Sourcefire 3D® System Security Target

Date: June 12, 2014

Version: 1.0

Sourcefire © 2014

Prepared for and by:

Sourcefire, Inc.
9770 Patuxent Woods Drive
Columbia, MD 21046

Table of Contents

Table of Figures

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conformance claims, ST conventions, acronyms, and the ST organization. The TOE is an Intrusion Detection and Prevention System with stateful inspection firewall capability. The 3D System is comprised of two main components: Defense Center® (DC) and 3D Managed Devices (hereafter referred to as Devices). The DC provides a centralized management console and event database for the system, and aggregates and correlates intrusion, discovery, and connection data from managed devices. Devices monitor all network traffic for security events and violations, and can alert on and/or block[1] malicious traffic as defined in the intrusion and access control rules. In order to meet all the security functional requirements, the TOE must have both PROTECTION and CONTROL licenses.

1.1 Security Target, TOE and CC Identification

ST Title – Sourcefire 3D® System Security Target

ST Version – 1.0

ST Date – 6/12/2014

ST Author – Sourcefire, Inc.

TOE Identification – Sourcefire 3D® System[2] running Version 5.2.0.1

Table 1: TOE Models

TOE Series	Appliance Models
Defense Center	DC750
	DC1500
	DC3500
Devices	3D7010
	3D7020
	3D7030
	3D7110
	3D7115
	3D7120
	3D7125
	3D8120
	3D8130
	3D8140
	3D8250

[1] Blocking or altering traffic requires inline deployment.

[2] In the tested evaluated configuration, the TOE is comprised of at least one DC and one Device both running version 5.2.0.1.

TOE Series	Appliance Models
	3D8260
	3D8270
	3D8290
	3D9900

TOE Developer – Sourcefire, Inc.

Evaluation Sponsor – Sourcefire, Inc.

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- This ST is conformant to the *Security Requirements for Network Devices, 13 January 2013, Version 1.1 Errata 2* and *Network Device Protection Profile (NDPP) Extended Package Stateful Traffic Firewall, 19 December 2011, Version 1.0*.
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 4, September 2012.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.
  - Part 3 Conformant
  - Assurance Level: NDPP Errata 2[3]

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a number in parentheses placed at the end of the component. For example, FPT_ITT.1(1) and FPT_ITT.1(2) indicate that the ST includes two iterations of the FPT_ITT.1 requirement, (1) and (2).
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and underlined with embedded bold brackets (e.g., [***[selected-assignment]***]).

[3] Test 2 of FCS_TLS_EXT.1.1 has been waived based on TRRT decision TD0004. In addition, Test 3 of FPT_ITT.1 has been waived based on TRRT decision TD0005.

- Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
- Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… some **big** things …"). Note that 'cases' that are not applicable in a given SFR have simply been removed without any explicit identification.

- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.4 Acronyms

The following table defines product specific and CC specific acronyms used within this Security Target:

Table 2: TOE and CC Acronyms

Acronym	Definition
CC	Common Criteria [for IT Security Evaluation]
CLI	Command Line Interface
CM	Configuration Management
DB	Database
DC	Defense Center
FIPS	Federal Information Processing Standards
FTP	File Transfer Protocol
HTTP	HyperText Transmission Protocol
HTTPS	HyperText Transmission Protocol, Secure
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
NIST	National Institute of Standards and Technology
PP	Protection Profile
SAR	Security Assurance Requirement
SEU	Security Enhancement Updates
SF	Security Function
SFIDS	Sourcefire Intrusion Detection System
SFR	Security Functional Requirement
SNMP	Simple Network Management Protocol
ST	Security Target
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Security Layer
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TOE Security Functions Interface
TSP	TOE Security Policy
UDP	User Datagram Protocol

1.5 Security Target Organization

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8)

2. TOE Description

The TOE is an Intrusion Detection and Prevention System with stateful inspection firewall capability, which consists of the DC and Devices. The DC provides a centralized management console and event database for the system, and aggregates and correlates intrusion, discovery, and connection data from managed devices. Devices monitor all network traffic for security events and violations, and can alert and/or block malicious traffic as defined in the intrusion and access control rules. The TOE in the evaluated configuration deploys at least one DC managing at one or more Devices. Each model of the TOE consists of a set of appliances which vary primarily based on the processing power, memory performance, disk space, and port density. For more information, please refer to the "Hardware Specifications" section in the Sourcefire 3D System Installation Guide.

2.1 TOE Overview

The TOE combines the security of a network intrusion protection system with the power of access control based on network attributes such as addresses, ports, protocols, and more. The TOE monitors incoming and outgoing network traffic and performs real-time traffic analysis and logging using the industry-leading Snort® engine. All packets on the monitored network are scanned, decoded, preprocessed and compared against a set of rules to determine whether inappropriate traffic, such as system attacks, is being sent over the network. The system generates alerts or blocks the traffic when deviations of the expected network behavior are detected or when there is a match to a known attack pattern.

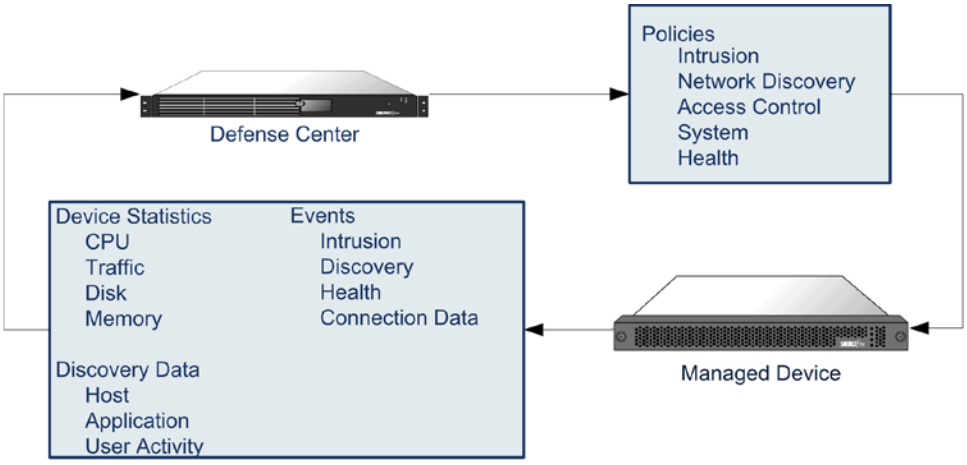
In addition, the TOE also provides stateful inspection filtering capability, hereafter referred to as access control. Access control is a policy-based feature that allows administrators to inspect and log the traffic that can enter, exit, or travel within the monitored network. An access control policy determines how the system handles traffic on the network. Administrators can include access control rules in an access control policy to further define how traffic is handled. For example, administrators can specify a rule action, such as to permit, deny, log, or inspect matching traffic with an intrusion policy.

The DC is a key component in the Sourcefire 3D System. Administrators can use the DC to manage the full range of devices that comprise the Sourcefire 3D System, and to aggregate, analyze, and respond to the threats they detect on their network. By using the Defense Center to manage devices, administrators can:

- Configure policies for all devices from a single location, making it easier to change configurations.
- Install various types of software updates on devices.
- Push policies to managed devices and monitor their health status from the DC.

The DC aggregates and correlates intrusion events, network discovery information, and device performance data, allowing administrators to monitor the information the devices are reporting in relation to one another, and to assess the overall activity occurring on their network. The following illustration lists what is transmitted between a DC and its managed devices.

Figure 1: Communication between DC and Device



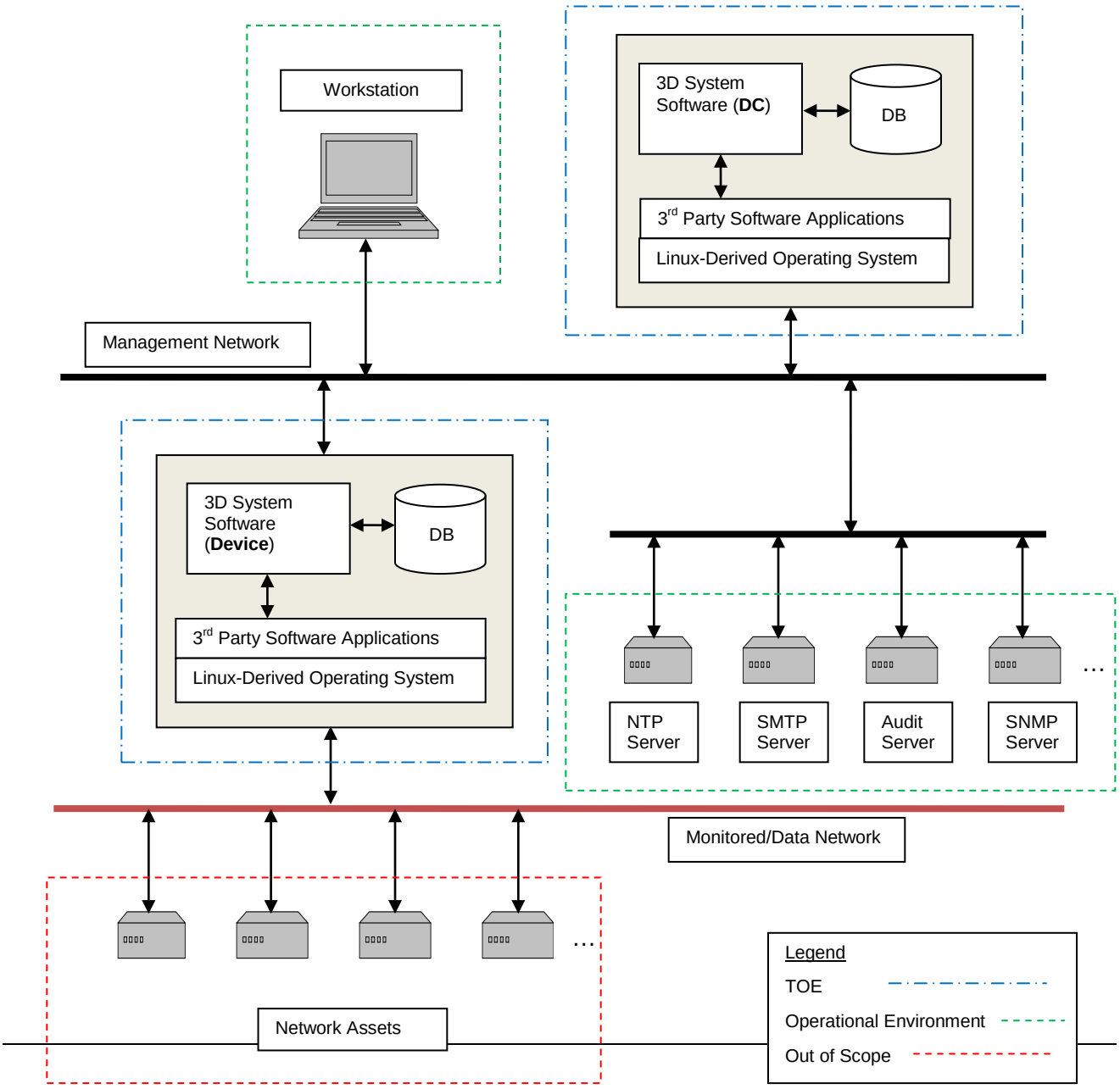
The administrators can configure the devices in either a passive or inline deployment. In a passive deployment, the device monitors traffic flowing across a network using a switch SPAN or mirror port. However, when configured in a passive deployment, the system cannot take certain actions such as blocking or shaping traffic. In an inline IPS deployment, the devices operate as a bump in the wire and is transparent (i.e., no IP address) on a network segment. The device can be configured to drop or alter packets, if necessary, in addition to generating alerts.

2.2 TOE Architecture

The TOE hardware appliances are purpose-built running on top of a customized, hardened Linux kernel. The hardware and operating system on which the Sourcefire 3D System application software operates provide the support necessary for the software applications to exist as processes and to access necessary disk, memory, and network connection resources. The appliance hardware, the underlying operating systems, embedded database, and third-party applications installed on the appliances provide support for the security functions and associated security management of the TOE.

The Sourcefire 3D System architecture can be depicted as follows:

Figure 2: TOE Architecture



The TOE main subsystems are summarized as followed:

- 3D System Software – The TOE main processes that provide the majority of the security management functions— including the SF CLI and web GUI—and proprietary algorithms to analyze, correlate, and display intrusion events.

- Database (DB) – The TOE contains a MySQL database which acts as the data repository for audit records, system event data, user account data, TOE and system configuration data. The system events sent from the managed 3D Devices are also stored in the database of the Defense Center.

- 3rd Party Software Applications – The TOE uses third party software processes and daemons to provide support to the 3D System Software subsystem. The supports include providing the underlying security protocols to protect the management communications (e.g., OpenSSH), FIPS-certified cryptographic algorithms (e.g., OpenSSL), web server to host web GUI (e.g., Tomcat Apache), auditing capability (e.g., auditd, sysklogd), other network services (e.g., ntp, net-snmp, dhcp), etc.

- Linux-Derived Operating System – The TOE uses a customized Linux kernel (based on version 2.6.35.14) to provide domain separation, memory management, disk access, file I/O, network stacks (IPv4/IPv6), and communications with the underlying hardware including the network interface cards. Only the services and packages required by the TOE for secure operation are enabled.

The TOE is composed of subsystems designed to implement security and management functions. For example there are subsystems dedicated to SNMP traps, web, and CLI management. There are also subsystems dedicated to the IPv4 and IPv6 network stacks as well as the applicable network protocols and switching, routing, etc.

From a security perspective, the TOE includes a FIPS-certified cryptographic module that supports SSH, TLSv1, and HTTPS (HTTP over TLSv1) and also digital signatures used to protect the available remote management and cryptographic hash to enable secure update capabilities of the TOE. Otherwise, the TOE also implements a wide range of non-security functions such as network switching protocols and high-availability.

2.2.1 Physical Boundaries

The TOE is a physical network rack-mountable appliance that supports modules that serve to offer a wide range of network ports varying in number, form factor (copper or fiber), processor power , disk space (40 – 400 GB) and memory performance (1 – 12 GB). The list of applicable models and devices is provided in section 1.1.

The TOE can be configured to rely on and utilize a number of other components in its operational environment.

- Management Workstation – The TOE supports CLI and web access and as such an administrator would need a terminal emulator or SSH client (supporting SSHv2) or web browser (supporting HTTPS such as Firefox 22.0 or later, or Internet Explorer 9 and 10) to utilize those administrative interfaces.

- Audit server – TOE can be configured to deliver audit records to an external log server (HTTP server).
- Authentication servers – The TOE can be configured to utilize external authentication servers.
- Certificate Authority (CA) server – The TOE can be configured to utilize digital certificates, e.g., for HTTPS connections.
- NTP server – The TOE can be configured to obtain time from a trusted time source.
- SMTP (E-mail) server – The TOE can be configured to send e-mail to alert specified users.
- SNMP server – The TOE can be configured to issue and received SNMP traps. Note that the TOE supports SNMPv3.
- DNS server – The TOE supports domain name service in the network.

2.2.2 Logical Boundaries

This section summarizes the security functions at a high-level and defines the logical boundaries of the TOE. For more details, please refer to section 6 in the ST. The following are the security functions provided by the TOE:

- Security Audit
- Cryptographic Support
- User Data Protection
- Stateful Traffic Filtering
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

2.2.2.1 Security Audit

The TOE is designed to be able to generate logs for a wide range of security relevant events such as login attempts and management functions. The complete list of auditable events and contents is in section 6.1. The TOE can be configured to store the logs locally so they can be accessed by an administrator or alternately to send the logs to a designated audit server over a secure communication channel. The timestamp included in the audit content can be manually set or set by an external NTP server in the operational environment.

2.2.2.2 Cryptographic Support

The TOE includes a FIPS-certified cryptographic module that provides key management, random bit generation, encryption/decryption, digital signature and secure hashing and key-hashing features in support of higher level cryptographic protocols including TLS, HTTPS, and SSH. The complete specification of algorithms, key sizes, and other attributes is in section 6.2.

2.2.2.3 User Data Protection

The TOE performs a wide variety of network switching, routing, and IDS/IPS functions, passing network traffic among its various physical and logical (e.g., VLAN) network connections. While

implementing applicable network protocols associated with network traffic storing and forwarding, the TOE is designed to ensure that it doesn't inadvertently reuse data found in network traffic pool.

2.2.2.4 Stateful Traffic Filtering

The TOE provides access control and intrusion protection to the monitored network. The TOE can process the standard network protocols such as ICMPv4, ICMPv6, IPv4, IPv6, TCP, and UDP and provide filtering based on network attributes such as addresses, ports, transport protocols, and more. Administrators can define what action is applied to a network packet when its attributes match the corresponding rule. In addition, the TOE maintains session state tables to track establishing connections and can dynamically allow packets that belong or in response to an existing, allowed connections. Finally, network packets that are invalid according to the standard RFCs are dropped.

2.2.2.5 Identification and Authentication

The TOE requires users (i.e., administrators) to be successfully identified and authenticated before they can access any security management functions available in the TOE. The TOE offers both a locally connected console and serial as well as network accessible interfaces (SSHv2 and HTTPS) for remote interactive administrator sessions.

The TOE supports the local (i.e., on device) definition of administrators with usernames and passwords. All authorized TOE users must have a user account with security attributes that control the user's access to TSF data and management functions. These security attributes include user name, password, and roles for TOE users.

Optionally, the TOE can be configured to utilize the services of trusted RADIUS and LDAP servers in the operational environment to support, for example, centralized user administration.

2.2.2.6 Security Management

The TOE provides a web-based (using HTTPS) management interface for all TOE administration, including the IDS and access control rule sets, user accounts and roles, and audit functions. The ability to manage various security attributes, system parameters and all TSF data is controlled and limited to those users who have been assigned the appropriate administrative role.

The TOE also provides a command line interface (CLI) and shell access to the underlying operating system of the TOE components. The shell access must be restricted to off-line installation, pre-operational configuration, and maintenance and troubleshooting of the TOE. The CLI provides only a subset of the management functions provided by the web GUI and is only available on the Devices. The use of the web GUI is highly recommended over the CLI.

Security management relies on a management workstation in the operational environment with a properly supported web browser or SSH client to access the management interfaces.

2.2.2.7 Protection of the TSF

The TOE implements a number of features design to protect itself to ensure the reliability and integrity of its security features. It protects particularly sensitive data such as stored passwords and cryptographic keys so that they are not accessible even by an administrator. It also provides its own timing mechanism to ensure that reliable time information is available (e.g., for log accountability) or can utilize a trusted time server in the operational environment.

The TOE ensures that data transmitted between separate parts of the TOE are protected from disclosure or modification. This protection is ensured by transmission of data between the TOE components over a secure, TLS-protected TCP tunnel.

The TOE includes functions to perform self-tests so that it might detect when it is failing. It also includes mechanisms so that the TOE itself can be updated while ensuring that the updates will not introduce malicious or other unexpected changes in the TOE.

2.2.2.8 TOE Access

The TOE can be configured to display an informative advisory banner when an administrator establishes an interactive session and subsequently enforce an administrator-defined inactivity timeout value after which the inactive session will be terminated. The administrators can also terminate their own interactive sessions when needed.

2.2.2.9 Trusted Path/Channels

The TOE protects interactive communication with administrators using SSHv2 for CLI access or HTTPS for web GUI access. The TOE protects communication with network peers, such as a log server, using HTTPS connections. All the underlying algorithms for the specified security protocols are FIPS-certified.

2.3 Excluded from Evaluation

The section identifies the features and capabilities that are provided by the TOE but are not evaluated. Section 1.1 of the NDPP, Compliant Targets of Evaluation, states "It is intended that the set of requirements in this PP is limited in scope in order to promote quicker, less costly evaluations that provide some value to end users. STs that include a large amount of additional functionality (and requirements) are discouraged." The list below identifies features that are not evaluated and the rationale why. Note that this does not mean the features cannot be used in the evaluated configuration. It means that the features were not evaluated and/or validated by an independent third party and the functional correctness of the implementation is vendor assertion.

The following features are not evaluated:

- Virtual Appliances – In past CC evaluations, the Sourcefire's virtual appliances (DC and Device) were successfully evaluated. According to CCEVS, the NDPP is only applicable to hardware appliances, not virtual appliances.
- VPN Gateway with IPSec – While the TOE meets (vendor assertion) the FCS_IPSEC_EXT.1 SFR, the NDPP states "The intent of the above requirement is to use a cryptographic protocol to protect external communications with authorized IT entities that the TOE interacts with to perform its functions. This is not, however, to be used to specify VPN Gateway functionality; a separate VPN Protection Profile should be used in these instances." Therefore, the VPN IPSec gateway feature is not evaluated.
- Access control rule conditions – The following conditions can be used in the access control rule but are beyond the scope of evaluation:
    - VLAN Tag
    - User
    - Application, with exception of FTP
    - URL
- Any features not associated with SFRs in claimed NDPP and extended PP – NDPP forbids adding additional requirements to the ST. If additional functionalities are mentioned in the ST, it is for completeness only.

In addition, this section also identifies features and capabilities provided by the TOE that are not allowed in the CC evaluated configuration. This means that these features would violate the requirements in the PPs or would make the TOE vulnerable to attacks if enabled.

The following things that are not allowed in the CC evaluated configuration:

- External Authentication Servers – The NDPP does not require external authentication servers. However, if they are used, the connection between the TOE and server must be protected by the approved security protocol.
- Shell Access – The shell access is only allowed for pre-operational installation, configuration, and post-operational maintenance and troubling shooting.
- Timeout Exemption Option – The use of the "Exempt from GUI Session Timeout" setting is not permitted. This allows a user to be exempted from the inactivity timeout feature.

2.4 TOE Documentation

The Sourcefire 3D System documentation set includes online help and PDF files.

The following product guidance documents are provided with the TOE on the Documentation CD included with the product:

Sourcefire 3D System Installation Guide, Version 5.2, 2013-June-07 13:25
Sourcefire 3D System Release Notes Version 5.2, June 27, 2013
Sourcefire 3D System User Guide, Version 5.2, 2013-June-04 13:22
Sourcefire 3D System CC Supplemental User Guide, Version 2.0, April 30, 2014

Online help can be accessed in two ways:

- By clicking the context-sensitive help links on each page
- By selecting Help > Online

The most up-to-date versions of the documentation can be accessed on the Sourcefire Support web site (https://support.sourcefire.com/).

3. Security Problem Definition

The Security Problem Definition (composed of organizational policies, threat statements, and assumption) has been drawn verbatim from the *Security Requirements for Network Devices, 08 June 2012, Version 1.1* and *Network Device Protection Profile (NDPP) Extended Package Stateful Traffic Firewall, 19 December 2011, Version 1.0*. The assumptions and threats from the Extended Package Stateful Traffic Firewall (EPFW) are added on top of those defined in the NDPP. There is no modification, addition, or deletion to any of the defined organizational policies, threat statements, or assumptions. If more information is of interest, please refer to the actual PP document.

In general, the NDPP has presented a Security Problem Definition appropriate for network infrastructure devices while the EPFW has augmented additional assumptions and threats more specifically targeted for network devices with filtering technology such as Stateful Traffic Filter Firewalls.

3.1 Assumptions

The assumptions state the specific conditions that are expected to be met by the development environment, operational environment, and/or administrators.

Table 3: TOE Assumptions

Assumption Name	Assumption Definition
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.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.PHYSICAL*	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
A.TRUSTED_ADMIN*	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

* - NDPP

3.2 Threats

The threats are security risks to the operational environment or the TOE itself that are addressed by the technology required in the PPs.

Table 4: Threats

Threat Name	Threat Definition
T.ADMIN_ERROR*	An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.
T.NETWORK_DISCLOSURE	Sensitive information on a protected network might be disclosed resulting from ingress- or egress-based actions.
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_MISUSE	Access to services made available by a protected network might be used counter to Operational Environment policies.
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.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.

* - NDPP

3.3 Organizational Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs.

Table 5: Organizational Security Policies

Policy Name	Policy Definition
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.

* - NDPP

4. Security Objectives

The Security Objectives have been drawn verbatim from the NDPP and EPFW. The Security Objectives for the TOE and operational environment from the EPFW are added on top of those defined in the NDPP. There is no modification, addition, or deletion to any of the defined Security Objectives. If more information is of interest, please refer to the actual PP document.

In general, the NDPP has presented a Security Objectives appropriate for network infrastructure devices while the EPFW has augmented additional Security Objectives more specifically targeted for network devices with filtering technology such as Stateful Traffic Filter Firewalls.

4.1 Security Objectives for the TOE

The TOE security objectives are listed in alphabetical order.

Table 6: Security Objectives for the TOE

TOE Security Objective	TOE Security Objective Definition
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.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.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 and (optionally) from a trusted source.

* - NDPP

4.2 Security Objectives for the Operational Environment

The security objectives for the operational environment are listed in alphabetical order.

Table 7: Security Objectives for Operational Environment

Security Objective	Security Objective Definition
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.NO_GENERAL_PURPOSE*	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.PHYSICAL*	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.TRUSTED_ADMIN*	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

* - NDPP

5. IT Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the TOE and to scope the evaluation effort. The NDPP identifies the core security requirements for any network infrastructure devices. The EPFW augments those core requirements with additional stateful filtering requirements.

The SFRs have all been drawn from the following Protection Profiles (PPs): *Security Requirements for Network Devices, 08 June 2012, Version 1.1, Security Requirements for Network Devices, 13 January 2013, Version 1.1 Errata 2,* and *Network Device Protection Profile (NDPP) Extended Package Stateful Traffic Firewall, 19 December 2011, Version 1.0.* As a result, refinements and operations already performed in that PPs are not identified (e.g., highlighted) here, rather the requirements have been copied from the respective PP and any residual operations have been completed herein. Of particular note, the NDPP, NDPP Errata 2, and EPFW made a number of refinements and completed some of the SFR operations defined in the CC and the actual PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the NDPP or NDPP Errata 2. Additionally, the SARs are effectively refined since the 'Assurance Activities' defined in the NDPPs and EPFW have been reproduced in section 5.4 to ensure they are included in the scope of the evaluation effort.

5.1 Extended Requirements

All of the extended requirements in this ST have been drawn from the NDPPs and EPFW. The NDPP and EPFW define the following extended SFRs and since they are not redefined in this ST, the NDPPs should be consulted for more information in regard to those CC extensions. The ST does not include any new extended requirements outside of those already defined in the NDPPs and EPFW.

Table 8: Extended Requirements

SFR	Name	PP
FAU_STG_EXT.1	External Audit Trail Storage	NDPP
FCS_CKM_EXT.4	Cryptographic Key Zeroization	NDPP
FCS_HTTPS_EXT.1	Explicit: HTTPS	NDPP
FCS_RBG_EXT.1	Extended: Cryptographic Operation (Random Bit Generation)	NDPP
FCS_SSH_EXT.1	Explicit: SSH	NDPP
FCS_TLS_EXT.1	Explicit: TLS	NDPP
FFW_RUL_EXT.1	Stateful Traffic Filtering	EPFW
FIA_PMG_EXT.1	Password Management	NDPP
FIA_UAU_EXT.2	Extended: Password-based Authentication Mechanism	NDPP
FIA_UIA_EXT.1	User Identification and Authentication	NDPP
FPT_SKP_EXT.1	Extended: Protection of TSF Data (for reading of all symmetric	NDPP

	keys)	
FPT_APW_EXT.1	Extended: Protection of Administrator Passwords	NDPP
FPT_TUD_EXT.1	Extended: Trusted Update	NDPP
FPT_TST_EXT.1	TSF Testing	NDPP
FTA_SSL_EXT.1	TSF-initiated Session Locking	NDPP

5.2 TOE Security Functional Requirements

The section identifies and describes the SFRs for the TOEs. All of the security requirements in this ST have been drawn from the NDPP and EPFW. The SFRs that appear in table below are described in more detail in the following subsections:

Table 9: TOE Security Functional Requirements

Requirement Class	Requirement Component	PP
FAU: Security Audit	FAU_GEN.1: Audit Data Generation	NDPP[4]
	FAU_GEN.2: User Identity Association	NDPP
	FAU_STG_EXT.1: External Audit Trail Storage	NDPP
FCS: Cryptographic Support	FCS_CKM.1: Cryptographic Key Generation (for asymmetric keys)	NDPP
	FCS_CKM_EXT.4: Cryptographic Key Zeroization	NDPP
	FCS_COP.1(1): Cryptographic Operation (for data encryption/decryption)	NDPP
	FCS_COP.1(2): Cryptographic Operation (for cryptographic signature)	NDPP
	FCS_COP.1(3): Cryptographic Operation (for cryptographic hashing)	NDPP
	FCS_COP.1(4): Cryptographic Operation (for keyed-hash message authentication)	NDPP
	FCS_HTTPS_EXT.1: Explicit: HTTPS	NDPP
	FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)	NDPP
	FCS_SSH_EXT.1: Explicit: SSH	NDPP
	FCS_TLS_EXT.1: Explicit TLS	NDPP
FDP: User Data Protection	FDP_RIP.2: Full Residual Information Protection	NDPP
FFW: Extended Firewall	FFW_RUL_EXT.1: Stateful Traffic Filtering	EPFW
FIA: Identification and Authentication	FIA_PMG_EXT.1: Password Management	NDPP
	FIA_UIA_EXT.1: User Identification and Authentication	NDPP
	FIA_UAU_EXT.2: Extended: Password-based Authentication Mechanism	NDPP
	FIA_UAU.7: Protected Authentication Feedback	NDPP

[4] There is no FAU_GEN.1 in EPFW but there are auditable events required in the EPFW which are added to auditable events specified in NDPP.

Requirement Class	Requirement Component	PP
FMT: Security Management	FMT_MTD.1: Management of TSF Data (for general TSF data)	NDPP
	FMT_SMF.1: Specification of Management Functions	NDPP[5]
	FMT_SMR.2: Restrictions on Security Roles	NDPP
FPT: Protection of the TSF	FPT_ITT.1: Basic Internal TSF Data Transfer Protection	NDPP
	FPT_SKP_EXT.1: Extended: Protection of TSF Data (for reading of all symmetric keys)	NDPP
	FPT_APW_EXT.1: Extended: Protection of Administrator Passwords	NDPP
	FPT_STM.1: Reliable Time Stamps	NDPP
	FPT_TUD_EXT.1: Extended: Trusted Update	NDPP
	FPT_TST_EXT.1: TSF Testing	NDPP
FTA: TOE Access	FTA_SSL_EXT.1: TSF-initiated Session Locking	NDPP
	FTA_SSL.3: TSF-initiated Termination	NDPP
	FTA_SSL.4: User-initiated Termination	NDPP
	FTA_TAB.1: Default TOE Access Banners	NDPP
FTP: Trusted Path/Channels	FTP_ITC.1: Inter-TSF Trusted Channel	NDPP
	FTP_TRP.1: Trusted Path	NDPP

[5] There is no FMT_SMF.1 in EPFW but there are security management functions required in the EPFW which are added to ones specified in NDPP.

5.2.1 Security Audit (FAU)

5.2.1.1 Audit Data Generation (FAU_GEN.1)

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

a) Start-up and shut-down of the audit functions;

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

c) All administrative actions;

d) [Specifically defined auditable events listed in ~~Table 1~~ **Table 12**].

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [information specified in column three of ~~Table 1~~ **Table 12**].

Table 10: Auditable Events and Contents

Requirement	Auditable Events	Additional Audit Record Contents	PP
FAU_GEN.1	None.		**NDPP**
FAU_GEN.2	None.		**NDPP**
FAU_STG_EXT.1	None.		**NDPP**
FCS_CKM.1	None.		**NDPP**
FCS_CKM_EXT.4	None.		**NDPP**
FCS_COP.1(1)	None.		**NDPP**
FCS_COP.1(2)	None.		**NDPP**
FCS_COP.1(3)	None.		**NDPP**
FCS_COP.1(4)	None.		**NDPP**
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.	**NDPP**
FCS_RBG_EXT.1	None.		**NDPP**
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.	**NDPP**
FCS_TLS_EXT.1	Failure to establish a TLS Session. Establishment/Termination of a	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and	**NDPP**

Requirement	Auditable Events	Additional Audit Record Contents	PP
	TLS session.	failures.	
FDP_RIP.2	None.		**NDPP**
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	**EPFW**
	Indication of packets dropped due to too much network traffic	TOE interface that is unable to process packets	**EPFW**
FIA_PMG_EXT.1	None.		**NDPP**
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).	**NDPP**
FIA_UAU_EXT.2	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).	**NDPP**
FIA_UAU.7	None.		**NDPP**
FMT_MTD.1	None.		**NDPP**
FMT_SMF.1	None.		**NDPP**
FMT_SMR.2	None.		**NDPP**
FPT_ITT.1	None.		**NDPP**
FPT_SKP_EXT.1	None.		**NDPP**
FPT_APW_EXT.1	None.		**NDPP**
FPT_STM.1	Changes to the time.	The old and new values for the time. Origin of the attempt (e.g., IP address).	**NDPP**
FPT_TUD_EXT.1	Initiation of update.	No additional information.	**NDPP**
FPT_TST_EXT.1	None.		**NDPP**
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	No additional information.	**NDPP**
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	No additional information.	**NDPP**
FTA_SSL.4	The termination of an interactive session.	No additional information.	**NDPP**
FTA_TAB.1	None.		**NDPP**
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.	**NDPP**
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.	**NDPP**

5.2.1.2 User Identity Association (FAU_GEN.2)

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

5.2.1.3 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/HTTPS***] protocol.

5.2.2 Cryptographic Support (FCS)

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

FCS_CKM.1.1 Refinement: The TSF shall generate asymmetric cryptographic keys used for key establishment in accordance with [

- ***NIST Special Publication 800-56B, "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.

5.2.2.2 Cryptographic Key Zeroization (FCS_CKM_EXT.4)

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

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

FCS_COP.1.1(1) Refinement: The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES operating in [**CBC mode**]] and cryptographic key sizes 128-bits and 256-bits that meets the following:

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

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

FCS_COP.1.1(2) Refinement: The TSF shall perform cryptographic signature services in accordance with a [

(2) RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater]

that meets the following:

Case: RSA Digital Signature Algorithm

- FIPS PUB 186-2 or FIPS PUB 186-3, "Digital Signature Standard"

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

FCS_COP.1.1(3) Refinement: The TSF shall perform [cryptographic hashing services] in accordance with a specified cryptographic algorithm [***SHA-1, SHA-224, SHA-256, SHA-384, SHA-512***] and message digest sizes [***160, 224, 256, 384, 512***] bits that meet the following: FIPS Pub 180-3, "Secure Hash Standard."

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

FCS_COP.1.1(4) Refinement: The TSF shall perform [keyed-hash message authentication] in accordance with a specified cryptographic algorithm HMAC-[***SHA-1, SHA-224, SHA-256, SHA-384, SHA-512***], key size [**20 bytes (160 bits)**], and message digest sizes [***160, 224, 256, 384, 512***] bits that meet the following: FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."

5.2.2.7 Explicit: HTTPS (FCS_HTTPS_EXT.1)

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.

5.2.2.8 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 [***FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using 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 [***128 bits***] of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

5.2.2.9 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, 4254, and [***no other RFCs***].

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 [**256 K**] 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, [***AES-CBC-192***].

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 and [***no other method***] are the only allowed key exchange method used for the SSH protocol.

Application Notes: The support for hmac-md5, though allowed, has been removed due to conflict with other government requirements. The SSH transport implementation also

supports stronger hmac-sha2-256 and hmac-sha2-512 as the data integrity algorithm.

5.2.2.10 Explicit: TLS (FCS_TLS_EXT.1)

FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols [***TLS 1.0 (RFC 2246)***] supporting the following ciphersuites:

Mandatory Ciphersuites:

TLS_RSA_WITH_AES_128_CBC_SHA

Optional Ciphersuites:

[***TLS_RSA_WITH_AES_256_CBC_SHA***
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA].

5.2.3 User Data Protection (FDP)

5.2.3.1 Full Residual Information Protection (FDP_RIP.2)

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

5.2.4 Stateful Traffic Filtering (FFW)

5.2.4.1 Stateful Traffic Filtering (FFW_RUL_EXT.1)

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

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

- Internet Control Message Protocol version 4 (ICMPv4)
- Internet Control Message Protocol version 6 (ICMPv6)
- Internet Protocol (IPv4)
- Internet Protocol version 6 (IPv6)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

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

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

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

- ICMPv4
  - o Type
  - o Code
- ICMPv6
  - o Type
  - o Code
- IPv4
  - o Source address
  - o Destination Address
  - o Transport Layer Protocol
- IPv6
  - o Source address
  - o Destination Address
  - o Transport Layer Protocol
- TCP
  - o Source Port
  - o Destination Port
- UDP
  - o Source Port
  - o Destination Port

and distinct interface.

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

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

FFW_RUL_EXT.1.6 The TSF shall:

a) accept a network packet without further processing of Stateful Traffic Filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [***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:

- FTP: TCP data sessions in accordance with the FTP protocol as specified in RFC 959,
- [***none***].

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

1. The TSF shall reject and be capable of logging packets which are invalid fragments;

2. The TSF shall reject and be capable of logging fragmented IP packets which cannot be re-assembled completely;

3. The TSF shall reject and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;

4. The TSF shall reject and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received;

5. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a broadcast network;

6. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a multicast network;

7. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;

8. The TSF shall reject and be capable of logging network packets where the source address of the network packet is a multicast;

9. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is a link-local address;

10. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as being an address “reserved for future use” as specified in RFC 5735 for IPv4;

11. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as an

"unspecified address" or an address "reserved for future definition and use" as specified in RFC 3513 for IPv6;

12. The TSF shall reject and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and

13. [***no other rules***].

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

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

5.2.5 Identification and authentication (FIA)

5.2.5.1 Password Management (FIA_PMG_EXT.1)

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

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and special characters: [***"!", "@", "#", "$", "%", "^", "&", "*", "(", ")"***, ***"-", "_", "=" "+", "[", "]", "{", "}", ":", ":", "'", """, ",", "<", ".", ">", "/", "?"***];
2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater;

5.2.5.2 User Identification and Authentication (FIA_UIA_EXT.1)

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

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

FIA_UIA_EXT.1.2 The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

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

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

Application Notes: The TOE supports the use of LDAP or RADIUS servers for remote authentication but this was not evaluated and is not allowed in the evaluated configuration.

5.2.5.4 Protected Authentication Feedback (FIA_UAU.7)

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

5.2.6 Security Management (FMT)

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

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

5.2.6.2 Specification of Management Functions (FMT_SMF.1)

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

- Ability to administer the TOE locally and remotely;
- Ability to update the TOE, and to verify the updates using [***published hash***] capability prior to installing those updates;
- [***Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1***
- ***Configure Firewall rules***]

5.2.6.3 Security Roles (FMT_SMR.2)

FMT_SMR.2.1 The TSF shall maintain the roles:

- Authorized Administrator.

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

FMT_SMR.2.3 The TSF shall ensure that the conditions

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

are satisfied.

5.2.7 Protection of the TSF (FPT)

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

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

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

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

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

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

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

5.2.7.4 Reliable Time Stamps (FPT_STM.1)

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

5.2.7.5 Extended: Trusted Update (FPT_TUD_EXT.1)

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

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

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

5.2.7.6 TSF Testing (FPT_TST_EXT.1)

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

5.2.8 TOE Access (FTA)

5.2.8.1 TSF-initiated Session Locking (FTA_SSL_EXT.1)

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

- ***terminate the session***]

after a Security Administrator-specified time period of inactivity.

5.2.8.2 TSF-initiated Termination (FTA_SSL.3)

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

5.2.8.3 User-initiated Termination (FTA_SSL.4)

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

5.2.8.4 Default TOE Access Banners (FTA_TAB.1)

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

5.2.9 Trusted Path/Channels (FTP)

5.2.9.1 Inter-TSF Trusted Channel (FTP_ITC.1)

FTP_ITC.1.1 Refinement: The TSF shall use [***TLS/HTTPS***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***[no other capabilities]***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

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

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [**transmitting audit events**].

5.2.9.2 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1 Refinement: The TSF shall use [***SSH, TLS/HTTPS***] **to** provide a trusted communication path between itself and remote administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and detection of modification of the communicated data.

FTP_TRP.1.2 Refinement: The TSF shall permit remote administrators to initiate communication via the trusted path.

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

5.3 TOE Security Assurance Requirements

The SARs have all been drawn verbatim from the NDPP. The security assurance requirements from the NDPP are specified in Part 3 of the Common Criteria. The "Assurance Activities" listed in section 5.4 of the ST are performed by the Common Criteria Testing Lab (CCTL) as part of the evaluation.

Table 11: Assurance Components

Assurance Class	Assurance Component
ADV: Development	ADV_FSP.1: Basic Functional Specification
AGD: Guidance Documents	AGD_OPE.1: Operational User Guidance
	AGD_PRE.1: Preparative User Guidance
ATE: Tests	ATE_IND.1: Independent Testing - Conformance
AVA: Vulnerability Assessment	AVA_VAN.1: Vulnerability Survey
ALC: Life-cycle Support	ALC_CMC.1: Labeling of the TOE
	ALC_CMS.1: TOE CM Coverage

5.3.1 Development (ADV)

5.3.1.1 Basic Functional Specification (ADV_FSP.1)

Developer action elements:

ADV_FSP.1.1D The developer shall provide a functional specification.

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

Content and presentation elements:

ADV_FSP.1.1C The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2C The functional specification shall identify all parameters associated with each SFR- enforcing and SFR-supporting TSFI.

ADV_FSP.1.3C The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements:

ADV_FSP.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.3.2 Guidance Documents (AGD)

5.3.2.1 Operational User Guidance (AGD_OPE.1)

Developer action elements:

AGD_OPE.1.1D The developer shall provide operational user guidance.

Content and presentation elements:

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

Evaluator action elements:

AGD_OPE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2.2 Preparative Procedures (AGD_PRE.1)

Developer action elements:

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

Content and presentation elements:

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

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational

environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements:

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

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

5.3.3 Tests (ATE)

5.3.3.1 Independent Testing - Conformance (ATE_IND.1)

Developer action elements:

ATE_IND.1.1D The developer shall provide the TOE for testing.

Content and presentation elements:

ATE_IND.1.1C The TOE shall be suitable for testing

Evaluator action elements:

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

ATE_IND.1.2E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.3.4 Vulnerability Assessment (AVA)

5.3.4.1 Vulnerability Survey (AVA_VAN.1)

Developer action elements:

AVA_VAN.1.1D The developer shall provide the TOE for testing.

Content and presentation elements:

AVA_VAN.1.1C The TOE shall be suitable for testing.

Evaluator action elements:

AVA_VAN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.1.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3E The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

5.3.5 Life-cycle Support (ALC)

5.3.5.1 Labeling of the TOE (ALC_CMC.1)

Developer action elements:

ALC_CMC.1.1D	The developer shall provide the TOE and a reference for the TOE.
	Content and presentation elements:
ALC_CMC.1.1C	The TOE shall be labeled with its unique reference.
	Evaluator action elements:
ALC_CMC.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5.2 TOE CM Coverage (ALC_CMS.1)

	Developer action elements:
ALC_CMS.1.1D	The developer shall provide a configuration list for the TOE.
	Content and presentation elements:
ALC_CMS.1.1C	The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.
ALC_CMS.1.2C	The configuration list shall uniquely identify the configuration items.
	Evaluator action elements:
ALC_CMS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.4 Assurance Activities

Please refer to the specific PPs for the defined assurance activities.

6. TOE Summary Specification

This section describes the security functions of the TOE:

- Security Audit
- Cryptographic Support
- User Data Protection
- Stateful Traffic Filtering
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

6.1 Security Audit

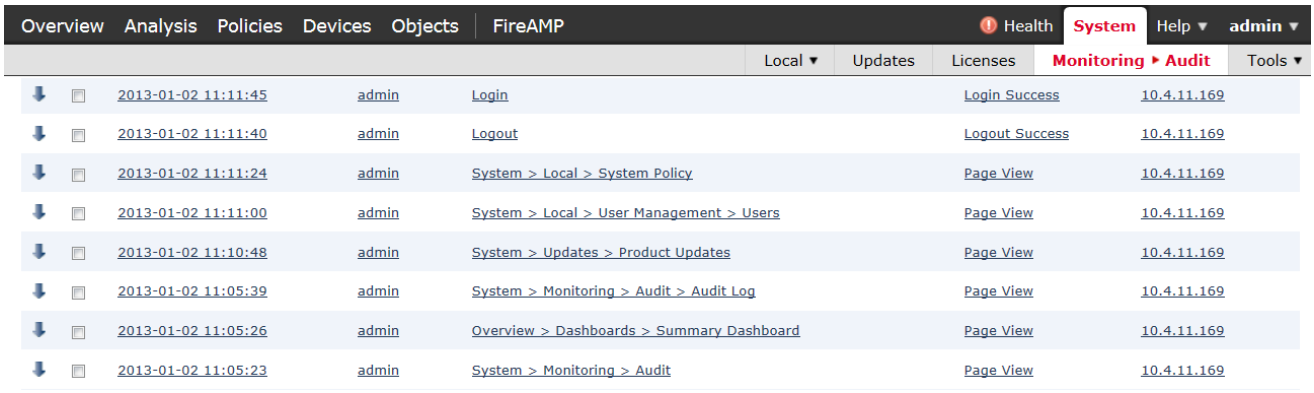
Auditing is the recording of events within the system. The TOE generates log records for a wide range of security relevant and other events as they occur. The events that can cause an audit record to be logged include starting the audit function[6], any use of an administrator command or action via the CLI and web interfaces, and all of the required auditable events identified in Table 12. For more information about the required audit events, please refer to Table 12 and the operational user guide (also known as the CC Supplemental User guide).

The TOE can record activity on the system in two ways. The system can generate an audit record for each user interaction with the web interface and each command in the CLI interface in the audit log, and can also record system status messages in the system log (i.e., SYSLOG). In addition, the TOE can generate traffic events as part of the intrusion and access control policies and these event records are stored in logs separate from the audit logs for performance and security reasons. More information about the traffic events is presented in section 6.4.

Defense Centers and managed devices log auditing information for all user activity in a read-only format. Audit logs are presented in a standard event view that allows administrators to view, sort, and filter audit log messages based on any item in the audit view. The audit view contains columns with information field for each audit event such as time, user, subsystem, message, and source IP. Please see the figure below for example.

[6] Note that the audit function cannot be disabled other than shutting down the entire system.

Figure 3: Audit View



The following fields are recorded for each audit event in the audit view:

- **Time**: The time and date that the appliance generated the audit record.
- **User**: The user name of the user that triggered the audit event.
- **Subsystem**: The menu path the user followed to generate the audit record. For example, "System > Monitoring > Audit" is the menu path to view the audit log.
- **Message**: The action the user performed. For example, "Page View" signifies that the user simply viewed the page indicated in the Subsystem, while "Save" means that the user clicked the Save button on the page.
- **Source IP**: The IP address of the host used by the user.

The user can also view the audit log using the command "show audit-log" via the CLI interface. All GUI actions and CLI commands are recorded in the audit log and can only be viewed by authorized administrators. To distinguish between the two, the Subsystem field will identify "Command Line" for commands and the Message field will identify the executed command.

In general, the logged audit records identify the date and time, the identity of the actor (e.g., user or network host) responsible for the event, the subsystem that triggers the event, an indication of whether the event succeeded, failed or had some other outcome (if applicable), and the source IP. The logged audit records also include event-specific content that includes at least all of the content required in Table 12.

The TOE includes an internal log database implementation that can be used to store and review audit records locally. However, the internal log only stores a maximum of 100,000 entries in the local database. When the audit log is full, the oldest audit records are overwritten by the newest audit records. To prevent the losing of critical audit records, the administrators can configure the system to transmit all the audit event logs in real-time over a secure HTTPS connection to an external audit server in the operational environment. When an audit event is generated, it is sent to the local database and external audit server simultaneously. This ensures that current audit events can be viewed locally while all events, new or old, are stored off-line as required by the NDPP.

Note that the protection of the audit records stored at the external audit server is the responsibility of the operational environment. The TOE is only responsible for the secure communication channel. It is recommended that the audit server is physically or logically separated (e.g., VLANs) from the other networks.

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

- FAU_GEN.1: The TOE can generate audit records for events include starting the audit function, administrator commands/actions, and all other events identified in Table 12. Furthermore, each audit record identifies the date/time, responsible subject/user, event type, outcome of the event, IP source, as well as the additional event-specific content indicated in Table 13.
- FAU_GEN.2: The TOE identifies the responsible user for each event based on the specific administrator or network entity (identified by IP address) that caused the event.
- FAU_STG_EXT.1: The TOE can be configured to transmit audit records to an external audit server over a secure communication channel.

6.2 Cryptographic Support

The TOE utilizes a FIPS 140-2 validated cryptographic module certificate #1051[7] (i.e., OpenSSL FIPS object module) providing supporting cryptographic functions. The algorithm implementations have been tested and validated on Sourcefire appliances in accordance to validation suites set by the Cryptographic Algorithm Validation Program (CAVP). The following algorithms have been FIPS validated in accordance with the identified standards:

Table 14: FIPS 140-2 Algorithms

Algorithms	Standards	Certificate Numbers
Asymmetric Key Generation		
• Domain parameter generation	NIST Special Publication 800-56B NIST Special Publication 800-57	1227
• Random number generation	See RBG below	
Encryption/Decryption		
• AES (128, 192, and 256 bits) in CBC mode	FIPS PUB 197 NIST SP 800-38A	2575
Cryptographic Signature Services		
• RSA Digital Signature Algorithm (rDSA) (modulus 2048)	FIPS PUB 186-2	1322
Cryptographic Hashing		
• SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512 (digest sizes 160, 224, 256, 384 and 512 bits)	FIPS PUB 180-3	2174
Keyed-hash Message Authentication		
• HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 (message digest sizes 160, 224, 256, 384, and 512 bits)	FIPS PUB 198-1 FIPS PUB 180-3	1598
Random Bit Generation (RBG)		
• RBG with independent software-based noise source of 128 bits	FIPS PUB 140-2 Annex C: X9.31 Appendix 2.4 using AES	1227

[7] Please see the Security Policy for more information: http://www.openssl.org/docs/fips/SecurityPolicy-1.2.2.pdf

The TOE only supports RSA in the evaluated configuration. RSA digital signature is used to sign HTTPS/TLS and SSH certificates. The TOE complies with NIST SP 800-56B, Recommendation for Pair-wise Key Establishment Schemes Using Integer Factorization Cryptography. The TOE does not implement any functionality that is identified as “shall not” or “should not” in the applicable sestions of the SP 800-56B. Specifically, the TOE complies strictly with the following applicable sections in the SP 800-56B:

5.1 (Cryptographic Hash Functions), 5.2 (Message Authentication Code Algorithm), 5.3 (Random Bit Generation), 5.4 (Prime Number Generators), 5.5 (Primality Testing Methods), 5.6 (Nounces), 5.9 (Key Derivation Functions for Key Establishment Schemes), 6.1 (RSA Key Pairs - General Requirements), 6.2 (Criteria for RSA Key Pairs for Key Establishment), 6.3 (RSA Key Pair Generators), 6.4 (Assurance of Validity), 6.5 (Assurance of Private Key Possession), 6.6 (Key Confirmation), and 8 (Key Agreement Schemes). The TOE complies with RSA key pair generation according to FIPS 186-2 in SP 800-56B. Note that FIPS 186-3 is referenced and recommended in SP 800-56B but it is not mandatory.

The TOE uses a software-based random bit generator that complies with FIPS 140-2 ANSI X9.31 Random Number Generation (RNG) operating in FIPS mode. The RNG ensures that the seed keys (using Approved AES as the underlying algorithm) and seed values are not the same. In addition, the RNG is seeded by an entropy source that is at least 128-bit value derived from various highly sensitive and proprietary noise sources described in a separate document. In future release of the TOE, a minimum of 256-bits of entropy will be used according to SP 800-90 Appendix B.

Additionally, the TOE is designed to zeroize secret and private keys when they are no longer required by the TOE. This zeroization mechanism is performed by overwritten the sensitive keys and data with all 0’s before deleting them. The following table identifies the applicable secret and private keys and summarizes, how they are generated, what are their purpose, where are they stored, and when and how are they deleted:

Table 15: Zeroization Methods

Name	Generation/ Algorithm	Purpose	Storage Location	Zeroization Summary
RSA public/private keys	ANSI X9.31	Identity certificates for the security appliance itself and also used in TLS, and SSH negotiations. The security appliance supports 2048 bit modulus key sizes or higher.	Private Key – hard disk (plaintext) and RAM (plain text) Public Key – hard disk (plaintext) and RAM (plain text)	Private Key - are zeroized then deleted from hard disk when the CA certificates are deleted by the administrators. Public Key - are deleted from hard disk when the CA certificates are deleted by the administrators.
Diffie-Hellman Key Pairs	ANSI X9.31	Key agreement for TLS, and SSH sessions.	RAM (plain text)	Keys in RAM will be zeroized upon resetting (i.e., terminating all sessions) or rebooting the appliance.
RSA public/private keys	RSA	For communication between the DC and managed devices.	Hard disk (plain text)/RAM (plain text)	Private Key - The private key is zeroized when the DC and managed devices are de-registered.

Name	Generation/ Algorithm	Purpose	Storage Location	Zeroization Summary
TLS Session Keys	DH / ANSI X9.31 Algorithm: AES	Used in HTTPS connections	RAM (plain text)	Keys in RAM will be zeroized upon rebooting the appliance.
SSH Session Keys	DH / ANSI X9.31 Algorithm: AES	SSH keys	RAM (plain text)	Keys in RAM will be zeroized upon rebooting the appliance.
Passwords	User generated	Critical security parameters used to authenticate the administrator login.	Hard disk (Hashed with SHA-512 and salt value)	Passwords are not stored in plaintext. Only the hashed of the passwords and a 32-bit nounces are stored.
Certificates of Certificate Authorities (CAs)	ANSI X9.31	Necessary to verify certificates issued by the CA. Install the CA's certificate prior to installing subordinate certificates.	Hard disk (plain text) and RAM (plain text)	CA certificates are zeroized from hard disk when the CA certificates are deleted by the administrators. CA certificates in RAM will be zeroized upon rebooting the security appliance.
PRNG Seed Key	Entropy	Seed key for X9.31 PRNG	RAM (plain text)	Seed keys are overwritten with the generation of new seed

The supporting cryptographic algorithms identified above are included to support the SSHv2 (RFCs 4251, 4252, 4253, and 4254) and TLSv1 (RFC 2246)/HTTPS (RFC 2818) security communication protocols. Note that IPSec communication protocol is out-of-scope.

The TOE supports TLSv1 with AES 128 or 256 bit symmetric ciphers in CBC mode, in conjunction with SHA and RSA. The following cipher suites are implemented by the TOE:

- TLS_RSA_WITH_AES_128_CBC_SHA,
- TLS_RSA_WITH_AES_256_CBC_SHA,
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA, and
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA.

While the OpenSSL supports additional cipher suites (for example, RSA_3DES_EDE_CBC_SHA, RSA_DES_CBC_SHA, RSA_RC4_128_MD5, RSA_RC4_128_SHA, etc.), they are all disabled while operating in CC evaluated configuration. If the client web browser does not support any of the required TLSv1 cipher suites, the TLSv1 connection will fail and the administrators will not establish a HTTPS web-based session with the TOE.

The TOE can be configured with “client-verify enable” in which case the client will be required to provide a certificate suitable for authentication via the TLSv1 protocol (i.e., mutual authentication). If that certificate-based authentication fails, no session will be established and the client user cannot further attempt to log in. If that succeeds, the user is still required to provide a username and password in order to log in to obtain access to security management functions. By default, the TLSv1 protocol only uses server authentication whereby the server must provides a trusted server certificate for authentication. Note that client authentication is the only optional TLSv1 characteristic implemented in the TOE.

The TOE supports SSHv2 with AES (in CBC mode) 128, 192, or 256 bit ciphers for encryption, in conjunction with HMAC-SHA for integrity and authenticity, and RSA with diffie-hellman-group14-sha1 for the key exchange method. While DES and 3DES, HMAC-MD5 and HMAC-MD5-96, and diffie-hellman-group-1 and other diffie-hellman-exchange groups are all implemented, they are disabled while the TOE is operating in CC evaluated configuration. In addition, SSHv1 is also disabled by default for security reasons. If the SSH client does not support the Approved algorithms and SSH version, the SSH connection will fail and the administrators will not establish an SSHv2 web-based session with the TOE.

The TOE uses OpenSSH implementation version 5.9p1 to support the SSHv2 connections. SSHv2 connections are rekeyed prior to reaching 2^{28} packets. The authentication timeout period is 90 seconds allowing clients to retry only 3 times. In addition, both public-key and password-based authentication can be configured with password-based being the default method used. The SSH packets are limited to 256 Kbytes. If OpenSSH detects packet larger than maximum (#define PACKET_MAX_SIZE (256 * 1024)), then it will dropped the packet. Note that the TOE manages a packet counter for each SSH session so that it can initiate a new key exchange when the 2^{28} packet limit is reached. Whenever the timeout period or authentication retry limit is reached, the TOE closes the applicable TCP connection and releases the SSH session resources. As SSH packets are being received, the TOE uses a buffer to build all packet information. Once complete, the packet is checked to ensure it can be appropriately decrypted. However, if it is not complete when the buffer becomes full (256 Kbytes) the packet will be dropped.

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

- FCS_CKM.1: The TOE generates Approved RSA public/private key pairs for key establishment to support other security protocols such as SSHv2 and TLSv1. The RSA modulus key size is 2048 bit, which according to NIST PUB 800-57, is equivalent to a symmetric key strength of 112 bits.
- FCS_CKM_EXT.4: Keys are zeroized when they are no longer needed by the TOE. Please see table above.
- FCS_COP.1(1): The TOE supports Approved AES symmetric algorithm for encryption and decryption of communication data, in support other security protocols such as SSHv2 and TLSv1. Please see table above.
- FCS_COP.1(2): The TOE supports Approved RSA digital signature algorithm for signature generation and verification, in support of digital certificates. Please see table above.
- FCS_COP.1(3): The TOE supports Approved SHA hashing algorithm for hashing of communication data, in support other security protocols such as SSHv2 and TLSv1. Please see table above.
- FCS_COP.1(4): The TOE supports Approved HMAC-SHA message authentication algorithm for authenticating of communication data, in support other security protocols such as SSHv2 and TLSv1. Please see table above.
- FCS_HTTPS_EXT.1: The TOE supports HTTPS web-based secure administrator sessions.
- FCS_RBG_EXT.1: The TOE uses Approved ANSI X9.31 Appendix 2.4 implementation to generate random numbers for generating cryptographic keys, and seeds the RNG with a software-based entropy source.

- FCS_SSH_EXT.1: The TOE supports SSHv2 interactive CLI-based administrator sessions as described above.
- FCS_TLS_EXT.1: The TOE supports the required cipher suites and provides the secure transport protocol for the HTTPS web-based secure administrator sessions.

6.3 User Data Protection

The TOE is designed to ensure its own internal integrity as well as to protect user data from potential, unintended reuse by clearing resources (e.g., memory) as they are allocated to create objects used in the implementation of the TOE operations.

The system implements thousands of separate packet buffers which are fixed length long each for each data interface. For each packet, the system tracks the length of packet field and will ensure enough packet buffers are reserved for that packet. For example, if a packet is 1 KB long, the system will only use 1 KB of the packet buffer. The buffers are not overwritten until the next new packet data is written to the buffer. Note that if the new packet is over 1 KB, then the old data will be completely overwritten. Otherwise, the old data can still be present but the length of the packet field will ensure that the old, residual data is not put in the new packet.

The User data protection function is designed to satisfy the following security functional requirements:

- FDP_RIP.2: The TOE overwrites resources when allocated for use in objects and ensures no residual data is accidentally used.

6.4 Stateful Traffic Filtering

The TOE supports access control policy with rules which provide granular control over how the system handles and logs network traffic. For each rule, administrators can specify a rule action, for example, to permit (i.e., allow), deny (i.e., block), and/or log matching traffic. Each rule contains a set of conditions that identify the specific traffic administrators want to control. Rules can be simple or complex, matching traffic by any combination of attributes such as security zone, IPv4 and IPv6 addresses, ICMPv4 and ICMPv6 type and code, transport protocol, and ports.

The TOE is designed to permit network packets to flow only when the ruleset contains a rule that permits the flow, or the packet is deemed to belong to an established connection that has been permitted to flow. During start-up and initialization, the TOE runs a series of system checks and self-tests to ensure the system is functioning correctly. The hardware network cards are then powered up and configured with the appropriate values (e.g., IP, VLAN tag, etc.). The Snort engine is then immediately started and will inspect the packets flowing through the network cards. There is a split second where packets may flow uninspected between the time when the network cards are powered up and the Snort engine is started. The CC Supplemental User Guide will instruct the administrators to perform any maintenance or upgrade process during off-peak hours and disconnect the appliance from the network during the process. . In addition, the administrators should configure the inline interfaces to fail secure[8] to prevent packets from flowing through the system without being processed in the event of a component failure (e.g., Snort engine failure, buffers or memory overload, etc.). In this case, failure will result in packets being dropped. By default, the inline interfaces are set to fail open.

The TOE can be configured to detect when “too much traffic” is coming from a network against the interface and drop that the traffic and generate event. The rate-based configuration setting is used to define the number of packets over the specified period of time (e.g., 1,000,000 packets over 20 seconds). If the traffic rate falls below the configured rate, the traffic will not be dropped automatically. Under no circumstances will the TOE pass packets that do not satisfy a rule that permit them or belong to an allowed established session.

The system matches traffic to rules in top-down order by rule number. In addition to its rule order and some other basic attributes, each rule has the following major components:

- a set of rule conditions that identifies the specific traffic
    - Security zones such as internal and external or trusted and untrusted.
    - IPv4 and IPv6 addresses including range of addresses or “any” address
    - Source and destination ports from 1 – 65,535
    - ICMPv4 and ICMPv6 code and types
    - Network protocols
- a rule action, which determines how the system handles traffic that meets the rule’s conditions
- intrusion inspection options, which examine (and optionally block) matching traffic[9]
- logging options, which allow to keep a record of the matching traffic

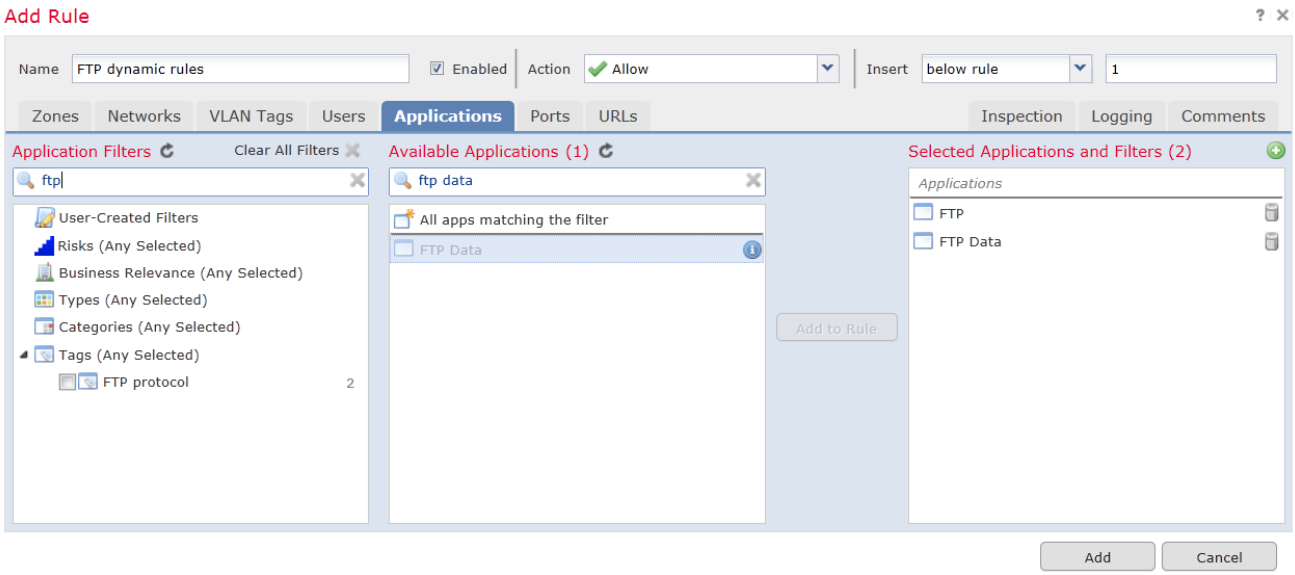
[8] In an inline deployment, this setting “fail safe” can cause a Denial of Service (DoS) attack on the monitored network.

[9] Note that the “Protection” License needs to be installed to associate a rule with intrusion policies.

The access control policy's default action handles traffic that does not match the ruleset. The default action is to block all traffic unless implicitly allowed by an established connection. In addition to access control policy, the system supports intrusion policies that detect anomalies such as malformed packets or fragmentation attacks.

The TOE supports creation of dynamic rules to permit FTP connections when the data ports are not known or are random at creation time. The implementation uses FTP application detector and access control rules to support this requirement. Logging can be enabled for the rule if necessary. Note that to use the Application tab for the access control rules, CONTROL license is required which requires PROTECTION license. The CONTROL license is required to use our application-level access control feature because to use dynamic rule, application-level control is required. By allowing FTP and FTP data in the rule, the TOE is smart enough to detect FTP data connections, allow those connections and log them, if configured using the Logging tab. The following figure below illustrates the setting for the access control rule to detect and permit FTP and FTP data connections:

Figure 4: FTP Rule



Before packets can be inspected, the packets must be captured from the network. As the system captures packets, it sends them to the packet decoder. The packet decoder converts the packet headers and payloads into a format that can be easily used by the preprocessors and the rules engine. Each layer of the TCP/IP stack is decoded in turn, beginning with the data link layer and continuing through the network and transport layers, as described in the following table.

TCP/IP Layer	Decoded Packets
Data Link	Ethernet
	Virtual local area network (VLAN)
Network	Internet Protocol version 4 (IPv4)
	Internet Protocol version 6 (IPv6)
	Internet Control Message Protocol version 4 (ICMPv4)

	Internet Control Message Protocol version 6 (ICMPv6)
Transport	Transmission Control Protocol (TCP)
	User Datagram Protocol (UDP)

After the packets are decoded through the first three TCP/IP layers, they are sent to preprocessors, which normalize traffic at the application layer and detect protocol anomalies. The following three preprocessors must be enabled and configured in the evaluated configuration:

- TCP Streaming Preprocessor - Administrators can configure the system so that the preprocessor detects any TCP traffic that cannot be identified as part of an established TCP session. Stateful inspection allows administrators to ignore these packets because they are not part of an established TCP session and do not provide meaningful information.
- UDP Streaming Preprocessor - UDP data streams are not typically thought of in terms of sessions. However, the stream preprocessor uses the source and destination IP address fields in the encapsulating IP datagram header and the port fields in the UDP header to determine the direction of flow and identify a session.
- IP Defragmentation Preprocessor - When an IP datagram is broken into two or more smaller IP datagrams because it is larger than the maximum transmission unit (MTU), it is fragmented. A single IP datagram fragment may not contain enough information to identify a hidden attack. Attackers may attempt to evade detection by transmitting attack data in fragmented packets or attempt to crash the system when reassembling the fragmented packets. The IP defragmentation preprocessor reassembles fragmented IP datagrams, and if fragmented datagrams cannot be reassembled, they will be rejected (i.e., dropped). The TOE reassembles fragmented IP datagrams before the rules engine executes rules against them. There is a configurable ***timeout*** value which specifies the maximum amount of time, in seconds, that the TOE can use when reassembling a fragmented packet. If the packet cannot be reassembled within the specified time period, the TOE will stop attempting to reassemble the packet and will discard received fragments.

After the packets have passed through the preprocessors, they are sent to the rules engine. The rules engine inspects the packet headers and payloads to determine whether they trigger rules.

The TOE can be configured to detect and log the following traffic conditions:

1. The TSF shall reject and be capable of logging packets which are invalid fragments (by IP Defragmentation Preprocessor);
2. The TSF shall reject and be capable of logging fragmented IP packets which cannot be re-assembled completely (by IP Defragmentation Preprocessor);
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 (by administrator-defined access control rules);
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 (by administrator-defined access control rules);
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 (by administrator-defined access control rules);
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 (by administrator-defined access control rules);

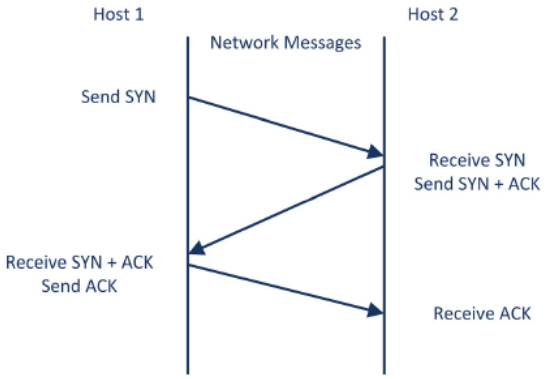
7. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being a loopback address (by administrator-defined access control rules);
8. The TSF shall reject and be capable of logging network packets where the source address of the network packet is a multicast (by administrator-defined access control rules);
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 (by administrator-defined access control rules);
10. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as being an address "reserved for future use" as specified in RFC 5735 for IPv4 (by administrator-defined access control rules);
11. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as an "unspecified address" or an address "reserved for future definition and use" as specified in RFC 3513 for IPv6 (by administrator-defined access control rules);
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 (by administrator-defined access control rules);

The CC Supplemental guide will instruct administrators on how to create and order the access control rules to optimally detect and log the default traffic conditions above.

- FFW_RUL_EXT.1.1: The TOE supports filtering based on stateful inspection and session tracking.
- FFW_RUL_EXT.1.2: The TOE supports the required RFCs: RFC 792 (ICMPv4), RFC 4443 (ICMPv6), RFC 791 (IPv4), RFC 2460 (IPv6), RFC 793 (TCP), and RFC 768 (UDP). Sourcefire products have been certified to meet these RFCs and others by DoD Joint Interoperability Test Command (JITC) and third-party independent ICSA labs. While the certified versions are older, Sourcefire asserted the conformance to RFCs remain the same in the CC version.
  - http://jitc.fhu.disa.mil/tssi/cert_pdfs/sourcefire_3d_ips_ids_aug12.pdf
  - https://www.icsalabs.com/technology-program/ipv6-/-usgv6/usgv6-tested-products
- FFW_RUL_EXT.1.3: The TOE supports the rule conditions to match traffic. Rules can be simple or complex, matching traffic by any combination of attributes such as security zone, IPv4 and IPv6 addresses, ICMPv4 and ICMPv6 type and code, transport protocol, and ports.
- FFW_RUL_EXT.1.4: The TOE supports the following rule actions including permit (allow), deny (block), and/or log.
- FFW_RUL_EXT.1.5: The TOE applies the policies to the interface of the managed devices. The administrator can create interface sets on the device and apply the access control policy to those interfaces on the device. In addition, administrator can create multiple access control policies and apply them to different interfaces on different devices. The only restriction is only one access control policy can be applied to one interface at anytime.
- FFW_RUL_EXT.1.6: The TOE tracks TCP sessions based on TCP flags and sequence number (inherently when the stream preprocessors are enabled) in addition to source and destination IP, IP protocol (TCP), and source and destination port. For ICMP, the TOE tracks the source and destination addresses and type and code. For example, ping reply is allowed

in response to a ping request. Stateful inspection evaluates the traffic that is part of a TCP session established with a legitimate three-way handshake between a client and server. The following figure below illustrates a legitimate three-way handshake:

Figure 5: Three-way Handshake



Once the TCP connection is terminated normally via FIN or RST flag, packets on the same 5-tuple (Source/Destination IP, Source/Destination Port, Protocol Attributes) are treated as not part of the connection and will be dropped if not explicitly allowed by rule. The TOE supports a feature called Strict TCP Enforcement. To maximize TCP security, the administrator can enable strict enforcement, which blocks connections where the three-way handshake was not completed. Strict enforcement also blocks:

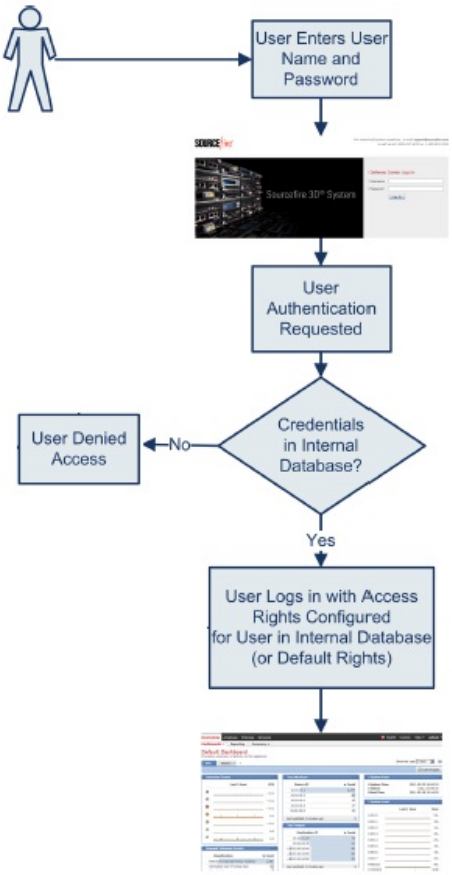
- non-SYN TCP packets for connections where the three-way handshake was not completed
- non-SYN/RST packets from the initiator on a TCP connection before the responder sends the SYN-ACK
- non-SYN-ACK/RST packets from the responder on a TCP connection after the SYN but before the session is established
- SYN packets on an established TCP connection from either the initiator or the responder

- FFW_RUL_EXT.1.7: The TOE supports creation of dynamic rules to permit FTP connections when the data ports are not known or are random at creation time.
- FFW_RUL_EXT.1.8: The TOE supports dropping of invalid and malformed fragmented packets (by preprocessor); fragmented packets that cannot be reassembled (by preprocessor); All other can be detected and logged by access control rules.
- FFW_RUL_EXT.1.9: The TOE applies the rules in a top-down order by rule number defined by the administrators. Rules in an access control policy are numbered, starting at 1. The system matches traffic to access control rules in top-down order by ascending rule number.
- FFW_RUL_EXT.1.10: The TOE supports creation of access control policy with default action of “Block all traffic” selected. This option will block all traffic—not permitted by ruleset or part of established connections—from entering the network. When a packet enters the device, the packet is examined to see if it is part of an allowed established session using the 5-tuple set. If not, the packet is analyzed against the rule set using a top to down approach. If no rule matches, then the default action rule is applied.

6.5 Identification and Authentication

The TOE is designed to successfully identify and authenticate user before allowing access to the TOE's security function. When identification and authentication data is entered (username and password), the TOE attempts to identify the applicable user account from the provided identity and if a match is found, the password provided is hashed with a salt value and compared against the stored hash[10] with the user account information in the internal database. If a user account cannot be associated with the provided identity or the hashed password does not match that stored hash with the user account information, the process will fail. No actions are allowed, other than re-entry of identification and authentication data or viewing the login banner. Once the user has successfully log in, the privilege level or role will control what management functions he or she has access and authorization to. Figure below shows the authentication process.

Figure 6: Authentication Process



Users can connect to the TOE via a local console or remotely using SSHv2 or HTTPS. In each case, the user is required to log in prior to successfully establishing a session through which TOE security functions can be performed. By default, the Sourcefire 3D System uses internal authentication to

[10] The password is hashed with Approved SHA-512 and the salt value is 32-bit long.

check user credentials when a user logs in. Alternately, the TOE can be configured to use an external authentication server for user identification and authentication. In the evaluated configuration, the use of an external authentication server such as RADIUS or LDAP is not allowed.

When logging in, the TOE will not echo passwords such that passwords are not inadvertently displayed to the user and any other users that might be able to view the login display. The TOE replaced the entered password character with a "*" character or not show any character at all. This depends on where the user is logging in from, for example, using web GUI versus the SSH client. If the authentication fails, the TOE is designed to not indicate either the username and/or password were incorrect. The error message would just state access denied or unable to authorize access. No other information about the failed login in can be ascertained from the error message.

Note also that should a user have their session terminated (e.g., due to inactivity), they are required to successfully re-authenticate, by reentering their identity and authentication data, in order to gain access to their session. The authentication data is not cached by the TOE for any reason.

When creating or changing passwords, the passwords must be composed of upper and lower case letters, numbers and special characters including blank space and ~`!@#$%^&*()_+-={}|[]\:";'<>,./?. The password must have at least one upper case, one lower case, one number, and one special character. This is configured by checking on "Enable Password Strength[11]" option per each user (See CC Supplement User Guide for details). Also, the passwords have to satisfy configured minimum password length which is set in the System Policy for all users. The minimum password length can range from 8 (default) to 32 characters (maximum) long, which includes 15 characters required by the NDPP. Note: The user password is limited to 32 characters maximum.

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

- FIA_PMG_EXT.1: The TOE implements a rich set of password composition and aging constraints as described above.
- FIA_UIA_EXT.1: The TOE requires all users to be identified and authenticated successfully before allowing access to the TOE security function. The only action allowed before is viewing the login banner.
- FIA_UAU_EXT.2: The TOE can be configured to utilize local authentication or optionally, external RADIUS and LDAP authentication servers.
- FIA_UAU.7: The TOE does not echo passwords as they are entered. The character is either replaced with "*" or not shown at all.

[11] This option also prevents word that appears in dictionary or include consecutive repeating characters.

6.6 Security Management

The TOE provides a web-based GUI (using HTTPS) management interface and limited CLI or shell (using SSH or serial connection) for all TOE administration, including the policy rule sets, user accounts and roles, and audit functions. The ability to manage various security attributes, system parameters and all TSF data is controlled and limited to those users who have been assigned the appropriate administrative role and privileges associated with those roles. Note that all users created are TOE administrators.

Predefined User Roles

The TOE supports the following predefined user roles:

- **Administrators** can set up the appliance's network configuration, manage user accounts, and configure system policies and system settings. The Administrator Role provides access to analysis and reporting features, rule and policy configuration, system management, and all maintenance features. Users with the Administrator role have ALL access rights.

Note: For all sensors, the only TOE user role is "Administrator". This role is granted when a new user account is created and cannot be changed.

CLI and CLI Access levels

The administrator can use the CLI to view, configure, and troubleshoot the Sourcefire systems. When administrators create a user account, they can assign it one of the following CLI access levels:

- **Basic** The user has read-only access and cannot run commands that impact system performance.
- **Configuration** The user has read-write access and can run commands that impact system performance.
- **None** The user is unable to log in.

Note that the CLI contains only a subset of all available functions. Therefore, the web-based GUI is highly recommended for management of the TOE. Local access to the shell which allows access to the underlying operating system is allowed in the CC evaluated configuration for the initial configuration only. The local management of the TOE is allowed for configuring SSH and TLS allowed ciphersuites, SSH timeout value and version, public-key authentication method, and viewing the self-test log. For normal daily operations, the web GUI is still the recommended method.

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

- FMT_MTD.1: The TOE restricts the access to manage TSF data that can affect the security functions of the TOE to Security Administrators (i.e., administrator roles). The TSF data here includes user accounts and roles, login banner, inactivity timeout values, password complexity setting, TOE updates, audit records, and audit server information.
- FMT_SMF.1: The TOE includes the functions necessary to administer the TOE locally and remotely, to manage login banner, and to manage and verify updates of the TOE software and firmware. Please see the CC Supplemental Guide for more information.

- FMT_SMR.1: The TOE includes one evaluated role which corresponds to the required ‘Security Administrator’ described above.

6.7 Protection of the TSF

The TOE is designed to communicate securely with itself (i.e., TOE components) and components in the operation environment. The communication between the TOE and the administrators is either protected by physical security (e.g., local connection with serial port) or by SSHv2 or HTTPS security protocols. The communication between the TOE components is protected by TLSv1 security protocol. The communication between the TOE and the audit server is protected by HTTPS security protocol. Protocol failures due to issues such as version mismatch (e.g., attempting to use SSHv1) or unsupported ciphersuites (e.g., using weak TLS ciphersuite) will be recorded by the TOE.

The TOE is designed to not to disclose or store plaintext passwords (e.g., passwords are never recorded in the audit records or display during authentication process). The passwords are stored hashed using Approved SHA-512 with a 32-bit salt value. Only 'root' user account with access to the shell can view the hashed passwords and this is prohibited in the evaluated configuration. The same is true for cryptographic keys such as encryption symmetric keys and RSA private keys. The public keys can be viewed but cannot be modified without detection. Note that access to RSA public keys is restricted to administrators.

The TOE is a hardware appliance that includes a hardware-based real-time clock. The TOE's embedded OS manages the clock and the GUI exposes the clock management function to the administrators. Optionally, the TOE can be configured to use a NTP server in the operational environment. The time source is updated frequently from the time server to ensure accuracy. The time is used for the timestamp in the audit records and events.

The TOE includes a number of built in diagnostic tests that are run during start-up to determine whether the TOE is operating properly. The built-in BIOS self tests include basic read-write memory, flash read, software checksum tests, and device detection tests. In addition, the TOE is designed to run the power-on self-tests that comply with the FIPS 140-2 requirements for self testing (e.g., known answer tests (KATs) and zeroization tests). If the TOE fails any of the FIPS power-on self-tests, the TOE will enter an error state and will not be operational. The following self-tests are executed: AES encryption/decryption KAT, RSA key generation and encryption/decryption KAT, SHA hash KATs, HMAC-SHA hash KATs, PRNG KATs, and key overwriting tests.

The TOE can be updated manually or automatically[12]. For manual update, the user will download the TOE upgrade file, compute the hash of the upgrade file, and verify the computed hash matches the hash published on the secure website. Each upgrade file will have a corresponding published SHA-512 hash value with it. Note that the published hash value is also embedded with the TOE upgrade file in case the users choose not to use the manual update method. The TOE also includes a validity checking function that is always run when upgrading the TOE firmware (including patches) and Sourcefire Rule Updates (SRUs). This ensures TOE updates are always validated prior to installation. In either case,manual or automatic, the upgrade version will be checked to ensure it is appropriate (e.g., not upgrading to an older version) and the upgrade file will be verified using an embedded SHA-512 hashed value verified against the value computed during upgrade. If the version is incorrect or the SHA-512 hashed value cannot be verified, the upgrade will not proceed in order to protect the integrity of the TOE. More specifically, each update includes a header and metadata with the version and hashed value. In order to verify the data, the TOE generates its own SHA-512 secure has of the update data, compares it with the embedded hash in the update header to ensure they match.

[12] This process requires access to the Internet and is out of scope of the evaluation.

During the update process, if the Snort engine is updated and restarted, then is a split second where the managed devices do not perform any traffic inspection on the network. The CC Supplemental user guide will address this situation by requiring the upgrade and maintenance actions be performed during off-peak hours where the appliance can be disconnected from the network during the upgrade process to be upgraded, restarted, and verified before re-connecting back to the network to ensure complete traffic inspection.

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_ITT.1: The TOE components are designed to communicate through TLSv1 communication channels.
- FPT_SKP_EXT.1: The TOE does not offer any functions that will disclose to any users a cryptographic key. The protection provided by the TOE is that there is no interface available. Only 'root' user account with access to the shell can potentially view the stored keys and this is prohibited in the evaluated configuration.
- FPT_AWP_EXT.1: The TOE does not offer any functions that will disclose to any user a plaintext password. The TOE never store passwords in the plaintext.
- FPT_STM.1: The TOE includes its own hardware clock which the administrators can manually set. Optionally, the administrator can configure the TOE to get the time from a trusted NTP server in the operational environment.
- FPT_TST_EXT.1: The TOE includes a number of power-on diagnostics that will serve to ensure the TOE is functioning properly. The tests include ensure memory and flash can be accessed correctly as expected, and to ensure that cryptographic functions are operating normally.
- FPT_TUD_EXT.1: The TOE provides functions to query and upgrade the versions of the TOE firmware (including installing patches/hotfixes) and SRUs. SHA-512 cryptographic hashes are used to ensure the integrity of each upgrade prior to performing the upgrade.

6.8 TOE Access

The TOE can be configured to display administrator-configured advisory banners that will appear when users initiate an interactive session with the TOE. The login banner can be configured in the system policy (System > Local > System Policy > “Policy Name” > Login Banner) and can be applied to DC itself and push out all its managed devices by the administrator. The login banner can be configured to display welcome information or legal in conjunction with login prompts. In each case, the banners will be displayed when accessing the TOE via the local console/serial, SSHv2, or HTTPS interfaces.

The TOE can be configured by an administrator to set an interactive session timeout value in the System Policy, as with the login banner. The System Policy applies to all users and for both local and remote interactive sessions. The timeout value can be any positive integer value from 1 minute to 1,440 minutes (24 hours), with 0 disabling the timeout – the default timeout is 60 minutes. The administrators can configure an exemption to the timeout feature on a per user basis. This means that the user will be exempted from the being timeout. This option is not allowed in the evaluated configuration and the administrators are advised in the CC Supplement User Guide against using this option.

A remote or local session that is inactive (i.e., no commands or actions from the remote client) for the defined timeout value will be terminated and logged by audit function. The user will be required to re-enter their username and their password to start another session. The users can also terminate their own interactive local or remote sessions, anytime they choose.

The TOE access function is designed to satisfy the following security functional requirements:

- FTA_SSL_EXT.1: The TOE terminates local sessions that have been inactive for an administrator-configured period of time. Terminated sessions are disconnected from the local console input/output functions and can be reconnected only if the locked user correctly re-authenticates their username and password.
- FTA_SSL.3: The TOE terminates remote sessions that have been inactive for an administrator-configured period of time.
- FTA_SSL.4: The TOE provides a logout option for users to terminate their own sessions when they choose.
- FTA_TAB.1: The TOE can be configured to display administrator-defined advisory banners when administrators successfully establish interactive sessions with the TOE.

6.9 Trusted Path/Channels

The TOE can be configured to transmit audit records to an external audit server. In order to protect exported audit records from disclosure or modification, the TOE utilizes HTTPS connections. The HTTP is an application-level protocol and runs on top of TLSv1, a transport-level security protocol. The TLSv1 provides authentication, key exchange, encryption and integrity protection of the data. For every audit event generated, the TOE stores it locally and performs a HTTP POST to send it to the audit server. The administrators can enable and configure this feature to provide the HTTPS URL to specify where to post the audit logs. All the cryptographic algorithms and functions are provided by OpenSSL.

To support secure remote administration, the TOE includes implementations of SSHv2 (by OpenSSH) and HTTPS (HTTP over TLSv1). In each case, a remote host (presumably acting on behalf of an administrator) can initiate a secure remote connection for the purpose of security management. Note that only the local console is available by default and each of these remote administration services can be independently enabled by an administrator. For added security, only these security protocols and ports 22 and 443 are enabled and allowed by default. The administrators can also setup an access list to restrict only allowed IP addresses to access the TOE.

In the cases of SSHv2 and HTTPS, the TOE offers both a secure command line interface (CLI) and a graphical user interface (GUI) interactive administrator sessions. An administrator with appropriate SSHv2 or HTTPS capable clients can establish secure remote connections with the TOE. However, to successfully establish such an interactive session, the administrator must be able to provide acceptable user credentials (e.g., user name and password), after which they will be able to issue commands or actions within their assigned authorizations.

All of the security protocols are supported by the cryptographic operations provided by the FIPS certified cryptomodule included in the TOE implementation.

The Trusted Path/Channels function is designed to satisfy the following security functional requirements:

- FTP_ITC.1: The TOE can be configured to use HTTPS to ensure that any transmitted audit records are protected from tampering, and are sent only to the configured audit server so they are not subject to inappropriate disclosure or modification.
- FTP_TRP.1: The TOE provides SSH and HTTPS, based on its embedded FIPS cryptomodule, to support secure remote administration. In each case, the administrator can initiate the remote session, the remote session is secured (disclosure and modification) using FIPS certified cryptographic operations, and all remote security management functions require the use of one of these secure channels.

7. Protection Profile Claims

This ST is conformant to the *Security Requirements for Network Devices, 08 June 2012, Version 1.1* and *Network Device Protection Profile (NDPP) Extended Package Stateful Traffic Firewall, 19 December 2011, Version 1.0* – with the conditional HTTPS, SSH, and TLS requirements from NDPP.

The TOE is a network infrastructure device with Intrusion Protection and Stateful Firewall capabilities. Section 1.1 of the NDPP, Compliant Targets of Evaluation, states "Examples of a "network device" that should claim compliance to this PP include routers, firewalls, IDSs, audit servers, and switches that have Layer 3 functionality." As such, the ST is correct in claiming conformance to both the NDPP and EPFW.

As explained in section 3, Security Problem Definition, the Security Problem Definition of the NDPP and EPFW has been copied verbatim into this ST.

As explained in sections 4, Security Objectives, the Security Objectives of the NDPP and EPFW have been copied verbatim into this ST.

As explained in section 5, IT Security Requirements, each SFR is drawn verbatim from the NDPP and EPFW, while the SAR is drawn verbatim from the NDPP. Table 9 in section 5.2 identifies all the SFRs in this ST, and the PP the SFR was drawn from. If changes were made to the SFR, it was mainly due to clarify the requirement—not to change its meaning. The changes are identified and explained in the table below.

Table 16: Requirement Change Rationale

Requirement Component	Rationale
FAU_GEN.1: Audit Data Generation	• Combine required auditable events from the two claimed PPs. • "Table 1" is replaced with correct table reference in the ST. • New column is added to identify where the SFR belongs to.
FMT_SMF.1: Specification of Management Functions	• Combine security management functions from the two claimed PPs.
FTP_TRP.1: Trusted Path	• Grammatical Error Fix

No change was made to the SAR. The only change made to the "Assurance Activities" was to fix the references.

8. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Requirement Dependencies;
- TOE Summary Specification.

8.1 Security Objectives Rationale

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat.

8.1.1 Security Objectives Rationale for the TOE and Environment

This section provides evidence demonstrating the coverage of organizational policies and usage assumptions by the security objectives. Note that the NDPP does not explicitly or clearly correspond or rationale correspondence between its Security Problem Definition and Security Objectives, so the mapping had to be inferred and correspondence rationale has been devised to complete this ST appropriately.

Table 17: Environment to Objective Correspondence

	P.ACCESS_BANNER*	T.ADMIN_ERROR*	T.NETWORK_ACCESS	T.NETWORK_DISCLOSURE	T.NETWORK_DOS	T.NETWORK_MISUSE	T.TSF_FAILURE*	T.UNAUTHORIZED_ACCESS*	T.UNAUTHORIZED_UPDATE*	T.UNDETECTED_ACTIONS*	T.USER_DATA_REUSE*	A.CONNECTIONS	A.NO_GENERAL_PURPOSE*	A.PHYSICAL*	A.TRUSTED_ADMIN*
O.ADDRESS_FILTERING			X	X	X	X									
O.DISPLAY_BANNER*	X														
O.PORT_FILTERING			X	X	X	X									
O.PROTECTED_COMMUNICATIONS*								X							
O.RELATED_CONNECTION_FILTERING			X												
O.RESIDUAL_INFORMATION_CLEARING*											X				
O.SESSION_LOCK*								X							

	P.ACCESS_BANNER*	T.ADMIN_ERROR*	T.NETWORK_ACCESS	T.NETWORK_DISCLOSURE	T.NETWORK_DOS	T.NETWORK_MISUSE	T.TSF_FAILURE*	T.UNAUTHORIZED_ACCESS*	T.UNAUTHORIZED_UPDATE*	T.UNDETECTED_ACTIONS*	T.USER_DATA_REUSE*	A.CONNECTIONS	A.NO_GENERAL_PURPOSE*	A.PHYSICAL*	A.TRUSTED_ADMIN*
O.STATEFUL_INSPECTION					X										
O.SYSTEM_MONITORING*		X				X		X		X					
O.TOE_ADMINISTRATION*								X							
O.TSF_SELF_TEST*							X								
O.VERIFIABLE_UPDATES*									X						
OE.CONNECTIONS												X			
OE.NO_GENERAL_PURPOSE*													X		
OE.PHYSICAL*														X	
OE.TRUSTED_ADMIN*															X

* - NDPP

8.1.1.1 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.

This Organizational Policy is satisfied by ensuring that:

- O.DISPLAY_BANNER: To fulfill the policy for displaying advisory information to users prior to their use of the TOE, the TOE is required to display a configured banner when users login to establish an interactive session.

8.1.1.2 T.ADMIN_ERROR

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

This Threat is satisfied by ensuring that:

- O.SYSTEM_MONITORING: To reduce the potential of an administrative error that might be unnoticed or untraceable, the TOE is required to log security relevant events and transmit those logs to external log server.

8.1.1.3 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.

This Threat is satisfied by ensuring that:

- O.ADDRESS_FILTERING: To reduce the potential of an unauthorized access, the TOE is required to be able to filter[13] packets based on source and destination network addresses, and log the packet if rule is met.
- O.PORT_FILTERING: To reduce the potential of an unauthorized access to services, the TOE is required to be able to filter packets based on source and destination network ports, and log the packet if rule is met.
- O.RELATED_CONNECTION_FILTERING: To distinguish unauthorized access from authorized access, the TOE is required to track and dynamically permit network traffic flow that was a result of an allowed connection.

8.1.1.4 T.NETWORK_DISCLOSURE

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

This Threat is satisfied by ensuring that:

- O.ADDRESS_FILTERING: To reduce the potential of an unauthorized disclosure, the TOE is required to be able to filter packets based on source and destination network addresses, and log the packet if rule is met.
- O.PORT_FILTERING: To reduce the potential of an unauthorized disclosure of sensitive data, the TOE is required to be able to filter packets based on source and destination network ports, and log the packet if rule is met.

8.1.1.5 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.

This Threat is satisfied by ensuring that:

- O.ADDRESS_FILTERING: To mitigate the potential of a Denial of Service (DoS) attack, the TOE is required to be able to filter packets based on source and destination network addresses, and log the packet if rule is met. Thus, permitting packets from a trusted source to the allowed destination only based on ruleset and blocking everything else by default.
- O.PORT_FILTERING: To mitigate the potential of a DoS attack on a service, the TOE is required to be able to filter packets based on source and destination network ports, and log the packet if rule is met. Thus, permitting packets from a trusted source to the allowed destination only based on ruleset and blocking everything else by default.

[13] The term 'filter' here is used to define several configurable actions such as permit, deny, log, etc.

- O.STATEFUL_INSPECTION: To mitigate the potential of a DoS attack, the TOE is required to track and only permit network packets that belong to an established and allowed connection.

8.1.1.6 T.NETWORK_MISUSE

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

This Threat is satisfied by ensuring that:

- O.ADDRESS_FILTERING: To reduce the potential of network misuse, the TOE is required to be able to filter packets based on source and destination network addresses, and log the packet if rule is met.
- O.PORT_FILTERING: To reduce the potential of service misuse, the TOE is required to be able to filter packets based on source and destination network ports, and log the packet if rule is met.
- O.SYSTEM_MONITORING: To reduce the potential of network and service misuses that may go unnoticed, the TOE is required to log security relevant events and transit those logs to external log server.

8.1.1.7 T.TSF_FAILURE

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

This Threat is satisfied by ensuring that:

- O.TSF_SELF_TEST: To reduce the potential for undetected TOE failures and to help ensure that the TOE security functions are operating properly, the TOE is required to perform self-tests.

8.1.1.8 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.

This Threat is satisfied by ensuring that:

- O.PROTECTED_COMMUNICATIONS: To reduce the potential that an attacker might gain unauthorized access to the TOE or its data via data transmitted across a network, the TOE is required to protect its communication channels.
- O.SESSION_LOCK: To reduce the potential for unauthorized access to TOE security functions and data, the TOE is required to lock or terminate unattended or inactive sessions.

- O.SYSTEM_MONITORING: To reduce the potential of unauthorized access attempts that might go unnoticed, the TOE is required to log security relevant events and transit those logs to external log server.
- O.TOE_ADMINISTRATION: To reduce the potential of unauthorized access to TOE security functions and data, the TOE is required to ensure that only authorized administrators can log in and access security management functions and TSF data.

8.1.1.9 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.

This Threat is satisfied by ensuring that:

- O.VERIFIABLE_UPDATES: To reduce the potential that an update might contain malicious or unintended features, the TOE is required to provide mechanisms that serve to ensure the integrity of updates prior to their use.

8.1.1.10 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.

This Threat is satisfied by ensuring that:

- O.SYSTEM_MONITORING: To reduce the potential of security relevant actions occurring without notice, the TOE is required to log security relevant events and transmit those logs to external log server.

8.1.1.11 T.USER_DATA_REUSE

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

This Threat is satisfied by ensuring that:

- O.RESIDUAL_INFORMATION_CLEARING: To reduce the potential of data being erroneously sent to an unintended recipient, the TOE is required to ensure that residual data is appropriately managed.

8.1.1.12 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.

This Assumption is satisfied by ensuring that:

- 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.

8.1.1.13 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.

This Assumption is satisfied by ensuring that:

- 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.

8.1.1.14 A.PHYSICAL

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

This Assumption is satisfied by ensuring that:

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

8.1.1.15 A.TRUSTED_ADMIN

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

This Assumption is satisfied by ensuring that:

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

8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that table below indicates the requirements that effectively satisfy the individual objectives.

8.2.1 Security Functional Requirements Rationale

All SFRs identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective for which it is intended to satisfy. Note that the NDPP and EPFW identify the correspondence between Security Objectives and SFRs, but fail to provide any rationale for the correspondence. As such, correspondence rationale has been devised to complete this ST appropriately.

Table 18: Objective to Requirement Correspondence

	O.ADDRESS_FILTERING	O.DISPLAY_BANNER*	O.PORT_FILTERING	O.PROTECTED_COMMUNICATIONS*	O.RELATED_CONNECTION_FILTERING	O.RESIDUAL_INFORMATION_CLEARING*	O.SESSION_LOCK*	O.STATEFUL_INSPECTION	O.SYSTEM_MONITORING**	O.TOE_ADMINISTRATION**	O.TSF_SELF_TEST*	O.VERIFIABLE_UPDATES*
FAU_GEN.1*									X			
FAU_GEN.2*									X			
FAU_STG_EXT.1*									X			
FCS_CKM.1*				X								
FCS_CKM_EXT.4*				X								
FCS_COP.1(1)*				X								
FCS_COP.1(2)*				X								X
FCS_COP.1(3)*				X								X
FCS_COP.1(4)*				X								
FCS_HTTPS_EXT.1*				X								
FCS_RBG_EXT.1*				X								
FCS_SSH_EXT.1*				X								
FCS_TLS_EXT.1*				X								
FDP_RIP.2*						X						
FFW_RUL_EXT.1	X		X		X			X	X			

	O.ADDRESS_FILTERING	O.DISPLAY_BANNER*	O.PORT_FILTERING	O.PROTECTED_COMMUNICATIONS*	O.RELATED_CONNECTION_FILTERING	O.RESIDUAL_INFORMATION_CLEARING*	O.SESSION_LOCK*	O.STATEFUL_INSPECTION	O.SYSTEM_MONITORING**	O.TOE_ADMINISTRATION**	O.TSF_SELF_TEST*	O.VERIFIABLE_UPDATES*
FIA_PMG_EXT.1*										X		
FIA_UIA_EXT.1*										X		
FIA_UAU_EXT.2*										X[14]		
FIA_UAU.7*										X		
FMT_MTD.1*										X		
FMT_SMF.1*										X		
FMT_SMR.2*										X		
FPT_ITT.1*				X								
FPT_SKP_EXT.1*				X								
FPT_APW_EXT.1*										X		
FPT_STM.1*									X			
FPT_TST_EXT.1*											X	
FPT_TUD_EXT.1*												X
FTA_SSL_EXT.1*							X			X		
FTA_SSL.3*							X			X		
FTA_SSL.4*							X			X		
FTA_TAB.1*		X										
FTP_ITC.1*				X								
FTP_TRP.1*				X								

*- NDPP

** - Originated in NDPP

[14] This SFR was not mapped in the NDPP. It should be mapped to TOE administration. The same is true for FTA_SSL.4.

8.2.1.1 O.ADDRESS_FILTERING

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

This TOE Security Objective is satisfied by ensuring that:

- FFW_RUL_EXT.1: The TOE is required provide rules to filter and log network traffic based on attributes such as source and destination addresses.

8.2.1.2 O.DISPLAY_BANNER

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

This TOE Security Objective is satisfied by ensuring that:

- FTA_TAB.1: The TOE is required to display the configured advisory banner whenever a user/administrator connects to the TOE.

8.2.1.3 O.PORT_FILTERING

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

This TOE Security Objective is satisfied by ensuring that:

- FFW_RUL_EXT.1: The TOE is required provide rules to filter and log network traffic based on attributes such as source and destination ports.

8.2.1.4 O.PROTECTED_COMMUNICATIONS

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

This TOE Security Objective is satisfied by ensuring that:

- FCS_CKM.1: The TOE is required to be able to generate strong cryptographic keys to support other cryptographic operations.
- FCS_CKM_EXT.4: The TOE is required to zeroize cryptographic keys and sensitive data when no longer need to prevent subsequent disclosure.
- FCS_COP.1(1): The TOE is required to implement FIPS Approved AES in support of cryptographic protocols.
- FCS_COP.1(2): The TOE is required to implement FIPS Approved rDSA (i.e., RSA) in support of cryptographic protocols.
- FCS_COP.1(3): The TOE is required to implement FIPS Approved SHA-1, SHA-224, SHA-256, SHA-384, and/or SHA-512 in support of cryptographic protocols.
- FCS_COP.1(4): The TOE is required to implement FIPS Approved HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and/or HMAC-SHA-512 in support of cryptographic protocols.

- FCS_HTTPS_EXT.1: The TOE is required to implement HTTPS properly to protect applicable network communication channels.
- FCS_RBG_EXT.1: The TOE is required to implement FIPS Approved Random Bit Generation in support of cryptographic protocols.
- FCS_SSH_EXT.1: The TOE is required to implement SSH properly to protect applicable network communication channels.
- FCS_TLS_EXT.1: The TOE is required to implement TLS properly to protect applicable network communication channels.
- FPT_ITT.1: The TOE is required to protect communication between its distributed parts from disclosure and modification.
- FPT_SKP_EXT.1: The TOE is required to prevent even administrators from readily accessing sensitive user and TSF data such as cryptographic keys.
- FTP_ITC.1: The TOE is required to protect communication between itself and its external peers from disclosure and modification.
- FTP_TRP.1: The TOE is required to protect communication between itself and its administrators from disclosure and modification.

8.2.1.5 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.

This TOE Security Objective is satisfied by ensuring that:

- FFW_RUL_EXT.1: The TOE is required to allow network traffic, without an explicit rule, only in response to a connection permitted by ruleset for specific protocols.

8.2.1.6 O.RESIDUAL_INFORMATION_CLEARING

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

This TOE Security Objective is satisfied by ensuring that:

- FDP_RIP.2: The TOE is required to clear all information when allocating storage resources for subsequent activities.

8.2.1.7 O.SESSION_LOCK

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

This TOE Security Objective is satisfied by ensuring that:

- FTA_SSL_EXT.1: The TOE is required to terminate local sessions after an administrator defined period of inactivity indicating the user may not be in attendance.
- FTA_SSL.3: The TOE is required to terminate remote sessions after an administrator defined period of inactivity indicating the user may not be in attendance.
- FTA_SSL.4: The TOE is required to provide administrators a way to terminate their own sessions when no longer in use.

8.2.1.8 O.STATEFUL_INSPECTION

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

This TOE Security Objective is satisfied by ensuring that:

- FFW_RUL_EXT.1: The TOE is required to allow network packet that belongs to an allowed, establishing connection. If the network packet does not belong to an allowed, establishing connection, then the ruleset is applied.

8.2.1.9 O.SYSTEM_MONITORING

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

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: The TOE is required to be able to generate audit events for security relevant activities on the TOE.
- FAU_GEN.2: The TOE is required to associate audit events to users to ensure proper accountability.
- FAU_STG_EXT.1: The TOE is required to be able to transmit audit events to an external audit server via a secure channel to protect the integrity and confidentiality of those records.
- FFW_RUL_EXT.1: The TOE is required to provide the means for administrators to configure firewall rules to log when network traffic is found to match the configured rule.
- FPT_STM.1: The TOE is required to provide reliable time stamps to be used in its audit records for proper accounting.

8.2.1.10 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.

This TOE Security Objective is satisfied by ensuring that:

- FIA_PMG_EXT.1: The TOE is required to implement mechanisms allowing an administrator to constrain the construction of passwords to encourage more secure (or harder to guess) passwords.
- FIA_UIA_EXT.1: The TOE is required to ensure that users must be identified and authenticated in order to access functions, other than those specifically intended to be accessed without identification and authentication.
- FIA_UAU_EXT.2: The TOE is required to implement a local authentication mechanism for verification.
- FIA_UAU.7: The TOE is required to not echo passwords when being entered to mitigate the chance of an accidental password disclosure.
- FMT_MTD.1: The TOE is required to restrict access to security relevant data to administrators.
- FMT_SMF.1: The TOE is required to provide a minimum set of security functions to ensure the TOE security features can be properly managed.

- FMT_SMR.2: The TOE is required to implement a minimum of a Security Administrator role and can implement additional roles where necessary.
- FPT_APW_EXT.1: The TOE is required to prevent even administrators from readily accessing sensitive user and TSF data such as passwords.
- FTA_SSL_EXT.1: The TOE is required to terminate local sessions after an administrator defined period of inactivity indicating the administrator may not be in attendance.
- FTA_SSL.3: The TOE is required to terminate remote sessions after an administrator defined period of inactivity indicating the administrator may not be in attendance.
- FTA_SSL.4: The TOE is required to provide administrators a way to terminate their own sessions when no longer in use.

8.2.1.11 O.TSF_SELF_TEST

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

This TOE Security Objective is satisfied by ensuring that:

- FPT_TST_EXT.1: The TOE is required to perform self-tests during start-up to ensure that the TOE security functions appear to be operating correctly.

8.2.1.12 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 and (optionally) from a trusted source.

This TOE Security Objective is satisfied by ensuring that:

- FCS_COP.1(2): The TOE is required to either use digital signatures or cryptographic hashes to ensure the integrity of updates.
- FCS_COP.1(3): The TOE is required to either use digital signatures or cryptographic hashes to ensure the integrity of updates.
- FPT_TUD_EXT.1: The TOE is required to provide update functions and also the means for an administrator to initiate and verify updates before they are applied.

8.3 Security Assurance Requirements Rationale

The Security Assurance Requirements in this ST represents the SARs identified in the NDPP.

Note that the NDPP and EPFW include a number of ‘Assurance Activities’ which are in effect refinements of the underlying SARs. As such, those assurance activities have been reproduced in this ST since they need be addressed in the context of the evaluation.

8.4 Requirement Dependency Rationale

As can be seen in the following table all of the SFR and SAR dependencies are satisfied in this ST.

Table 19: Requirement Dependencies

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1 and FIA_UID.1	FAU_GEN.1 and FIA_UIA_EXT.1
FAU_STG_EXT.1	FAU_GEN.1	FAU_GEN.1
FCS_CKM.1	(FCS_CKM.2 or FCS_COP.1) and FCS_CKM.4	FCS_COP.1 and FCS_CKM_EXT.4
FCS_CKM_EXT.4	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1)	FCS_CKM.1
FCS_COP.1(1)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM_EXT.4
FCS_COP.1(2)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM_EXT.4
FCS_COP.1(3)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM_EXT.4
FCS_COP.1(4)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1 and FCS_CKM_EXT.4
FCS_HTTPS_EXT.1	FCS_TLS_EXT.1	FCS_TLS_EXT.1
FCS_RBG_EXT.1	none	none
FCS_SSH_EXT.1	FCS_COP.1	FCS_COP.1(*)
FCS_TLS_EXT.1	FCS_COP.1	FCS_COP.1(*)
FDP_RIP.2	none	none
FFW_RUL_EXT.1	none	none
FIA_PMG_EXT.1	none	none
FIA_UIA_EXT.1	none	none
FIA_UAU_EXT.2	none	none
FIA_UAU.7	FIA_UAU.1	FIA_UIA_EXT.1
FMT_MTD.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.2 and FMT_SMF.1
FMT_SMF.1	none	none
FMT_SMR.2	FIA_UID.1	FIA_UIA_EXT.1
FPT_ITT.1	none	none

ST Requirement	CC Dependencies	ST Dependencies
FPT_SKP_EXT.1	none	none
FPT_APW_EXT.1	none	none
FPT_STM.1	none	none
FPT_TST_EXT.1	none	none
FPT_TUD_EXT.1	none	none
FTA_SSL_EXT.1	none	none
FTA_SSL.3	none	none
FTA_SSL.4	none	none
FTA_TAB.1	none	none
FTP_ITC.1	none	none
FTP_TRP.1	none	none
ADV_FSP.1	ADV_FSP.1	ADV_FSP.1
AGD_OPE.1	ADV_FSP.1	ADV_FSP.1
AGD_PRE.1	none	none
ALC_CMC.1	ALC_CMS.1	ALC_CMS.1
ALC_CMS.1	none	none
ATE_IND.1	ADV_FSP.1 and AGD_OPE.1 and AGD_PRE.1	ADV_FSP.1 and AGD_OPE.1 and AGD_PRE.1
AVA_VAN.1	ADV_FSP.1 and AGD_OPE.1 and AGD_PRE.1	ADV_FSP.1 and AGD_OPE.1 and AGD_PRE.1

8.5 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. The table below demonstrates the relationship between security requirements and security functions.

Table 20: Security Function vs. Requirement Mapping

	Security Audit*	Cryptographic Support*	User Data Protection*	Stateful Traffic Filtering	Identification and Authentication *	Security Management*	Protection of the TSF*	TOE Access*	Trusted Path/Channels*
FAU_GEN.1*	X								
FAU_GEN.2*	X								
FAU_STG_EXT.1*	X								
FCS_CKM.1*		X							
FCS_CKM_EXT.4*		X							
FCS_COP.1(1)*		X							
FCS_COP.1(2)*		X							
FCS_COP.1(3)*		X							
FCS_COP.1(4)*		X							
FCS_HTTPS_EXT.1*		X							
FCS_RBG_EXT.1*		X							
FCS_SSH_EXT.1*		X							
FCS_TLS_EXT.1*		X							
FDP_RIP.2*			X						
FFW_RUL_EXT.1				X					
FIA_PMG_EXT.1*					X				
FIA_UID_EXT.1*					X				
FIA_UAU_EXT.2*					X				

FIA_UAU.7*					X				
FMT_MTD.1*						X			
FMT_SMF.1*						X			
FMT_SMR.2*						X			
FPT_ITT.1*							X		
FPT_SKP_EXT.1*							X		
FPT_AWP_EXT.1*							X		
FPT_STM.1*							X		
FPT_TST_EXT.1*							X		
FPT_TUD_EXT.1*							X		
FTA_SSL_EXT.1*								X	
FTA_SSL.3*								X	
FTA_SSL.4*								X	
FTA_TAB.1*								X	
FTP_ITC.1*									X
FTP_TRP.1*									X

*- NDPP