXT.1).

Finally, the TOE provides failover capabilities between redundant systems. This is modeled in FPT_FLS.1.

The external states for failover are as follows:

- Active – operative and handling traffic
- Standby – operative, ready and able to take over from the Active unit if it should fail

- ForcedOffline – operative (has not failed) but has been manually made unavailable
- Offline – Not a displayed state, but represents the failure of the unit

6.3 Security Assurance Requirements

6.3.1 Assurance Requirements

The security assurance requirements (SARs) for the TOE are the Evaluation Assurance Level 4 components as specified in [CC] part 3, augmented by ALC_FLR.3.

The following table shows the SARs, and the operations performed on the components according to CC part 3: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
ADV Development	ADV_ARC.1 Security architecture description	CC Part 3	No	No	No	No
	ADV_FSP.4 Complete functional specification	CC Part 3	No	No	No	No
	ADV_IMP.1 Implementation representation of the TSF	CC Part 3	No	No	No	No
	ADV_TDS.3 Basic modular design	CC Part 3	No	No	No	No
AGD Guidance documents	AGD_OPE.1 Operational user guidance	CC Part 3	No	No	No	No
	AGD_PRE.1 Preparative procedures	CC Part 3	No	No	No	No
ALC Life-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation	CC Part 3	No	No	No	No
	ALC_CMS.4 Problem tracking CM coverage	CC Part 3	No	No	No	No
	ALC_DEL.1 Delivery procedures	CC Part 3	No	No	No	No
	ALC_DVS.1 Identification of security measures	CC Part 3	No	No	No	No
	ALC_FLR.3 Systematic flaw remediation	CC Part 3	No	No	No	No
	ALC_LCD.1 Developer defined life-cycle model	CC Part 3	No	No	No	No
	ALC_TAT.1 Well-defined development tools	CC Part 3	No	No	No	No
ASE Security Target evaluation	ASE_INT.1 ST introduction	CC Part 3	No	No	No	No
	ASE_CCL.1 Conformance claims	CC Part 3	No	No	No	No
	ASE_SPD.1 Security problem definition	CC Part 3	No	No	No	No
	ASE_OBJ.2 Security objectives	CC Part 3	No	No	No	No
	ASE_ECD.1 Extended components definition	CC Part 3	No	No	No	No
	ASE_REQ.2 Derived security requirements	CC Part 3	No	No	No	No

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
	ASE_TSS.1 TOE summary specification	CC Part 3	No	No	No	No
ATE Tests	ATE_COV.2 Analysis of coverage	CC Part 3	No	No	No	No
	ATE_DPT.1 Testing: basic design	CC Part 3	No	No	No	No
	ATE_FUN.1 Functional testing	CC Part 3	No	No	No	No
	ATE_IND.2 Independent testing - sample	CC Part 3	No	No	No	No
AVA Vulnerability assessment	AVA_VAN.3 Focused vulnerability analysis	CC Part 3	No	No	No	No

Table 12: SARs

6.4 Security Assurance Requirements Rationale

The basis for the justification of EAL4 augmented with ALC_FLR.3 is the threat environment experienced by the typical consumers of the TOE. This matches the package description for EAL4 (enhanced-basic).

7 TOE Summary Specification

7.1 TOE Security Functionality

7.1.1 Device management

7.1.1.1 Security Function Management

The TOE offers administrators a number of interfaces to configure and manage the TSF. This includes the Configuration utility, tmsh, and iControl API.

Some general configuration options include the definition of an administrative IP address for the TOE's management network interface, the configuration of failover mechanisms and the configuration of trusted updates.

In addition to the management (network) interface, the TOE features multiple switch interfaces for the physical connection to the networks carrying the traffic that the TSF mediates. Those switch interfaces can be assigned VLANs, and self IP addresses (i.e., an IP address that is associated with the TOE in that VLAN).

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), facilitating the access control mechanisms described below.

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.)

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.

In addition to pre-defined behavior options that administrators can choose from in the TOE's administration interfaces, such as rejecting traffic that does not come from a specific set of sender addresses, administrators can define custom rules for traffic inspection using the TOE's iRules scripting language, for example in order to inspect network packets for application-specific content and determine how traffic is treated based on the results. iRules conform to Tcl grammar rules [Tcl] and are driven by a defined list of events [iRules_EVENTS] that trigger the execution of specified commands [iRules_COMMANDS].

The following sections will add details on specific configuration options related to the individual security functions discussed.

This functionality implements FMT_MSA.1 and FMT_SMF.1.

7.1.1.2 Authentication

Administrative users (i.e., all users authorized to access the TOE's administrative interfaces) are identified by a user name and authenticated by an individual password associated with that user's account.

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 passwords (every user can change their own password), role, or partition the user has access to, and enabling or disabling terminal access for the user. However, User Managers that have access to only one partition cannot change the partition access of other users, and cannot change their own partition access or role. (More on roles and access control can be found in section 7.1.1.3.)

This functionality implements FIA_ATD.1, FIA_UIA_EXT.1, FIA_UAU_EXT.2, FMT_MSA.1, FMT_MSA.3.

Password policy and user lockout

The TOE can enforce a password policy for all user accounts managed locally. This includes the definition of a minimum password length and required character types (numeric, uppercase, lowercase, others). The minimum password length in the evaluated configuration is 15 characters. This policy is enforced when administrators change their own passwords.

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

The maximum duration specifies the maximum number of days a password is valid; users must change their passwords before the maximum duration is reached. User accounts whose password has expired, based on the administrator-defined maximum password duration, are locked and require an administrator to reset them.

Access to the TOE for individual users can be disabled ("locked") after a configured number of failed authentication attempts; which, in the evaluated configuration, the default is 3 with a valid range from 1 to 10. Administrative users with the role of Administrator or User Manager have to manually unlock accounts that have been locked by the TOE.

This functionality implements FIA_AFL.1, FIA_ATD.1 and FIA_PMG_EXT.1.

Default settings

The TOE comes with a pre-defined "admin" user with the Administrator role assigned that cannot be deleted. A password is assigned during setup of the TOE. (The root user for the underlying OS is disabled in the evaluated configuration.)

New users created for the TOE have the following default settings:

- user role
  - local user accounts: No Access
  - remote user accounts: No Access
- partition
  - local user accounts: the partition selected by the user that is creating the account
  - remote user accounts: Common
- partition access
  - local user accounts: All
  - remote user accounts: Common
- terminal access
  - local user accounts: Disabled

- remote user accounts: Disabled

The default setting for individual accounts can be changed at the time of account creation. The default settings listed above for remote user accounts are those assigned to users without specific access control settings defined in the TOE, and can be changed by administrator as part of the remote authentication scheme definition.

This functionality implements FIA_ATD.1 and FMT_MSA.3.

Authentication Banners, Obscured Feedback, and Time-outs

For interactive user authentication at the web-based Configuration utility and the command line tmsh via SSH, BIG-IP obscures passwords entered by users, and implements the display of administrator-defined banners to users before they authenticate.

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

If a user with a local user account is logged on to the BIG-IP system, and the system's configuration is changed from local authentication to remote authentication, the local user remains authenticated until the user’s logon session terminates.

This functionality implements FIA_UAU.7, FIA_UIA_EXT.1, FTA_TAB.1, FTA_SSL.3, FTA_SSL.4.

7.1.1.3 Access Control

The TOE allows authorized 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.

Access of individual users to the web-based Configuration utility, tmsh, and iControl 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. The types of objects managed by the TOE are defined as part of the security policy in section 1.5.6. Table 13 contains the definition of user roles.

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
- read: view an object

In addition to roles, the TOE implements the concept of partitions for restricting access to objects. 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 have 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.
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.
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, custom monitors, and iRules. 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.
iRule Manager	This role grants users permission to create, modify, and delete iRules. Users with this role cannot affect the way that an iRule is deployed. For example, a user with this role can create an iRule but cannot assign it to a virtual server or move the iRule from one virtual server to another. A user with this role can be assigned universal access to administrative partitions.
Application Editor	This role grants users permission to modify nodes, pools, pool members, monitors and change their own passwords. These users can view all objects on the system.
Operator	This role grants users permission to enable or disable nodes and pool members. These users can view all objects.
Auditor	This role grants users permission to view all configuration data on the system, including logs and archives. Users with this role cannot create, modify, or delete any data, nor can they view SSL keys or user passwords.

Role	Associated rights
Guest	This role grants users permission to view all objects on the system in their partition and Common partition.
Firewall Manager	This user has access only to the firewall software panel, and performs tasks associated only with security.
No Access	This role prevents users from accessing the system.

Table 13: BIG-IP User Roles

This functionality implements FDP_ACC.1, FDP_ACF.1, FMT_MTD.1, FMT_SMR.1.

7.1.1.4 Auditing

BIG-IP uses syslog functionality to generate audit events.

BIG-IP systems generate different log types that capture different types of events. This includes:

- 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 underlying 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 as discussed in section 7.1.2
- local traffic events
  events related to network traffic handled by the system, including some events related to packet filtering

The TOE allows to configure syslog levels per daemon that generates the respective audit records. The Configuration utility GUI and tmsh allow to set those log levels.

Table 14 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	

Log content		Log type				
		System	Packet Filter	Local Traffic	Audit (mcp)	Audit (other)
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	
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.				X	
User Name	The name of the user who made the configuration change.				X	X

Table 14: Audit Logs and Their Content

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 as described in section 7.1.1.5.

The syslog mechanism provided by the underlying Linux system is used for the creation (and forwarding) of audit records, and is consequently part of the TOE. In addition, BIG-IP implements a high-speed logging mechanism for data traffic in TMM that is compatible with syslog.

This functionality implements FAU_GEN.1, FAU_GEN.2, FAU_STG_EXT.1.

7.1.1.5 Communications Security

Administrator Access

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:

- SSH

  Connections to the TOE's command line interface are protected using SSH version 2, using AES with 256 bit-sizes keys, data integrity protection uses HMAC-SHA1, and RSA for authentication. The SSH implementation monitors packet size on all channels and limits packet size as suggested in RFC 4253 Section 6.1; the maximum packet size is (256*1024)

bytes with larger packets being silently dropped. Additionally, the SSH implementation has hard-coded diffie-hellman-group14-sha1 key exchange, diffie-hellman-group1-sha1 key exchange is intentionally disabled.

This functionality implements FDP_UCT.1, FDP_UIT.1, FCS_SSH_EXT.1, and FTP_TRP.1.

Further, time-outs for sessions between administrative users are implemented. Administrators can configure time-outs for idle sessions both for Configuration utility and tmsh sessions.

This functionality implements FTA_SSL.3.

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

This functionality implements FTA_SSL.4.

Communication with external audit servers

The TOE supports TLS channels to audit servers for the protection of audit records sent from the TOE to an external audit server.

This functionality implements FAU_STG_EXT.1 and FTP_ITC.1.

7.1.2 Basic Traffic Management

7.1.2.1 Packet Filter / Stateful Firewall

Rule-based Filtering

The TOE implements packet filtering functionality that, in the evaluated configuration, can be configured via the tmsh.

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

- source and destination IP addresses (per [RFC791] (IPv4) and [RFC2460] (IPv6))
- the transport layer protocol used (in particular, TCP or UDP)
- source and destination ports (per [RFC793] (TCP) and [RFC768] (UDP))
- ICMP message type and code (per [RFC792] (ICMPv4) and [RFC4443] (ICMPv6))

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

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

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

For stateful protocols (in particular, TCP) the TOE's rule enforcement considers the state of a network session when deciding whether to forward a network packet or deny it. I.e., if network packets during session establishment were permitted based on existing rulesets, then subsequent packets for the same session (matching source and destination IP addresses and ports, sequence numbers,

and flags) will be permitted without further evaluation, as long as the session is still active (i.e., matches an entry in TMM's state table). Similarly, UDP packets that match source and destination addresses and port of a previously permitted packet will be accepted within the limits of defined time-outs.

SYN cookies are implemented by the TOE's SYN Check feature. Administrators can specify a threshold (in absolute numbers) of active TCP connections for the system. Once this threshold is reached, the TOE will start using SYN cookies for TCP connection requests, i.e., use a proprietary algorithm to calculate individual sequence numbers for use in SYN/ACK responses to clients, instead of storing the connection requests in memory. If an ACK response is received, the TOE will calculate the original sequence value, and whether it matches the sequence number included in the SYN/ACK response. Only if this is the case, and the response has not exceeded a specific time limit, the connection will be accepted.

The TOE is also capable of generating dynamic rule sets for protocols that require more than one connection. For example, if an FTP control session is established based on an administrator-defined rule that permits that traffic, the TOE will create and enforce a dynamic rule that permits traffic matching the data connection defined in that control session per [RFC959]. Other protocols that include the dynamic definition of additional communication channels include RTSP [RFC2326] and SIP [RFC3261].

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

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

Network packets that do not match an explicit rule are denied.

Static Filtering

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

This includes:

- packets which are invalid fragments (for IPv6, as defined in [RFC5722])
- fragmented IP packets which cannot be re-assembled completely
- packets where the source address of the network packet is equal to the address of the network interface where the network packet was received
- packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received
- packets where the source address of the network packet is defined as being on a broadcast network
- packets where the source address of the network packet is defined as being on a multicast network
- packets where the source address of the network packet is defined as being a loopback address

- packets where the source address of the network packet is a multicast
- packets where the source or destination address of the network packet is a link-local address
- packets where the source or destination address of the network packet is defined as being an address “reserved for future use” as specified in [RFC5735] for IPv4
- packets where the source or destination address of the network packet is defined as an “unspecified address” or an address “reserved for future definition and use” as specified in [RFC4291] for IPv6
- packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified

RFC Conformance

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

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

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

- F5-written test cases for compliance and interoperability scenarios, covering UDP, ICMPv4 and ICMPv6, TCP, HTTP, HTTPS, SSH, SSL, and TLS
- Ixia's[2] IxANVL (Automated Network Validation Library) Protocol Compliance Test Suite, covering TCP
- TAHI [3]Project's Test and Verification for IPv6 Test Suite, covering IPv6 and ICMPv6

This functionality implements FFW_RUL_EXT.1.

7.1.2.2 Replay Detection

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. This is further discussed in [RFC4346] and [RFC4346] for TLS, and [RFC4251] for SSH.

This functionality implements FCS_SSH_EXT.1 and FCS_TLS_EXT.1.

7.1.2.3 TLS offloading

BIG-IP offers to terminate TLS traffic from clients and to servers on behalf of an organization. The most relevant example would be BIG-IP acting as a reverse proxy for incoming TLS client traffic that is destined for an organization's web servers. Instead of tasking individual web servers with the additional load of terminating the TLS sessions, and having a copy of the necessary private key reside on each of the web servers, BIG-IP takes over this task and then forwards unencrypted traffic

[2] Ixia is a third party test developer for automated network/protocol validation. (http://http://www.ixiacom.com/)
[3] The TAHI Project is a joint effort between The University of Tokyo, YDC Corp., and Yokogawa Electric Corp., formed with the objective of developing and providing the verification technology for IPv6. (http://http://www.tahi.org/)

to the web servers via a local network that does not require encryption for the protection of the traffic. Responses from the web servers back to the client will be encrypted by BIG-IP again. In another scenario, LTM re-encrypts the client traffic destined for the web servers, and the decryption serves the purpose of other optimization functions not considered security functionality in the context of this evaluation, while under the control of BIG-IP. Before the traffic leaves the TOE, it is re-encrypted.

Administrators can configure SSL Client and Server profiles, and associate them with virtual servers, in order to specify that LTM will act as a TLS server in communication with clients, and as a TLS client when communicating with servers, respectively.

Administrators can import certificates, certificate chains, and private keys to the TOE using the Configuration utility, which can then be referenced for use as server or client certificates in profiles for use in traffic TLS connections.

Other security-relevant configuration items available to administrators in profiles include the identification of trusted CA certificates, cipher suites to be enabled for the communication, other protocol details (such as whether re-negotiation is enabled and under which conditions).

Administrators can specify iRules for use in SSL Client profiles that allow the insertion of custom headers into HTTP requests forwarded to servers containing details about the validation results, such as the serial number of the certificate presented by the client, or whether certificate validation was successful. This allows the receiving server to customize responses based on this information without having to be involved in termination of the TLS tunnel itself.

When certificate-based authentication of clients (in SSL Client profiles) or servers (in SSL Server profiles) is enabled, administrators can determine whether certificates are optional or required, and the number of certificates that can be traversed in a certificate chain.

The protocol versions supported by the evaluated configuration for this function are TLS 1.1 and 1.2. The ciphersuites covered by this evaluation are identified in FCS_TLS_EXT.1.

This implements FIA_UAU.5, FTP_ITC.1, FDP_ITC.1, FCS_HTTPS_EXT.1 and FCS_TLS_EXT.1.

7.1.3 Cryptographic mechanisms

The cryptographic support functions spelled out in the SFRs and described extensively in section 7.1.3. The following sections provide additional information on the implementation of random number generation and zeroization.

This implements FCS_CKM.1, FCS_CKM.1-RSA, FCS_COP.1(1), FCS_COP.1(2), and FCS_COP.1(3).

The cryptography in the BIG-IP rely on cryptographic mechanisms for their effective implementation. The following table identifies those functions:

Functions	ST SFR
Trusted paths for TOE administration:	
● SSH for tmsh	FTP_TRP.1
Trusted channels between the TOE and external entities:	
● TLS connections to external audit servers	FTP_ITC.1
● TLS proxy functionality for web applications and clients	FTP_ITC.1

Table 15: Communications Security in BIG-IP

TOE-provided cryptography (FTP_ITC.1) - The cryptographic operations for the trusted channels are implemented in the software module within the TOE boundary. The following sections describe the claims that are made for these operations.

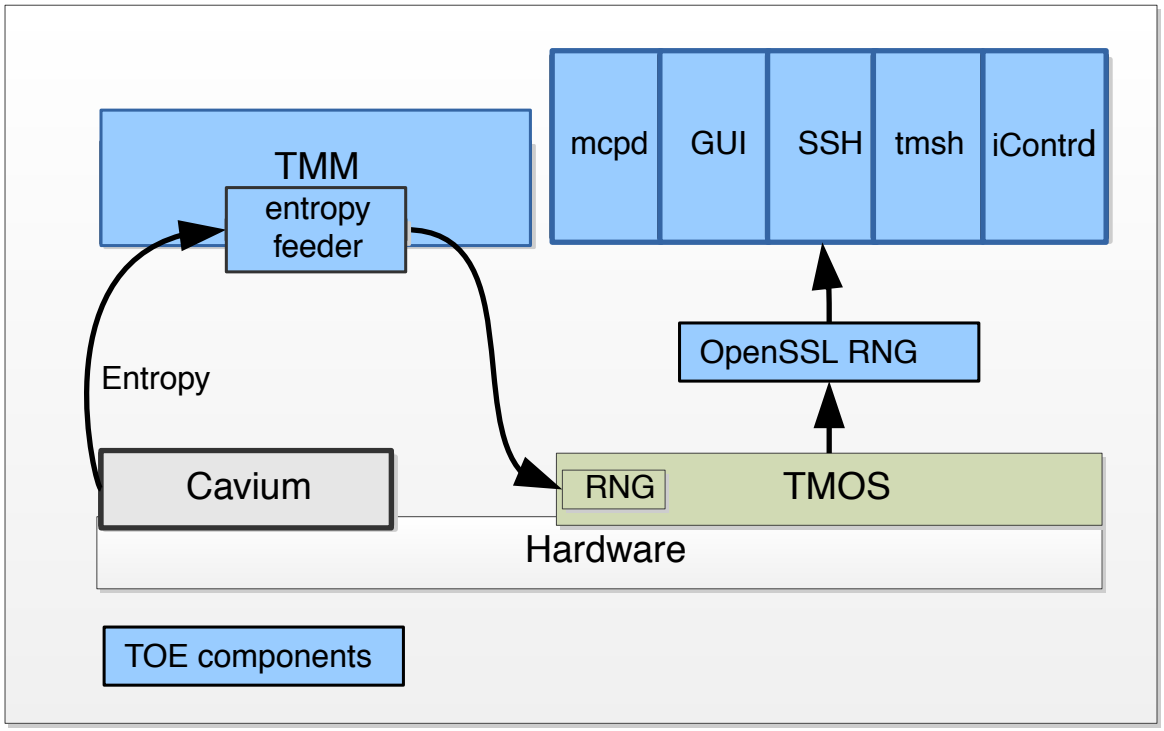

Figure 3: Cryptographic services in TOE and underlying hardware

The TOE incorporates a FIPS 140-2 validated OpenSSL library as cryptographic service provider, as indicated in Figure 3.

Higher-level protocol stacks can use this library in order to implement trusted communications:

1. SSH session for tmsh (SSH client to SSH server on TOE)
2. The SSL stack in TMM uses the host-provided library to implement the remaining, traffic-related TLS functionality use cases described above (also referred to as "traffic TLS"). Remote logging relies on traffic SSL.

The underlying hardware platforms of the TOE also include a proprietary crypto co-processor from Cavium that is used to provide sufficient entropy to support RNG. The Cavium Nitrox 3 processors are used on the Treadstone, Whitethorne, and Victoria II platforms; the Nitrox PX processor is used on the Centaur platform.

7.1.3.1 Key Generation

The TOE can generate key pairs for the SSH server.

(The operational environment is expected to provide TLS server and client keys and certificates for the traffic-related functions; however, the TOE generates all session keys.)

These keys are generated upon the request of an administrator by a Key Generator process that invokes the OpenSSL library on the Linux host. OpenSSL contains a pseudo-random number generator that has been configured to derive entropy from the Linux host via the /dev/random device. An Entropy Feeder process is invoked on an as needed basis to derive random numbers from the underlying Cavium hardware via TMM, and to feed those into the entropy pool used by /dev/random.

The Cavium hardware implements a physical noise source by using a large number of high-jitter free-running oscillators with output mixed in a linear feedback shift register, as documented in [US6954770].

The following table enumerates the types of keys that the TSF can generate:

Note: *The actual involvement of the TOE in the generation of TLS session keys depends on the key exchange method defined in the chosen cipher suite (see below), as well as the role (client or server) that the TOE acts in for a specific connection. (For administrative sessions, the TOE always acts as a server. For traffic sessions, the TOE may act as a TLS client or server.) In particular, this results in the TOE's dependency on entropy for session keys (secrets) as follows:*

- *TOE is server:*
  - *RSA: the pre_master_secret for further computational work is provided by the client, i.e., the operational environment*
  - *DHE, ECDHE: the TOE (as well as the client in the operational environment) both generate a random secret as part of computing the shared key*
- *TOE is client:*
  - *DHE, ECDHE: the TOE (as well as the server in the operational environment) both generate a random secret as part of computing the shared key*

All static keys such as the RSA key used for key transport and the static elliptic curve Diffie-Hellman key used for key agreement, are generated as part of the TOE environment. The ephemeral keys are generated within the TOE using domain parameters from the TOE environment.

7.1.3.2 Key Storage

Private keys, along with certificates, are stored in the TOE's configuration files and protected by the operational environment.

Optionally, the evaluated configuration can make use of a FIPS 140-2 Level 2 validated cryptographic module (if present) on the underlying system for the storage of RSA private keys that are used by either the Cavium crypto accelerator or the TMM SSL stack when the TOE acts as a TLS server toward remote network devices[4].

Likewise, functionality offered by the cards to synchronize the content of crypto modules in the redundant systems comprising the TOE, including the fipscardsync utility provided with BIG-IP for convenience, are considered to be out of scope for this evaluation.

7.1.3.3 Certificate validation

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

[4] Please note that the FIPS 140-2 Level 2 validated cryptographic module was not subject to evaluation

The TOE supports validation of X.509 digital certificates according to [RFC5280]. The TOE performs full certificate chain checking using Public Key Infrastructure X.509, verifies the expiration of the certificate (assuming a reliable time provided by the NTP server), and verifies its revocation using locally Certificate Revocation Lists

This implements FTP_ITC.1.

7.1.3.4 Random Number Generation

The TOE uses OpenSSL version 1.0 on the Linux host to generate key material, which is considered part of the TOE.

OpenSSL's pseudo-random number generator derives its seeds via /dev/random from the entropy pool of the underlying Linux host. This entropy pool, in turn, is primarily seeded by a physical noise source in the runtime environment, i.e., a Cavium crypto co-processor. This happens by means of a helper process that pools random numbers from the Cavium chip via its published APIs, and periodically feeds them into /dev/random.

The Cavium chip implements a physical noise source by using a large number of high-jitterfree-running oscillators. Their output is mixed in a linear feedback shift register, and goes through a SHA-1 hash engine in order to generate random numbers. As a result, the processor's physical noise source is (indirectly) providing the entropy for OpenSSL's random number generator.

The design how the entropy from the CAVIUM chip feeds into /dev/random can be characterized as follows:

- The TMM subsystem implements a driver to obtain random numbers from the CAVIUM chip. That driver exports the CAVIUM random data to a server listening on 127.1.1.2 port 3. A simple network request is able to obtain as many bytes as requested by the calling process.
- A daemon process implemented by F5 opens /dev/random and initializes a select() on the file descriptor. This select is triggered by the kernel when the entropy estimator for /dev/random shows that it runs low on entropy. If the kernel triggers the poll, the daemon process is woken up.
- Once the daemon is woken up, it reads 512 bytes from the TMM-exported CAVIUM random numbers and writes them into /dev/random using the IOCTL RNDADDENTROPY. The key now is that the IOCTL on /dev/random mixes the random data into the blocking_pool and nonblocking_pool and updates the entropy estimator by the number of bits written divided by 512 (i.e. the code implies that 512 bits from Cavium contain 1 bit of entropy). The daemon goes back to sleep afterwards to wait for the next select trigger.

The design discussion shows the following properties:

- Entropy gathered from the hardware RNG is injected into the input_pool of the Linux RNG using the mentioned IOCTL.
- The entropy estimator of the input_pool is updated by the number of bytes added to the input_pool divided by 512. This implies that the code assumes that 512 bits of data read from the Cavium hardware RNG contains one bit of entropy.
- Due to the frequent updates of the input_pool and corresponding update of the entropy estimator, the input_pool is able to deliver large amounts of entropy to the blocking_pool. This very fact can be interpreted such that (almost the entire) strength of /dev/random rests on the entropy that is collected from the Cavium hardware RNG. In the extreme case, no entropy from the Linux RNG original sources (HID, block devices, IRQs) are used; all entropy /dev/random provides is derived from the Cavium hardware RNG.

- The blocking behavior of /dev/random remains untouched. On the other hand, the Cavium hardware RNG can deliver random numbers at a very fast speed. The connection of the Linux RNG and the Cavium hardware RNG is such that every time the Linux RNG's entropy runs low, fresh entropy from the Cavium hardware RNG is delivered. This implies that the threshold that would cause /dev/random to block is rarely hit, if at all.

The amount of entropy produced by the noise source and provided through the Cavium chip's random number API is in excess of 200 Mbps. Random numbers are polled from the chip once per hour and fed into the Linux host's entropy pool. While there are other entropy sources fed by the Linux kernel into its entropy pool, this happens at a significantly slower rate.

Due to the large number of oscillators (> 100), failure of a number of them will not degrade the quality of the random numbers provided by the Cavium chip and used as entropy source. The underlying Cavium hardware implements the design described in [US6954770], which provides for independence from time, and reduces exposure to environmental conditions (like temperature changes) to negligible amounts of risk, in particular since the TOE hardware itself will be hosted in physically protected data centers.

This implements FCS_RBG_EXT.1.

7.1.3.5 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.”

The following table discusses how the cryptographic modules of the TOE, zeroize critical security parameters that are not needed for operation of the TSF anymore. This also includes key material used by the TSF that is stored outside of the crypto modules.

Application	Key type	Location	Zeroized when?	How?
Key generation	seeds, prime numbers, etc.	Stack/heap	After key has been generated.	These are zeroized in OpenSSL by calling OPENSSL_cleanse(), which overwrites the memory upon release
SSL/TLS	Session keys	Stack/heap	After session has ended	These are zeroized in OpenSSL by calling OPENSSL_cleanse(), which overwrites the memory upon release
SSL/TLS	private keys in TLS certificates	On the disk	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.

Application	Key type	Location	Zeroized when?	How?
SSH	Session keys	Stack/heap	After session has ended	These are zeroized in OpenSSL by calling OPENSSL_cleanse(), which overwrites the memory upon release
SSH	SSH keys	On the disk	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.

Table 16: Zeroization of Critical Security Parameters

This implements FCS_CKM_EXT.4.

7.1.3.6 Crypto Statement

The following table shows the cryptographic function of TLS used in the TOE.

#	Purpose	Cryptographic Mechanisms	Standard of Implementation	Key Size [Bits]	Comments
1	Authenticity	RSA signature verification (RSASSA-PKCS1-v1_5) using SHA-1 i	[FIPSPUB186-3][5] (RSA) referring to [RFC3447] (PKCS#1 v2.1) [FIPSPUB180-3][6] (SHA)	Modulus length: 2048, 3072, 4096	Algorithms used depending on the signature algorithm / hash algorithm used for signing the certificates and the accepted signature algorithms / hash algorithms by the peers. Server certificates required and client certificates optional. Verification of certificate signatures provided for authentication of peers. The certificates are not generated by the TOE* (imported into the TOE).
2		RSA signature verification (RSASSA-PKCS1-v1_5) using SHA-256, SHA-384	[RFC3447] (PKCS#1 v2.1) [FIPSPUB180-3] (SHA)	Modulus length: 2048, 3072 4096	
3		ECDSA signature generation and verification using SHA-1	[FIPSPUB186-3] (ECDSA), [FIPSPUB180-3] (SHA),	secp256r1 NIST P-256	
4		ECDSA signature generation and verification using SHA-256, SHA-384	[FIPSPUB186-3] (ECDSA), [FIPSPUB180-3] (SHA),	secp256r1 NIST P-256	

[5] Note, that [FIPSPUB186-3] is obsoleted by [FIPSPUB186-4]
[6] Note, that [FIPSPUB180-3] is obsoleted by [FIPSPUB180-4]

#	Purpose	Cryptographic Mechanisms	Standard of Implementation	Key Size [Bits]	Comments
5	Authentication Client Depending on client's certificate if any (subject public key info and key usage).	RSA signature generation (client) and verification (server). (RSASSA-PKCS1-v1_5[7]) using SHA-1	[RFC3447] (PKCS#1 v2.1) [RFC5246] (TLSv1.2)	Modulus length: 2048, 3072, 4096	Certificates with signing capability. Client cert type: rsa_sign. CertificateVerify: Client provides signature over the whole handshake message Only for TLSv1.1
6		RSA signature generation (client) and verification (server). (RSASSA-PKCS1-v1_5) using SHA-256, SHA-384	[RFC3447] (PKCS#1 v2.1) [RFC5246] (TLSv1.2)	Modulus length: 2048, 3072 4096	
7		RSA signature generation (client) and verification (server). (RSASSA-PKCS1-v1_5) using MD5 / SHA-1 combination	[RFC3447] (PKCS#1 v2.1) [RFC4346](TLSv1.1)	Modulus length: 2048, 3072 4096	
8		ECDSA signature generation and verification using SHA-1	[FIPSPUB186-3] (ECDSA), [FIPSPUB180-3] (SHA), [RFC4492] (ECC for TLS)	secp256r1 NIST P-256	Client cert Type: ecdsa_sign. The public key of the certificate MUST use a curve and point format supported by the server .
9		ECDSA signature generation and verification using SHA-256, SHA-384	[FIPSPUB186-3] (ECDSA), [FIPSPUB180-3] (SHA), [RFC4492] (ECC for TLS)	secp256r1 NIST P-256	
10	Authentication Server (static) [8]	Generating and verifying the PRF contained in the “Finished message”.	[RFC5246] (TLSv1.2) [RFC4346]		Please refer to PRF within key derivation below.

[7] implicitly EMSA-PKCS1-v1_5 encoding method is required based on block type 1 (PS= FF).

[8] Static keys / parameter contained in the certificate. Server Certificate must contain the RSA key or the ECDH parameters pub key. Key usage encipherment (RSA) and key exchange for (ECDH params). By successfully decoding the premaster secret (RSA) or computing / agree upon the premaster secret ECDH shared secret) and producing a correct “Finished message” with the master secret derived from the premaster secret as key, the server demonstrates that it knows the private key corresponding to the certificate. For TLS 1.0 the certificate must be signed with the algorithm contained in the cipher as key authentication algorithm. For TLS > 1.0 it is only historical to list the key authentication key within the cipher since cert signing is not any longer bound to the key contained in the cipher provided for key authentication.

#	Purpose	Cryptographic Mechanisms	Standard of Implementation	Key Size [Bits]	Comments
		(TLS_RSA, TLS_ECDH)	(TLSv1.1)		
11	Authentication Server (ephemeral)	RSA signature generation and verification (RSASSA-PKCS1-v1_5) using SHA-1 (TLS_ECDHE_RSA)	[RFC3447] (PKCS#1 v2.1) [FIPSPUB180-3] (SHA) [RFC5246] (TLSv1.2)	Modulus length: 2048, 3072 4096	See above, however with RSA as signature algorithm.
12		RSA signature generation and verification (RSASSA-PKCS1-v1_5) using SHA-256, SHA-384 (TLS_ECDHE_RSA)	[RFC3447](PKCS#1 v2.1) [FIPSPUB180-3] (SHA) [RFC5246] (TLSv1.2)	Modulus length: 2048, 3072 4096	
13		RSA signature generation (client) and verification (server). (RSASSA-PKCS1-v1_5) using MD5 / SHA-1 combination	[RFC3447](PKCS#1 v2.1) [RFC4346] (TLSv1.1)	Modulus length: 2048, 3072 4096	For TLS 1.1.
14		ECDSA signature generation and verification using SHA_1 (TLS_ECDHE_ECDSA)	[ANSI X9.62] (ECDSA), [FIPSPUB180-3] (SHA), [RFC4492] (ECC for TLS)	secp256r1 NIST P-256	See above however with ECDSA as signature algorithm.
15		ECDSA signature generation and verification using SHA-256, SHA-384 (TLS_ECDHE_ECDSA)	[ANSI X9.62] (ECDSA), [FIPSPUB180-3] (SHA), [RFC4492] (ECC for TLS),	secp256r1 NIST P-256	
16	Key establishment: Key transport	RSA encryption (client) and decryption (server) (RSAES-PKCS1-v1_5[9]) (TLS_RSA)	[RFC3447] (PKCS#1 v2.1) [SP800-56B] (IFC key establishment)	Modulus length: 2048, 3072 4096	Server certificate is used for key exchange. Encrypted exchange of pre-master secret generated at client side.

[9] implicitly EME-PKCS1-v1_5 encoding method is required based on block type 2 (PS= random data)

#	Purpose	Cryptographic Mechanisms	Standard of Implementation	Key Size [Bits]	Comments
					Server authentication (#11)
17	Key establishment: Key agreement Ephemeral	ECDHE	[RFC4492] (ECC for TLS) [TR-03111] (ECC) [SP800-56-A] (ECC DH)	secp256r1 NIST P-256	Unauthenticated ephemeral ECDH key / parameters provided by the server in the ServerKeyExchange. This is the only curve that is hard coded in the TOE.
18	Static	ECDH	[RFC4492] (ECC for TLS) [TR-03111] (ECC) [SP800-56-A] (ECC DH)	secp256r1 NIST P-256	The ECDH-parameters contained in the certificate (static). Since NIST P-256 is the only curve hard coded in the TOE – only certificates with ECDH parameters based on NIST P-256 will work. Server authentication (#14)
19	Key derivation	PRF: HMAC with SHA-256, 384 (default: prf_sha256 for TLSv1.2, also prf_sha384 possible)	[FIPS198-1] (HMAC) [FIPSPUB180-3] (SHA) [RFC5246] (TLSv1.2)	variable	Symmetric keys and MAC keys for record layer. Pre-master secret / (DH / ECDH shared secret) is converted into the master secret, the keys of the record layer are generated by expanding the master secret using the security parameters of the handshake protocol.
20		PRF: HMAC with MD5 and SHA-1 in combination(default: prf for TLS v1.1)	[FIPSPUB198-1] (HMAC) [RFC1321], RFC6151] (MD5) [FIPSPUB180-3] (SHA) [RFC4346] (TLS v1. 1)	variable	
21	Confidentiality	AES in CBC mode (AES_128_CBC, AES_256_CBC)	[FIPSPUB197] (AES) [SP800-38A] (CBC)	|k|=128, 256	Bulk data encryption / decryption(record layer) Supported by AES-NI
22	Authenticated Encryption	AES in GCM mode	[FIPSPUB197] (AES)	|k|=128, 256	

#	Purpose	Cryptographic Mechanisms	Standard of Implementation	Key Size [Bits]	Comments
		(AES_128_GCM, AES_256_GCM)	[RFC5228] (AES GCM within TLS)		
23	Integrity and authenticity	HMAC with SHA-1 or SHA-256 or SHA-384 (SHA), (SHA256), (SHA384)	[FIPS198-1] (HMAC) [FIPSPUB180-3] (SHA)	160 (SHA-1) 256 (SHA-256) 384(SHA-384)	Message authentication code (record layer)
26	Trusted Channel	FTP_ITC.1 [ST], sec. 6.1. 10.1 for HTTPS, syslog	Cf. all lines above	See above	

Table 17: Cryptograhic functions of TLS

The following table shows the cryptographic function of SSH used in the TOE.

#	Purpose	Cryptographic Mechanisms	Standard of Implementation	Key Size Bits	Comment
1	Authentication	RSA signature generation & verification RSASSAPKCS1-v1_5 using SHA-1 (ssh-rsa)	[RFC3447] (PKCS#1 v2.1) [FIPSPUB180-3] (SHA-1) [RFC4253] (SSH-TRANS) for host authentication [RFC4252], sec 7 (SSH-USERAUTH) for user authentication method: “publickey”	Modulus length: 1024	Pubkeys are exchanged trustworthy out of band. Authenticity is not part of the TOE. (no certificates used, server lists are in general possible at the client side – however client is not part of the TOE).
2		UserID & password	[RFC4252], sec. (SSH-USERAUTH) method; “password”	Guess success prob.	ST FIA_AFL.1: Recommendation as of [ECG] not to change the default setting where the blocking after 3 attempts is configured. Min 15 characters. No FIA_SOS claimed.

#	Purpose	Cryptographic Mechanisms	Standard of Implementation	Key Size Bits	Comment
3	Key establishment: Key agreement	DH with DH group14-sha1	[RFC4253] (SSH-TRANS) supported by [RFC3526] (DH groups IKE) [FIPSPUB180-3] (SHA-1)	plength= 2048	Hard coded in the TOE code.
4	Confidentiality	AES in CBC mode (aes256-cbc)	[FIPS197] (AES), [SP800-38A] (CBC), [RFC4253] (SSH using AES with CBC mode),	|k|= 256	Binary packet protocol: encryption Configured per *ccmode* script.
5	Integrity and authenticity	HMAC-SHA-1	[FIPSPUB180-3] (SHA), [RFC2104] (HMAC), [RFC4251] / [RFC4253] (SSH general / detailed HMAC support), [RFC4253] (SSH detailed HMAC support)	|k|=160	Binary packet protocol: message authentication
6	Key generation	RSA key generation	[FIPSPUB186-2], A.2.1 for Miller Rabin primality tests.	n/a	FCS_CKM.1 Host key generation using FCS_RBG_EXT.1
7	Trusted path	FTP_TRP.1, section 6.1.10.2 for SSH			

Table 18: Cryptograhic functions of SSH

7.1.4 TSF Protection and Support Functions

7.1.4.1 Failover of Redundant Systems

The TOE is comprised of two identically configured BIG-IP, i.e., same Part Number, systems running on homogenous hardware in a failover configuration. As a result, failure of (services on) one device will result in services being taken over by the redundant device.

failover requires the two devices of the evaluated configuration to be configured in a device group as a failover group, with one device being configured as active (the preferred device) and the other one configured as being in standby.

This involves, in principle, the following mechanisms:

1. Synchronization of configuration data between the two devices.

2. Synchronization of service availability between the two devices.
3. Optionally, state and persistence mirroring for network connections, for some protocols.

Synchronization of configuration data is implemented using BIG-IP's Central Management Infrastructure (CMI) architecture. The devices synchronize individual configuration changes, once they are committed, via messages sent through a dedicated network connection over the management network port. If device A becomes unavailable, device B will implement configuration changes locally and queue them for synchronization, which will occur once device A is available again.

State and persistence monitoring allows the TMM instances on the active and standby devices to exchange information about the state of individual traffic connections currently being handled by the active device via dedicated ports on network switches of the devices.

failover monitoring is performed both on the active and the standby device. A dedicated "overdog" process on the active device monitors the hearbeat of all critical daemons on the device, and communicates missing local hearbeats to a dedicated "switchover daemon" (sod). If the active device determines that it cannot reliable provide all services anymore, it will co-ordinate a failover to the standy device via CMI.

In addition, if the standby device detects missing heartbeats/communication from the active device, the standby device will become active and start managing traffic. even in the absence of a failure report from the active device. Regardless of whether state and persistence monitoring are enabled or not, the TOE as a whole will maintain a secure state, i.e., the enforcement of all security policies configured on the devices continues both for existing and new connections handled by the TOE.

State mirroring occurs at the transport layer. It does not apply to encrypted sessions.

Configuration data between the devices is exchanged over a dedicated line.

This functionality implements FPT_FLS.1.

7.1.4.2 Self-tests

The following self-test are implemented by the TOE:

1. The BIOS POST test is run at boot time
2. The OpenSSL integrity tests are run at boot for OpenSSL and for TMM SSL.
3. The sys-icheck utility is run at boot and restart to check the integrity of the RPMs

This functionality implements FPT_TST_EXT.1.

7.1.4.3 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 patches provided by F5.

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

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

This functionality implements FPT_TUD_EXT.1 and FMT_SMF.1.

7.1.4.4 Denial-of-Service Mitigation

The TOE implements multiple mechanisms to both mitigate network-originated DOS attacks that might lead to service degradation in particular, and resource exhaustion that might lead to loss of administrative control over the TOE in particular.

On the administrative interface side, this starts with the architectural property of having a dedicated network interface for device management traffic. TMM, as part of the dataplane that is handling the external (non-management) network traffic mediated by the TOE, is not involved in serving this interface, i.e. administrators can reach and configure the TOE without TMM being available on the system. In addition, administrators can configure the maximum number of concurrent sessions that the Configuration utility should accept.

In order to mitigate flooding attacks, administrators can define time-outs for inactive TCP and UDP connections in traffic profiles. If a connection does not show activity past that amount of time, it will be dropped.

When it comes to traffic processing, the TOE allocates dedicated memory to each TMM instance, and to the base Linux system hosting the configuration and management mechanisms. Administrators can specify quotas that activate TOE behavior to actively control the amount of connections mediated by the device based on how much memory TMM is currently using to handle existing connections.

This is implemented by BIG-IP's adaptive reaper mechanism. During normal operations, BIG-IP removes expired sessions entries from TMM's connection table as described above. When an administrator-specified "low-water mark threshold" is reached, the reaper mechanism is activated and parses connection table entries following a defined schedule in order to remove additional connection table entries in relation to the administrator-specified "hi-water mark threshold" as follows:

- If the relative level (number of connections) is greater than 20 % of the difference between lo-water and hi-water marks, the reaper mechanism will scale down the time-out for each connection, i.e., reduce the "normal" time-out, based on that level, and terminate connections based on that reduced time-out.
- If the relative level is greater than 50 %, the reaper will terminate connections that show a throughput rate of less than one packet per second of traffic.
- If the relative level is greater than 90 %, the reaper will randomly select 1/4 of connections and terminate them without sending a connection RST.

If the amount of memory used by TMM exceeds a certain percentage of the overall memory allocated to TMM, as specified by the administrator-defined *high-water mark threshold*, the TOE will not accept any new connections, but will continue to serve established connections.

This functionality implements FRU_RSA.1.

7.1.4.5 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 uses the underlying operating system for authentication of local user accounts. The TOE's administrative interfaces do not offer any sort of retrieval of user passwords or configuration files used by the underlying system.
- In transit between the TOE and authentication services

See section 7.1.1.5.

- In transit between remote users and the TOE

  See section 7.1.1.5.

Pre-shared keys, symmetric keys, and private keys are protected as follows:

- Pre-shared keys, such as credentials for remote servers, are stored in the TOE's configuration files.

This functionality implements FPT_APW_EXT.1 and FPT_SKP_EXT.1.

7.1.4.6 Residual Information Protection

Per the NDPP application note for FDP_RIP.2, “Resources” in the context of the FDP_RIP.2 requirement are network packets being sent through the TOE, therefore, the concern is that outgoing network packets do not unknowingly contain data that contains residual information. Each outgoing packet is comprised of data from one or more segment of physical memory; each linked with a header that contains the start address and the number of bytes to written into a part of the outgoing packet. When the packet is ready to be transmitted, for each segment that makes the packet, the corresponding physical address and of bytes (obtained from the header for that piece) is sent to the DMA driver code, which performs the DMA operations.

For any packet that is smaller than minimum payload size, the rest of the bytes that makes the minimum size are zeroed out with memclr().

The DMA driver DMAs only set count of bytes for each piece of the outgoing packet

This functionality implements FDP_RIP.2.

8 Abbreviations, Terminology and References

8.1 Abbreviations

ADF
Application Delivery Firewall

CMI
Central Management Infrastructure

CRL
Certificate Revocation List

CRLDP
Certificate Revocation List Distribution Point

DTLS
Datagram Transport Layer Security

FPGA
Field-Programmable Gate Array

FTP
File Transfer Protocol

GUI
Graphical User Interface

HSB
High-Speed Bridge

HSL
High-Speed Logging

LTM
Local Traffic Manager

OCSP
Online Certificate Status Protocol

RTSP
Real Time Streaming Protocol

SIP
Session Initiation Protocol

SOAP
Simple Object Access Protocol

TACACS+
Terminal Access Controller Access-Control System

TLS
Transport Layer Security

TMM
Traffic Managment Microkernel

TMOS
Traffic Management Operating System

vCMP
Virtual Clustered Multi-Processing

8.2 Terminology

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

Administrators
Administrators are administrative users of the TOE, i.e. those users defined in the TOE to be authorized to access the configuration interfaces of the TOE. Different roles can be assigned to administrators, including the Administrator role -- the name of the role is not to be confused with the general reference to an administrator being an administrative user of the TOE in any role.

User
Humans or machines interacting with the TOE via the provided user and programmatic interfaces. The TOE deals with different types of users -- administrators in charge of configuring and operating the TOE, traffic users who are subject to the TOE's firewalling capabilities.

8.3 References

AIS_20 **Funktionalitätsklassen und Evaluierungsmethodologie für deterministische Zufallszahlengeneratoren**
Version 3
Date 2013-05-15

CC **Common Criteria for Information Technology Security Evaluation**
Version 3.1R4
Date September 2012
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R4.pdf
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R4.pdf
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R4.pdf

ECG **Guidance Supplement: AGD_PRE and AGD_OPE**
Version 1.17
Date 2014-12-15

FIPS197 **FIPS PUB 197: Specification for the ADVANCED ENCRYPTION STANDARD (AES).**
Date November 26, 2001

FIPSPUB180-3 **FIPS PUB 180-3: Secure Hash Standard (SHS)**
Date October 2008

FIPSPUB180-4 **FIPS PUB 180-4: Secure Hash Standard (SHS)**
Date August, 2015

FIPSPUB186-2 **FIPS PUB 186-2: DIGITAL SIGNATURE STANDARD (DSS)**
Date 2000, January 27

FIPSPUB186-3 **FIPS PUB 186-3: Digital Signature Standard**
Date June, 2009

FIPSPUB186-4 **FIPS PUB 180-4: Digital Signature Standard (DSS)**
Date July, 2013

FIPSPUB197 **Federal Information Processing Standards Publication 197: Specification for the Advanced Encryption Standard (AES)**
Date November 26, 2001

FIPSPUB198-1 **FIPS PUB 198-1: The Keyed-Hash Message Authentication Code (HMAC)**
Date July 2008

FWPP **Network Device Protection Profile (NDPP) Extended Package Stateful Traffic Filter Firewall**
Author(s) NSA Information Assurance Directorate
Version 1.0
Date 2011-12-19

iRules_COMMANDS **Commands - DevCentral Wiki (registration required)**
Date received 2012-03-16
Location https://devcentral.f5.com/wiki/iRules.Commands.ashx

iRules_EVENTS **Events - DevCentral Wiki (registration required)**
Date received 2012-03-16
Location https://devcentral.f5.com/wiki/iRules.Events.ashx

NDPP **Protection Profile for Network Devices**
Author(s) NSA Information Assurance Directorate
Version 1.1
Date 2012-06-08

NIST800-90A **NIST Special Publication 800-90A: Recommendation for Random Number Generation Using Deterministic Random Bit Generators**
Date January 2012

RFC1321 **The MD5 Message-Digest Algorithm**
Author(s) R. Rivest
Date 1992-04-01
Location http://www.ietf.org/rfc/rfc1321.txt

RFC2104 **HMAC: Keyed-Hashing for Message Authentication**
Author(s) H. Krawczyk, M. Bellare, R. Canetti
Date 1997-02-01
Location http://www.ietf.org/rfc/rfc2104.txt

RFC2326 **RFC 2326: Real Time Streaming Protocol (RTSP)**
Author(s) H. Schulzrinne, A. Rao, R. Lanphier
Date April 1998

RFC2460 **RFC 2460: Internet Protocol, Version 6 (IPv6) Specification**
Author(s) S. Deering, R. Hinden
Date December 1998

RFC2818 **RFC 2818: HTTP Over TLS**
Author(s) E. Rescorla
Date May 2000

RFC3261 **RFC 3261: SIP: Session Initiation Protocol**
Author(s) J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler
Date June 2002

RFC3447 **Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1**
Author(s) J. Jonsson, B. Kaliski
Date 2003-02-01
Location http://www.ietf.org/rfc/rfc3447.txt

RFC3526 **More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)**
Author(s) T. Kivinen, M. Kojo
Date 2003-05-01
Location http://www.ietf.org/rfc/rfc3526.txt

RFC4251 **RFC 4251: The Secure Shell (SSH) Protocol Architecture**
Author(s) T. Ylonen, C. Lonvick, Ed.
Date January 2006

RFC4252 **RFC 4252: The Secure Shell (SSH) Authentication Protocol**
Author(s) T. Ylonen, C. Lonvick, Ed.
Date January 2006

RFC4253 **RFC 4253: The Secure Shell (SSH) Transport Layer Protocol**
Author(s) T. Ylonen, C. Lonvick, Ed.
Date January 2006

RFC4254 **RFC 4254: The Secure Shell (SSH) Connection Protocol**
Author(s) T. Ylonen, C. Lonvick, Ed.
Date January 2006

RFC4291 **RFC 4291: IP Version 6 Addressing Architecture**
Author(s) R. Hinden, S. Deering
Date February 2006

RFC4346 **RFC 4346: The Transport Layer Security (TLS) Protocol Version 1.1**
Author(s) T. Dierks, E. Rescorla
Date April 2006

RFC4443 **RFC 4443: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification**
Author(s) A. Conta, S. Deering, M. Gupta, Ed.
Date March 2006

RFC4492 **Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)**
Author(s) S. Blake-Wilson, N. Bolyard, V. Gupta, C. Hawk, B. Moeller
Date 2006-05-01
Location http://www.ietf.org/rfc/rfc4492.txt

RFC5228 **Sieve: An Email Filtering Language**
Author(s) P. Guenther, T. Showalter
Date 2008-01-01
Location http://www.ietf.org/rfc/rfc5228.txt

RFC5246 **The Transport Layer Security (TLS) Protocol Version 1.2**
Author(s) T. Dierks, E. Rescorla
Date 2008-08-01
Location http://www.ietf.org/rfc/rfc5246.txt

RFC5280 **Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile**
Author(s) D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk
Date 2008-05-01
Location http://www.ietf.org/rfc/rfc5280.txt

RFC5722 **RFC 5722: Handling of Overlapping IPv6 Fragments**
Author(s) S. Krishnan
Date December 2009

RFC5735 **RFC 5735: Special Use IPv4 Addresses**
Author(s) M. Cotton, L. Vegoda
Date January 2010

RFC768 **RFC 768: User Datagram Protocol**
Author(s) J. Postel
Date August 1980

RFC791 **RFC 791: Internet Protocol**
Author(s) J. Postel
Date September 1981

RFC792 **RFC 792: Internet Control Message Protocol**
Author(s) J. Postel
Date September 1981

RFC793 **RFC 793: Transmission Control Protocol**
Author(s) J. Postel
Date September 1981

RFC959 **RFC 959: File Transfer Protocol**
Author(s) J. Postel, J. Reynolds
Date October 1985

Tcl **Tcl Reference Manual**
Date received 2012-03-16
Location http://tmml.sourceforge.net/doc/tcl/index.html

TCPDUMP_FILTERS **TCPDUMP filters**
Date received 2012-03-16
Location http://www.cs.ucr.edu/~marios/ethereal-tcpdump.pdf

US6954770 **Random Number Generator. United States Patent No. US 6,954,770**
Author(s) David A. Carlson, Gregg A. Bouchard, Anand Varadharajan, Derek S. Brasili
Date Oct. 11, 2005