BAE Systems STOP™ 8.8.2

Security Target

Doc No: 2189-001-D102
Version: 0.24
15 September 2023

BAE Systems Information and Electronics Systems Integration, Inc.
11487 Sunset Hills Road
Reston, Virginia, USA
20190

Prepared by:
EWA-Canada, An Intertek Company
1223 Michael Street North, Suite 200
Ottawa, Ontario, Canada
K1J 7T2

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 SECURITY TARGET INTRODUCTION

This Security Target (ST) defines the scope of the evaluation in terms of the assumptions made, the intended environment for the Target of Evaluation (TOE), the Information Technology (IT) security functional and assurance requirements to be met, and the level of confidence (evaluation assurance level) to which it is asserted that the TOE satisfies its IT security requirements. This document forms the baseline for the Common Criteria (CC) evaluation.

1.1 DOCUMENT ORGANIZATION

Section 1, ST Introduction, provides the Security Target reference, the Target of Evaluation reference, the TOE overview and the TOE description.

Section 2, Conformance Claims, describes how the ST conforms to the Common Criteria and Packages. The ST does not conform to a Protection Profile.

Section 3, Security Problem Definition, describes the expected environment in which the TOE is to be used. This section defines the set of threats that are relevant to the secure operation of the TOE, organizational security policies with which the TOE must comply, and secure usage assumptions applicable to this analysis.

Section 4, Security Objectives, defines the set of security objectives to be satisfied by the TOE and by the TOE operating environment in response to the problem defined by the security problem definition.

Section 5, Extended Components Definition, defines the extended components which are then detailed in Section 6.

Section 6, Security Requirements, specifies the security functional and assurance requirements that must be satisfied by the TOE and the IT environment.

Section 7, TOE Summary Specification, describes the security functions that are included in the TOE to enable it to meet the IT security functional requirements.

Section 8, Acronyms, defines the acronyms used in this ST.

Appendix A, CAVP Certificates and Usage, identifies the cryptographic operations and Cryptographic Algorithm Validation Program (CAVP) certificate numbers.

1.2 SECURITY TARGET REFERENCE

ST Title:	BAE Systems STOP™ 8.8.2 Security Target
ST Version:	0.24
ST Date:	15 September 2023

1.3 TOE REFERENCE

TOE Identification:	BAE Systems STOP™ 8.8.2-11895
TOE Developer:	BAE Systems
TOE Type:	Operating System

1.4 TOE OVERVIEW

The TOE is the Secure Trusted Operating System (STOP) which is a general purpose, multi-user Operating System (OS) that includes common Linux commands and tools. It provides a secure platform that supports user permissions, access controls, auditing, and cryptographic functionality.

The TOE is a software only TOE.

1.4.1 TOE Environment

STOP is hosted on a hardware platform. This may be a hardware model supplied by BAE Systems as part of the XTS® product line, or any other hardware that meets the minimum system requirements:

- x86 CPU(s) supporting the x86_64 instruction set and features (Intel™ Core™, Intel Xeon or AMD™ x86_64)
- At least 1 GB of system memory.

In the evaluated configuration, the TOE is deployed on the XTS 752 hardware with an Intel Xeon Gold 6256 Central Processing Unit (CPU).

The following components are required for operation of the TOE in the evaluated configuration.

Component	Description
TOE Hardware Platform	XTS 752 with an Intel Xeon Gold 6256 CPU.
Administrative Workstation	General purpose hardware platform for administration of the TOE.
Update Server	General purpose hardware platform used by the TOE for checking and downloading OS and application updates.

Table 1 – Non-TOE Hardware and Software

1.5 TOE DESCRIPTION

1.5.1 Physical Scope

In the evaluated configuration, the TOE is the STOP 8.8.2 OS deployed on the XTS 752 hardware. As per the Protection Profile for General Purpose Operating Systems, Version 4.2.1 [PP_OS_V4.2.1], the TOE boundary encompasses the OS kernel and its drivers, shared software libraries, and some application software included with the OS.

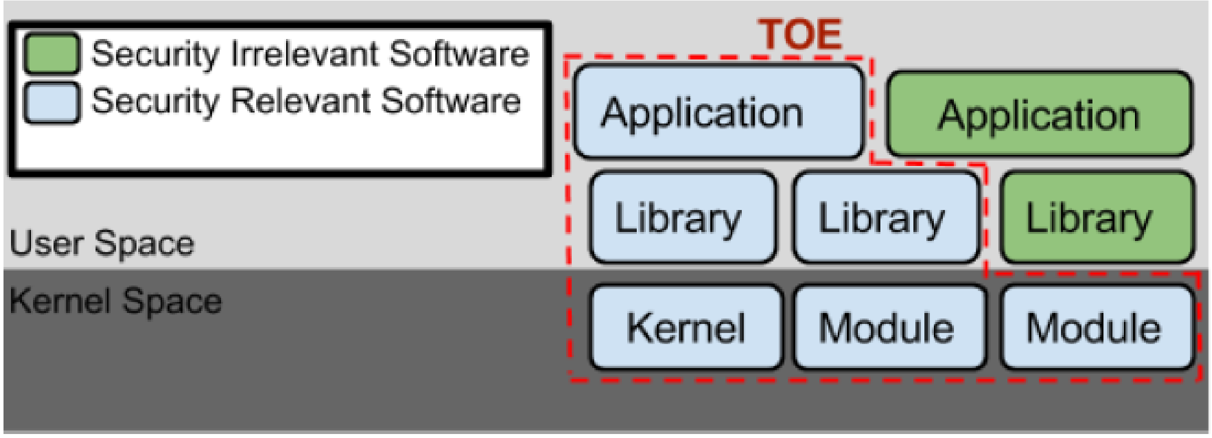

Figure 1 – TOE Boundary

1.5.1.1 TOE Delivery

The TOE is shipped as physical media using a trusted courier with tracked shipping. All TOE components, including the OS and guidance documentation, are delivered on read-only media (DVD[1]) with the name of the release (STOP 8.8.2 Release). The OS is provided as an ISO[2] image and the guidance provided in PDF[3] format. The TOE includes the following components:

STOP OS Software

- STOP 8.8.2 Operating System
  - *STOP_8.8.2_boot.iso*

Guidance Documentation

- BAE Systems STOP 8.08.02 User's Manual, Version XTDOC0159-48, February 13, 2023

The following Common Criteria Guidance Supplement is also available to customers, in PDF format, upon request:

- BAE Systems STOP™ 8.8.2 Common Criteria Guidance Supplement, Version 0.12, 15 September 2023

[1] Digital Video Disc
[2] International Standards Organization
[3] Portable Document Format

1.5.2 Logical Scope

The logical boundary of the TOE includes all interfaces and functions within the physical boundary. The logical boundary of the TOE may be broken down by the security function classes described in Section 6. Table 2 summarizes the logical scope of the TOE.

Functional Classes	Description
Audit Data Generation (FAU)	Audit entries are generated for security related events, and identify the user associated with the event.
Cryptographic Support (FCS)	Cryptographic functionality is provided to protect communication links between the TOE and IT entities. Refer to Table 13 for a list of associated Cryptographic Algorithm Validation Program (CAVP) certificate #s and usage.
User Data Protection (FDP)	The TOE provides access control through Multi-Level Security (MLS), Role-Based Access Control (RBAC), and Discretionary Access Control (DAC) mechanisms.
Identification and Authentication (FIA)	Users must identify and authenticate prior to gaining access to the TOE. Users are locked out after a defined number of unsuccessful authentication attempts. X509 certificates are used for trusted updates and executable code integrity verification, and TLS server authentication.
Security Management (FMT)	The TOE provides management capabilities via a command line interface, accessed locally. Management functions allow the administrators to configure access control settings, configure network settings, configure user settings, revoke user and object security attributes, and configure auditing options and review logs.
Protection of the TSF (FPT)	The TOE implements the following protection mechanisms: • Address Space Layout Randomization for user space code • Stack buffer overflow protection • Bootchain integrity through the OS and kernel • Trusted updates to the OS and application software.
Trusted Path/Channels (FTP)	The TOE uses Transport Layer Security (TLS) for communication with external update repositories.

Table 2 – Logical Scope of the TOE

1.5.3 Excluded Functionality

SSH is disabled in the evaluated configuration.

2 CONFORMANCE CLAIMS

2.1 COMMON CRITERIA CONFORMANCE CLAIM

This Security Target claims to be conformant to Version 3.1 of Common Criteria for Information Technology Security Evaluation according to:

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2017-04-001, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2017-04-002, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components CCMB-2017-04-003, Version 3.1, Revision 5, April 2017

As follows:

- CC Part 2 extended
- CC Part 3 extended

The Common Methodology for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017 has been taken into account.

2.2 PROTECTION PROFILE CONFORMANCE CLAIM

2.2.1 Protection Profiles

This ST claims exact conformance with the National Information Assurance Partnership (NIAP) Protection Profile for General Purpose Operating Systems, Version 4.2.1 [PP_OS_V4.2.1] and the Technical Decisions in Table 3.

Technical Decision (TD)	Applicable	Exclusion Rationale
TD0715 - Updates to FIA_X509_EXT.1 for Exception Processing and Test Conditions	Yes	
TD0680 - OS 4.2.1 Conformance Claims section updated to allow for MOD_WLAN_CLI_v1.0	Yes	
TD0649 – Conformance claims for OS PP v4.2.1	Yes	

Technical Decision (TD)	Applicable	Exclusion Rationale
TD0630 – FCS_COP.1 Requirements for Secure Shell	Yes	
TD0600 – VPN package approved	No	MOD_VPNC_V2.3 is not claimed.
TD0578 - SHA-1 is no longer mandatory	Yes	
TD0501 – Cryptographic selections and updates for OS PP	Yes	
TD0493 – X.509v3 certificates when using digital signatures for Boot Integrity	Yes	
TD0463 – Clarification for FPT_TUD_EXT	Yes	
TD0441 – Updated TLS Ciphersuites for OS PP	Yes	
TD0386 – Platform-Provided Verification of Updates	Yes	
TD0365 – FCS_CKM_EXT.4 selections	Yes	

Table 3 – PP_OS_V4.2.1 Technical Decisions

2.3 CONFORMANCE RATIONALE

The TOE is inherently consistent with the Compliant Targets of Evaluation described in the [PP_OS_V4.2.1]. The security problem definition, statement of security objectives and statement of security requirements in this ST conform exactly to the security problem definition, statement of security objectives and statement of security requirements contained in the [PP_OS_V4.2.1].

3 SECURITY PROBLEM DEFINITION

3.1 THREATS

Table 4 lists the threats addressed by the TOE. Mitigation of the threats is achieved through the instantiation of the objectives identified in Section 4.1, Security Objectives for the TOE.

Threat	Description
T.NETWORK_ATTACK	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with applications and services running on or part of the OS with the intent of compromise. Engagement may consist of altering existing legitimate communications.
T.NETWORK_EAVESDROP	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between applications and services that are running on or part of the OS.
T.LOCAL_ATTACK	An attacker may compromise applications running on the OS. The compromised application may provide maliciously formatted input to the OS through a variety of channels including unprivileged system calls and messaging via the file system.
T.LIMITED_PHYSICAL_ACCESS	An attacker may attempt to access data on the OS while having a limited amount of time with the physical device.

Table 4 – Threats

3.2 ORGANIZATIONAL SECURITY POLICIES

There are no Organizational Security Policies applicable to this TOE.

3.3 ASSUMPTIONS

The assumptions required to ensure the security of the TOE are listed in Table 5.

Assumption	Description
A.PLATFORM	The OS relies upon a trustworthy computing platform for its execution. This underlying platform is out of scope of this PP.
A.PROPER_USER	The user of the OS is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy. At the same time, malicious software could act as the user, so requirements which confine malicious subjects are still in scope.
A.PROPER_ADMIN	The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.

Table 5 – Assumptions

4 SECURITY OBJECTIVES

The purpose of the security objectives is to address the security concerns and to show which security concerns are addressed by the TOE, and which are addressed by the environment. Threats may be addressed by the TOE or the security environment or both. Therefore, the CC identifies two categories of security objectives:

- Security objectives for the TOE
- Security objectives for the environment

4.1 SECURITY OBJECTIVES FOR THE TOE

This section identifies and describes the security objectives that are to be addressed by the TOE.

Security Objective	Description
O.ACCOUNTABILITY	Conformant OSes ensure that information exists that allows administrators to discover unintentional issues with the configuration and operation of the operating system and discover its cause. Gathering event information and immediately transmitting it to another system can also enable incident response in the event of system compromise.
O.INTEGRITY	Conformant OSes ensure the integrity of their update packages. OSes are seldom if ever shipped without errors, and the ability to deploy patches and updates with integrity is critical to enterprise network security. Conformant OSes provide execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems.
O.MANAGEMENT	To facilitate management by users and the enterprise, conformant OSes provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration and application execution control.
O.PROTECTED_STORAGE	To address the issue of loss of confidentiality of credentials in the event of loss of physical control of the storage medium, conformant OSes provide data-at-rest protection for credentials. Conformant OSes also provide access controls which allow users to keep their files private from other users of the same system.

Security Objective	Description
O.PROTECTED_COMMS	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant OSes provide mechanisms to create trusted channels for CSP and sensitive data. Both CSP and sensitive data should not be exposed outside of the platform.

Table 6 – Security Objectives for the TOE

4.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

This section identifies and describes the security objectives that are to be addressed by the IT environment or by non-technical or procedural means.

Security Objective	Description
OE.PLATFORM	The OS relies on being installed on trusted hardware.
OE.PROPER_USER	The user of the OS is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy. Standard user accounts are provisioned in accordance with the least privilege model. Users requiring higher levels of access should have a separate account dedicated for that use.
OE.PROPER_ADMIN	The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.

Table 7 – Security Objectives for the Operational Environment

4.3 SECURITY OBJECTIVES RATIONALE

The following table maps the security objectives to the assumptions and threats identified for the TOE.

	T.NETWORK_ATTACK	T.NETWORK_EAVESDROP	T.LOCAL_ATTACK	T.LIMITED_PHYSICAL_ACCESS	A.PLATFORM	A.PROPER_USER	A.PROPER_ADMIN
O.ACCOUNTABILITY	X		X				
O.INTEGRITY	X		X				
O.MANAGEMENT	X	X					
O.PROTECTED_STORAGE				X			
O.PROTECTED_COMMS	X	X					
OE.PLATFORM					X		
OE.PROPER_USER						X	
OE.PROPER_ADMIN							X

Table 8 – Mapping Between Objectives, Threats, OSPs, and Assumptions

4.3.1 Security Objectives Rationale Related to Threats

The security objectives rationale related to threats traces the security objectives for the TOE and the Operational Environment back to the threats addressed by the TOE.

Threat: **T.NETWORK_ATTACK**	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with applications and services running on or part of the OS with the intent of compromise. Engagement may consist of altering existing legitimate communications.	
Objectives:	O.ACCOUNTABILITY	Conformant OSes ensure that information exists that allows administrators to discover unintentional issues with the configuration and operation of the operating system and discover its cause. Gathering event information and immediately transmitting it to another system can also enable incident response in the event of system compromise.
	O.INTEGRITY	Conformant OSes ensure the integrity of their update packages. OSes are seldom if ever shipped without errors, and the ability to deploy patches and updates with integrity is critical to enterprise network security. Conformant OSes provide execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems.
	O.MANAGEMENT	To facilitate management by users and the enterprise, conformant OSes provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration and application execution control.
	O.PROTECTED_COMMS	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant OSes provide mechanisms to create trusted channels for CSP and sensitive data. Both CSP and sensitive data should not be exposed

		outside of the platform.
Rationale:	The threat T.NETWORK_ATTACK is countered by O.PROTECTED_COMMS as this provides for integrity of transmitted data. The threat T.NETWORK_ATTACK is countered by O.INTEGRITY as this provides for integrity of software that is installed onto the system from the network. The threat T.NETWORK_ATTACK is countered by O.MANAGEMENT as this provides for the ability to configure the OS to defend against network attack. The threat T.NETWORK_ATTACK is countered by O.ACCOUNTABILITY as this provides a mechanism for the OS to report behavior that may indicate a network attack has occurred.	

Threat: **T.NETWORK_ EAVESDROP**	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between applications and services that are running on or part of the OS.	
Objectives:	O.MANAGEMENT	To facilitate management by users and the enterprise, conformant OSes provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration and application execution control.
	O.PROTECTED_COMMS	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant OSes provide mechanisms to create trusted channels for CSP and sensitive data. Both CSP and sensitive data should not be exposed outside of the platform.
Rationale:	The threat T.NETWORK_EAVESDROP is countered by O.PROTECTED_COMMS as this provides for confidentiality of transmitted data. The threat T.NETWORK_EAVESDROP is countered by O.MANAGEMENT as this provides for the ability to configure the OS to protect the confidentiality of its transmitted data.	

Threat: **T.LOCAL_ATTACK**	An attacker may compromise applications running on the OS. The compromised application may provide maliciously formatted input to the OS through a variety of channels including unprivileged system calls and messaging via the file system.	
Objectives:	O.ACCOUNTABILITY	Conformant OSes ensure that information exists that allows administrators to discover unintentional issues with the configuration and operation of the operating system and discover its cause. Gathering event information and immediately transmitting it to another system can also enable incident response in the event of system compromise.
	O.INTEGRITY	Conformant OSes ensure the integrity of their update packages. OSes are seldom if ever shipped without errors, and the ability to deploy patches and updates with integrity is critical to enterprise network security. Conformant OSes provide execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems.
Rationale:	The objective O.INTEGRITY protects against the use of mechanisms that weaken the TOE with regard to attack by other software on the platform. The objective O.ACCOUNTABILITY protects against local attacks by providing a mechanism to report behavior that may indicate a local attack is occurring or has occurred.	

Threat: **T.LIMITED_PHYSICAL_ACCESS**	An attacker may attempt to access data on the OS while having a limited amount of time with the physical device.	
Objectives:	O.PROTECTED_STORAGE	To address the issue of loss of confidentiality of credentials in the event of loss of physical control of the storage medium, conformant OSes provide data-at-rest protection for credentials. Conformant OSes also provide access controls which allow users to keep their files private from other users of the same system.
Rationale:	The objective O.PROTECTED_STORAGE protects against unauthorized attempts to access physical storage used by the TOE.	

4.3.2 Security Objectives Rationale Related to Assumptions

The security objectives rationale related to assumptions traces the security objectives for the operational environment back to the assumptions for the TOE’s operational environment.

Assumption: **A.PLATFORM**	The OS relies upon a trustworthy computing platform for its execution. This underlying platform is out of scope of this PP.	
Objectives:	OE.PLATFORM	The OS relies on being installed on trusted hardware.
Rationale:	The operational environment objective OE.PLATFORM is realized through A.PLATFORM.	

Assumption: **A.PROPER_USER**	The user of the OS is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy. At the same time, malicious software could act as the user, so requirements which confine malicious subjects are still in scope.	
Objectives:	OE.PROPER_USER	The user of the OS is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy. Standard user accounts are provisioned in accordance with the least privilege model. Users requiring higher levels of access should have a separate account dedicated for that use.
Rationale:	The operational environment objective OE.PROPER_USER is realized through A.PROPER_USER.	

Assumption: **A.PROPER_ADMIN**	The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.	
Objectives:	OE.PROPER_ADMIN	The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.
Rationale:	The operational environment objective OE.PROPER_ADMIN is realized through A.PROPER_ADMIN.	

5 EXTENDED COMPONENTS DEFINITION

5.1 SECURITY FUNCTIONAL REQUIREMENTS

This section provides a definition for all the extended components described within the [PP_OS_V4.2.1] claimed within this ST.

5.1.1 Cryptographic Support (FCS)

5.1.1.1 FCS_CKM_EXT.4 Cryptographic Key Destruction

FCS_CKM_EXT.4.1 The OS shall destroy cryptographic keys and key material in accordance with a specified cryptographic key destruction method [**selection**:

- *For volatile memory, the destruction shall be executed by a [**selection**:*
  - *single overwrite consisting of [**selection**: a pseudo-random pattern using the TSF's RBG, zeroes, ones, a new value of a key, [**assignment**: any value that does not contain any CSP]],*
  - *removal of power to the memory,*
  - *destruction of reference to the key directly followed by a request for garbage collection*

  ],
- *For non-volatile memory that consists of [**selection**:*
  - ***destruction of all key encrypting keys protecting the target key according to FCS_CKM_EXT.4.1, where none of the KEKs protecting the target key are derived***
  - *the invocation of an interface provided by the underlying platform that [**selection**:*
    - *logically addresses the storage location of the key and performs a [**selection**: single, [**assignment**: ST author defined multi-pass]] overwrite consisting of [**selection**: zeroes, ones, pseudo-random pattern, a new value of a key of the same size, [**assignment**: any value that does not contain any CSP]],*
    - *instructs the underlying platform to destroy the abstraction that represents the key]*

  ]

].

5.1.1.2 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1 The OS shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using [**selection**:

- *Hash_DRBG (any),*
- *HMAC_DRBG (any),*
- *CTR_DRBG (AES)*

].

FCS_RBG_EXT.1.2 The deterministic RBG used by the OS shall be seeded by an entropy source that accumulates entropy from a [**selection**:

- *Software-based noise source,*
- *Platform-based noise source*

] with a minimum of [**selection**:

- *128 bits,*
- *256 bits*

] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

5.1.1.3 FCS_STO_EXT.1 Storage of Sensitive Data

FCS_STO_EXT.1.1 The OS shall implement functionality to encrypt sensitive data stored in non-volatile storage and provide interfaces to applications to invoke this functionality.

5.1.1.4 FCS_TLSC_EXT.1.1 TLS Client Protocol

FCS_TLSC_EXT.1.1 The OS shall implement TLS 1.2 (RFC 5246) supporting the following cipher suites: [**selection**:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

].

FCS_TLSC_EXT.1.2 The OS shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3 The OS shall only establish a trusted channel if the peer certificate is valid.

5.1.2 User Data Protection (FDP)

5.1.2.1 FDP_ACF_EXT.1 Access Controls for Protecting User Data

FDP_ACF_EXT.1.1 The OS shall implement access controls which can prohibit unprivileged users from accessing files and directories owned by other users.

5.1.3 Identification and Authentication (FIA)

5.1.3.1 FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1 The OS shall implement functionality to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The OS shall validate a certificate path by ensuring the presence of the *basicConstraints* extension, that the *CA* flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The TSF shall validate that any CA certificate includes caSigning purpose in the key usage field.
- The OS shall validate the revocation status of the certificate using [**selection**: *OCSP as specified in RFC 6960, CRL as specified in RFC 8603, an OCSP TLS Status Request Extension (OCSP stapling) as specified in RFC 6066, OCSP TLS Multi-Certificate Status Request Extension (i.e., OCSP Multi-Stapling) as specified in RFC 6961*] with [**selection**: *no exceptions, [**assignment**: exceptional use cases and alternative status check]*].
- The OS shall validate the *extendedKeyUsage* field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the *extendedKeyUsage* field
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the *extendedKeyUsage* field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
  - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
  - [selection: Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field, no other rules]

FIA_X509_EXT.1.2 The OS shall only treat a certificate as a CA certificate if the *basicConstraints* extension is present and the CA flag is set to TRUE.

5.1.3.2 FIA_X509_EXT.2 Certificate Authentication

FIA_X509_EXT.2.1 The OS shall use X.509v3 certificates as defined by RFC 5280 to support authentication for TLS and [**selection**: *DTLS, HTTPS, [**assignment**: none], no other protocols*] connections.

5.1.4 Security Management (FMT)

5.1.4.1 FMT_MOF_EXT.1 Management of Security Functions Behavior

FMT_MOF_EXT.1.1 The OS shall restrict the ability to perform the function indicated in the "Administrator" column in FMT_SMF_EXT.1.1 to the administrator.

5.1.4.2 FMT_SMF_EXT.1 Specification of Management Functions

FMT_SMF_EXT.1.1 The OS shall be capable of performing the following management functions:

Management Function	Administrator	User
Enable/disable [**selection**: *screen lock, session timeout*]	X	O
Configure [**selection**: *screen lock, session*] inactivity timeout	X	O
Configure local audit storage capacity	O	O
Configure minimum password length	O	O
Configure minimum number of special characters in password	O	O
Configure minimum number of numeric characters in password	O	O
Configure minimum number of uppercase characters in password	O	O
Configure minimum number of lowercase characters in password	O	O
Configure lockout policy for unsuccessful authentication attempts through [**selection**: *timeouts between attempts, limiting number of attempts during a time period*]	O	O
Configure host-based firewall	O	O
Configure name/address of directory server with which to bind	O	O
Configure name/address of audit/logging server to which to send audit/logging records	O	O
Configure audit rules	O	O
Configure name/address of network time server	O	O
Enable/disable automatic software update	O	O

Management Function	**Administrator**	**User**
Configure WiFi interface	O	O
Enable/disable Bluetooth interface	O	O
Enable/disable [**assignment**: *list of other external interfaces*]	O	O
[**assignment**: *list of other management functions to be provided by the TSF*]	O	O

.

5.1.5 Protection of the TSF (FPT)

5.1.5.1 FPT_ACF_EXT.1 Access Controls

FPT_ACF_EXT.1.1 The OS shall implement access controls which prohibit unprivileged users from modifying:

- Kernel and its drivers/modules
- Security audit logs
- Shared libraries
- System executables
- System configuration files
- [**assignment**: *other objects*].

FPT_ACF_EXT.1.2 The OS shall implement access controls which prohibit unprivileged users from reading:

- Security audit logs
- System-wide credential repositories
- [**assignment**: list of other objects].

5.1.5.2 FPT_ASLR_EXT.1 Address Space Layout Randomization

FPT_ASLR_EXT.1.1 The OS shall always randomize process address space memory locations with [**selection**: *8, [**assignment**: number greater than 8]*] bits of entropy except for [**assignment**: *list of explicit exceptions*].

5.1.5.3 FPT_SBOP_EXT.1 Stack Buffer Overflow Protection

FPT_SBOP_EXT.1.1 The OS shall [**selection**: *employ stack-based buffer overflow protections, not store parameters/variables in the same data structures as control flow values*].

5.1.5.4 FPT_TST_EXT.1 Boot Integrity

FPT_TST_EXT.1.1 The OS shall verify the integrity of the bootchain up through the OS kernel and [**selection**:

- *all executable code stored in mutable media,*
- *[**assignment**: list of other executable code],*
- *no other executable code*

] prior to its execution through the use of [**selection**:

- *a digital signature using a hardware-protected asymmetric key,*
- *a digital signature using an X509 certificate with hardware-based protection,*
- *a hardware-protected hash*

].

5.1.5.5 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 The OS shall provide the ability to check for updates to the OS software itself and shall use a digital signature scheme specified in FCS_COP.1(3) to validate the authenticity of the response.

FPT_TUD_EXT.1.2 The OS shall cryptographically verify updates to itself using a digital signature prior to installation using schemes specified in FCS_COP.1(3).

5.1.5.6 FPT_TUD_EXT.2 Trusted Update for Application Software

FPT_TUD_EXT.2.1 The OS shall provide the ability to check for updates to application software and shall use a digital signature scheme specified in FCS_COP.1(3) to validate the authenticity of the response.

FPT_TUD_EXT.2.2 The OS shall cryptographically verify the integrity of updates to applications using a digital signature specified by FCS_COP.1(3) prior to installation.

5.1.6 Trusted Path/Channels (FTP)

5.1.6.1 FTP_ITC_EXT.1 Trusted Channel Communication

FTP_ITC_EXT.1.1 The OS shall use [**selection**:

- *TLS as conforming to FCS_TLSC_EXT.1,*
- *DTLS as conforming to FCS_DTLS_EXT.1,*
- *IPsec as conforming to the PP-Module for VPN Clients,*
- *SSH as conforming to the Functional Package for Secure Shell*

] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: [**selection**: *audit server, authentication server, management server, [**assignment**: 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.

5.2 SECURITY ASSURANCE REQUIREMENTS

This section provides a definition for all the extended assurance requirements described within the [PP_OS_V4.2.1] claimed within this ST.

5.2.1 ALC_TSU_EXT.1 Timely Security Updates

This component requires the OS developer, in conjunction with any other necessary parties, to provide information as to how the end-user devices are updated to address security issues in a timely manner. The documentation describes the process of providing updates to the public from the time a security flaw is reported/discovered, to the time an update is released. This description includes the parties involved (e.g., the developer, carriers(s)) and the steps that are performed (e.g., developer testing, carrier testing), including worst case time periods, before an update is made available to the public.

Developer action elements:

ALC_TSU_EXT.1.1D The developer shall provide a description in the TSS of how timely security updates are made to the OS.

ALC_TSU_EXT.1.2D The developer shall provide a description in the TSS of how users are notified when updates change security properties or the configuration of the product.

Content and presentation elements:

ALC_TSU_EXT.1.1C The description shall include the process for creating and deploying security updates for the OS software.

ALC_TSU_EXT.1.2C The description shall include the mechanisms publicly available for reporting security issues pertaining to the OS.

Note: The reporting mechanism could include web sites, email addresses, as well as a means to protect the sensitive nature of the report (e.g., public keys that could be used to encrypt the details of a proof-of-concept exploit).

Evaluator action elements:

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

6 SECURITY REQUIREMENTS

Section 6 provides security functional and assurance requirements that must be satisfied by a compliant TOE. These requirements consist of CC Part 2 functional components and extended requirements as they appear in [PP_OS_V4.2.1].

6.1 CONVENTIONS

The CC permits four types of operations to be performed on functional requirements: selection, assignment, refinement, and iteration. These operations, when performed on requirements that derive from CC Part 2, are identified in this ST in the same manner used in the claimed protection profile. Every attempt has been made to present the Security Functional Requirements (SFRs) exactly as shown and without correction. As a result, not all operations of the same type are shown using the same conventions.

6.2 SECURITY FUNCTIONAL REQUIREMENTS

The security functional requirements for this ST consist of the following components from Part 2 of the CC and extended components defined in Section 5.

Class	Identifier	Name
Audit Data Generation (FAU)	FAU_GEN.1	Audit Data Generation (Refined)
Cryptographic Support (FCS)	FCS_CKM.1	Cryptographic Key Generation (Refined)
	FCS_CKM.2	Cryptographic Key Establishment (Refined)
	FCS_CKM_EXT.4	Cryptographic key Destruction
	FCS_COP.1(1)	Cryptographic Operation – Encryption/Decryption (Refined)
	FCS_COP.1(2)	Cryptographic Operation – Hashing (Refined)
	FCS_COP.1(3)	Cryptographic Operation – Signing (Refined)
	FCS_COP.1(4)	Cryptographic Operation – Keyed-Hash Message Authentication (Refined)
	FCS_RBG_EXT.1	Random Bit Generation
	FCS_STO_EXT.1	Storage of Sensitive Data

Class	Identifier	Name
	FCS_TLSC_EXT.1	TLS Client Protocol
User Data Protection (FDP)	FDP_ACF_EXT.1	Access Controls for Protecting User Data
Identification and Authentication (FIA)	FIA_AFL.1	Authentication Failure Handling (Refined)
	FIA_UAU.5	Multiple Authentication Mechanisms (Refined)
	FIA_X509_EXT.1	X.509 Certificate Validation
	FIA_X509_EXT.2	X.509 Certificate Authentication
Security Management (FMT)	FMT_MOF_EXT.1	Management of Security Functions Behavior
	FMT_SMF_EXT.1	Specification of Management Functions
Protection of the TSF (FPT)	FPT_ACF_EXT.1	Access Controls
	FPT_ASLR_EXT.1	Address Space Layout Randomization
	FPT_SBOP_EXT.1	Stack Buffer Overflow Protection
	FPT_TST_EXT.1	Boot Integrity
	FPT_TUD_EXT.1	Trusted Update
	FPT_TUD_EXT.2	Trusted Update for Application Software
Trusted Path/Channels (FTP)	FTP_ITC_EXT.1	Trusted Channel Communication
	FTP_TRP.1	Trusted Path

Table 9 – Summary of Security Functional Requirements

6.2.1 Audit Data Generation (FAU)

6.2.1.1 FAU_GEN.1 Audit Data Generation (Refined)

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

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

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

c) *[*

- *Authentication events (Success/Failure);*
- *Use of privileged/special rights events (Successful and unsuccessful security, audit, and configuration changes);*
- *Privilege or role escalation events (Success/Failure);*
- *[no other specifically defined auditable events].*

]

].

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

a) Date and time of the event, type of event, subject identity (if applicable), and 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, [*no other audit relevant information*].

6.2.2 Cryptographic Support (FCS)

6.2.2.1 FCS_CKM.1 Cryptographic Key Generation (Refined)

FCS_CKM.1.1 The **OS** shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm **[**

- ***ECC schemes using "NIST curves" P-256, P-384 and [P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4,***

- ***FFC schemes using Diffie-Hellman group 14 that meet the following: RFC 3526***

- ***FFC Schemes using safe primes that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation*** **for Pair-Wise Key Establishment Schemes]**

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

6.2.2.2 FCS_CKM.2 Cryptographic Key Establishment (Refined)

FCS_CKM.2.1 The OS shall **implement functionality to perform cryptographic key establishment** in accordance with a specified cryptographic key **establishment** method:

[

- ***Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography",***
- ***Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography",***
- ***Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526]***

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

6.2.2.3 FCS_CKM_EXT.4 Cryptographic Key Destruction

FCS_CKM_EXT.4.1 The OS shall destroy cryptographic keys and key material in accordance with a specified cryptographic key destruction method [

- *For volatile memory, the destruction shall be executed by a [*
  [
  - *removal of power to the memory*

  ]

  ]

].

FCS_CKM_EXT.4.2 The OS shall destroy all keys and key material when no longer needed.

6.2.2.4 FCS_COP.1(1) Cryptographic Operation – Encryption/ Decryption (Refined)

FCS_COP.1.1(1) The **OS** shall perform [*encryption/decryption services for data*] in accordance with a specified cryptographic algorithm ***[***

- ***AES-CBC (as defined in NIST SP 800-38A)***

] and [

- ***AES-GCM (as defined in NIST SP 800-38D)***

] and cryptographic key sizes [*128-bit, 256-bit*] ~~that meet the following: [assignment: list of standards]~~.

6.2.2.5 FCS_COP.1(2) Cryptographic Operation – Hashing (Refined)

FCS_COP.1.1(2) The **OS** shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm *[*

- *SHA-1,*
- *SHA-256,*
- *SHA-384,*
- *SHA-512*

] **and message digest sizes [**

- ***160 bits,***
- ***256 bits,***
- **384 bits,**
- ***512 bits***

] that meet the following: [*FIPS Pub 180-4*].

6.2.2.6 FCS_COP.1(3) Cryptographic Operation – Signing (Refined)

FCS_COP.1.1(3) The **OS** shall perform [*cryptographic signature services (generation and verification)*] in accordance with a specified cryptographic algorithm **[**

- ***RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 4,***
- ***ECDSA schemes with curves P-256, P-384 and [P-521] that meet FIPS PUB 186-4, "Digital Signature Standard (DSS), Section 5"***

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

6.2.2.7 FCS_COP.1(4) Cryptographic Operation – Keyed-Hash Message Authentication (Refined)

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

6.2.2.8 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1 The OS shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using [*CTR_DRBG (AES)*].

FCS_RBG_EXT.1.2 The deterministic RBG used by the OS shall be seeded by an entropy source that accumulates entropy from a [

- *Software-based noise source,*
- *Platform-based noise source*

] with a minimum of [*256 bits*] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

6.2.2.9 FCS_STO_EXT.1 Storage of Sensitive Data

FCS_STO_EXT.1.1 The OS shall implement functionality to encrypt sensitive data stored in non-volatile storage and provide interfaces to applications to invoke this functionality.

6.2.2.10 FCS_TLSC_EXT.1 TLS Client Protocol

FCS_TLSC_EXT.1.1 The OS shall implement TLS 1.2 (RFC 5246) supporting the following cipher suites: [

- *TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,*
- *TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,*
- *TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,*
- *TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,*
- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*

].

FCS_TLSC_EXT.1.2 The OS shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3 The OS shall only establish a trusted channel if the peer certificate is valid.

6.2.3 User Data Protection (FDP)

6.2.3.1 FDP_ACF_EXT.1 Access Controls for Protecting User Data

FDP_ACF_EXT.1.1 The OS shall implement access controls which can prohibit unprivileged users from accessing files and directories owned by other users.

6.2.4 Identification and Authentication (FIA)

6.2.4.1 FIA_AFL.1 Authentication Failure Handling (Refined)

FIA_AFL.1.1 The **OS** shall detect when [*an administrator configurable positive integer within* [*1 – unlimited*]] unsuccessful authentication attempts occur related to **events with [*authentication based on user name and password*]**.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been **met**, the **OS** shall: **[*Account Lockout*]**.

6.2.4.2 FIA_UAU.5 Multiple Authentication Mechanisms (Refined)

FIA_UAU.5.1 The **OS** shall provide the following authentication mechanisms **[*authentication based on user name and password*]** to support user authentication.

FIA_UAU.5.2 The **OS** shall authenticate any user's claimed identity according to the [

- *Username and password: the TOE verifies a locally stored password hash associated with the provided username for local administration;*

].

6.2.4.3 FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1 The OS shall implement functionality to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation
- The certificate path must terminate with a trusted CA certificate
- The OS shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The TSF shall validate that any CA certificate includes caSigning purpose in the key usage field
- The OS shall validate the revocation status of the certificate using [selection: OCSP as specified in RFC 6960, CRL as specified in RFC 8603, an OCSP TLS Status Request Extension (OCSP stapling) as specified in RFC 6066, OCSP TLS Multi-Certificate Status Request Extension (i.e., OCSP Multi-Stapling) as specified in RFC 6961] with [*no exceptions*].
- The OS shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing Purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.

- S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
- OCSP certificates presented for OCSP responses shall have the OCSP Signing Purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
- [Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field].

FIA_X509_EXT.1.2 The OS shall only treat a certificate as a CA certificate if the *basicConstraints* extension is present and the CA flag is set to TRUE.

6.2.4.4 FIA_X509_EXT.2 Certificate Authentication

FIA_X509_EXT.2.1 The OS shall use X.509v3 certificates as defined by RFC 5280 to support authentication for TLS and [HTTPS, *no other protocols*] connections.

6.2.5 Security Management (FMT)

6.2.5.1 FMT_MOF_EXT.1 Management of Security Functions Behavior

FMT_MOF_EXT.1.1 The OS shall restrict the ability to perform the function indicated in the “Administrator” column in FMT_SMF_EXT.1.1 to the administrator.

6.2.5.2 FMT_SMF_EXT.1 Specification of Management Functions

FMT_SMF_EXT.1.1 The OS shall be capable of performing the following management functions:

Management Function	Administrator	User
Enable/disable [*session timeout*]	X	
Configure [*session*] inactivity timeout	X	
Configure local audit storage capacity	X	
Configure minimum password length	X	
Configure minimum number of special characters in password	X	
Configure minimum number of numeric characters in password	X	
Configure minimum number of uppercase characters in password	X	
Configure minimum number of lowercase characters in password	X	
Configure lockout policy for unsuccessful authentication attempts through [*timeouts between attempts, limiting number of attempts during a time period*]	X	
Configure host-based firewall	X	
Configure name/address of directory server with which to bind		

Management Function	Administrator	User
Configure name/address of audit/logging server to which to send audit/logging records		
Configure audit rules	X	
Configure name/address of network time server	X	
Enable/disable automatic software update		
Configure WiFi interface		
Enable/disable Bluetooth interface		
Enable/disable [*no other external interfaces*]		
[*configure network interfaces*]	X	

6.2.6 Protection of the TSF (FPT)

6.2.6.1 FPT_ACF_EXT.1 Access Controls

FPT_ACF_EXT.1.1 The OS shall implement access controls which prohibit unprivileged users from modifying:

- Kernel and its drivers/modules
- Security audit logs
- Shared libraries
- System executables
- System configuration files
- [*no other objects*].

FPT_ACF_EXT.1.2 The OS shall implement access controls which prohibit unprivileged users from reading:

- Security audit logs
- System-wide credential repositories
- [*no other objects*].

6.2.6.2 FPT_ASLR_EXT.1 Address Space Layout Randomization

FPT_ASLR_EXT.1.1 The OS shall always randomize process address space memory locations with [*[34]*] bits of entropy except for [*any file or object explicitly exempted*].

6.2.6.3 FPT_SBOP_EXT.1 Stack Buffer Overflow Protection

FPT_SBOP_EXT.1.1 The OS shall [*employ stack-based buffer overflow protections*].

6.2.6.4 FPT_TST_EXT.1 Boot Integrity

FPT_TST_EXT.1.1 The OS shall verify the integrity of the bootchain up through the OS kernel and [

- *no other executable code*

] prior to its execution through the use of [

- *a digital signature using a hardware-protected asymmetric key,*

].

6.2.6.5 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 The OS shall provide the ability to check for updates to the OS software itself and shall use a digital signature scheme specified in FCS_COP.1(3) to validate the authenticity of the response.

FPT_TUD_EXT.1.2 The OS shall cryptographically verify updates to itself using a digital signature prior to installation using schemes specified in FCS_COP.1(3).

6.2.6.6 FPT_TUD_EXT.2 Trusted Update for Application Software

FPT_TUD_EXT.2.1 The OS shall provide the ability to check for updates to application software and shall use a digital signature scheme specified in FCS_COP.1(3) to validate the authenticity of the response.

FPT_TUD_EXT.2.2 The OS shall cryptographically verify the integrity of updates to applications using a digital signature specified by FCS_COP.1(3) prior to installation.

6.2.7 Trusted Path/Channels (FTP)

6.2.7.1 FTP_ITC_EXT.1 Trusted Channel Communications

FTP_ITC_EXT.1.1 The OS shall use [

- *TLS as conforming to FCS_TLSC_EXT.1*

] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: [*[OS and application updates]*] 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.

6.2.7.2 FTP_TRP.1 Trusted Path

FTP_TRP.1.1 The **OS** shall provide a communication path between itself and [*local*] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [modification, disclosure].

FTP_TRP.1.2 The **OS** shall permit [*local users*] to initiate communication via the trusted path.

FTP_TRP.1.3 The **OS** shall require use of the trusted path for [[*all remote administrative actions*]].

Note: FTP_TRP.1.3 is not applicable as there are no remote users.

6.3 SECURITY ASSURANCE REQUIREMENTS

The following assurance requirements are applicable to the [PP_OS_V4.2.1] and are addressed by the TOE.

Assurance Class	Assurance Components	
	Identifier	**Name**
Development (ADV)	ADV_FSP.1	Basic functional specification
Guidance Documents (AGD)	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life-Cycle Support (ALC)	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
	ALC_TSU_EXT.1	Timely security updates
Security Target Evaluation (ASE)	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.2	Security objectives
	ASE_REQ.2	Derived security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification
Tests (ATE)	ATE_IND.1	Independent testing – conformance
Vulnerability Assessment (AVA)	AVA_VAN.1	Vulnerability survey

Table 10 – Security Assurance Requirements

6.4 SECURITY REQUIREMENTS RATIONALE

6.4.1 Security Functional Requirements Rationale

The following table provides a mapping between the SFRs and Security Objectives.

	O.ACCOUNTABILITY	O.INTEGRITY	O.MANAGEMENT	O.PROTECTED_STORAGE	O.PROTECTED_COMMS
FAU_GEN.1	X				
FCS_CKM.1					X
FCS_CKM.2					X
FCS_CKM_EXT.4					X
FCS_COP.1(1)				X	X
FCS_COP.1(2)		X			X
FCS_COP.1(3)		X			X
FCS_COP.14)		X			X
FCS_RBG_EXT.1				X	X
FCS_STO_EXT.1				X	
FCS_TLSC_EXT.1					X
FDP_ACF_EXT.1				X	
FIA_AFL.1		X			
FIA_UAU.5		X			
FIA_X509_EXT.1		X			X
FIA_X509_EXT.2					X
FMT_MOF_EXT.1			X		
FMT_SMF_EXT.1			X		

	O.ACCOUNTABILITY	O.INTEGRITY	O.MANAGEMENT	O.PROTECTED_STORAGE	O.PROTECTED_COMMS
FPT_ACF_EXT.1		X			
FPT_ASLR_EXT.1		X			
FPT_SBOP_EXT.1		X			
FPT_TST_EXT.1		X			
FPT_TUD_EXT.1		X			
FPT_TUD_EXT.2		X			
FTP_ITC_EXT.1	X	X			X
FTP_TRP.1			X		

Table 11 – Mapping of SFRs to Security Objectives

6.4.2 SFR Rationale Related to Security Objectives

The following rationale traces each SFR back to the Security Objectives for the TOE.

Objective: **O.ACCOUNTABILITY**	Conformant OSes ensure that information exists that allows administrators to discover unintentional issues with the configuration and operation of the operating system and discover its cause. Gathering event information and immediately transmitting it to another system can also enable incident response in the event of system compromise.	
Security Functional Requirements:	FAU_GEN.1	Audit Data Generation (Refined)
	FTP_ITC_EXT.1	Trusted Channel Communication
Rationale:	FAU_GEN.1 defines the auditable events that must be generated to diagnose the cause of unexpected system behavior. FTP_ITC_EXT.1 provides a mechanism for the TSF to transmit the audit data to a remote system.	

Objective: **O.INTEGRITY**	Conformant OSes ensure the integrity of their update packages. OSes are seldom if ever shipped without errors, and the ability to deploy patches and updates with integrity is critical to enterprise network security. Conformant OSes provide execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems.	
Security Functional Requirements:	FCS_COP.1(2)	Cryptographic Operation – Hashing (Refined)
	FCS_COP.1(3)	Cryptographic Operation – Signing (Refined)
	FCS_COP.1(4)	Cryptographic Operation – Keyed-Hash Message Authentication (Refined)
	FIA_AFL.1	Authentication Failure Handling (Refined)
	FIA_UAU.5	Multiple Authentication Mechanisms (Refined)
	FIA_X509_EXT.1	X.509 Certificate Validation
	FPT_ACF_EXT.1	Access Controls
	FPT_ASLR_EXT.1	Address Space Layout Randomization
	FPT_SBOP_EXT.1	Stack Buffer Overflow Protection
	FPT_TST_EXT.1	Boot Integrity
	FPT_TUD_EXT.1	Trusted Update
	FPT_TUD_EXT.2	Trusted Update for Application Software

	FTP_ITC_EXT.1	Trusted Channel Communication
Rationale:	FPT_SBOP_EXT.1 enforces stack buffer overflow protection that makes it more difficult to exploit running code. FPT_ASLR_EXT.1 prevents attackers from exploiting code that executes in static known memory locations. FPT_TUD_EXT.1 and FPT_TUD_EXT.2 enforce integrity of software updates. FCS_COP.1(2), FCS_COP.1(3), and FCS_COP.1(4) provide the cryptographic mechanisms that are used to verify integrity values. FPT_ACF_EXT.1 guarantees the integrity of critical components by preventing unauthorized modifications of them. FPT_X509_EXT.1 provides X.509 certificates as a way of validating software integrity. FPT_TST_EXT.1 verifies the integrity of stored code. FIA_UAU.5 provides mechanisms that prevent untrusted users from accessing the TSF and FIA_AFL.1 prevents brute-force authentication attempts. FTP_ITC_EXT.1 provides trusted remote communications which makes a remote authenticated session less susceptible to compromise.	

Objective: **O.MANAGEMENT**	To facilitate management by users and the enterprise, conformant OSes provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration and application execution control.	
Security Functional Requirements:	FMT_MOF_EXT.1	Management of Security Functions Behavior
	FMT_SMF_EXT.1	Specification of Management Functions
	FTP_TRP.1	Trusted Path
Rationale:	FMT_SMF_EXT.1 defines the TOE’s management functions and FMT_MOF_EXT.1 defines the privileges required to invoke them. FTP_TRP.1 provides one or more secure interfaces for management of the TSF.	

Objective: **O.PROTECTED_ STORGAE**	To address the issue of loss of confidentiality of credentials in the event of loss of physical control of the storage medium, conformant OSes provide data-at-rest protection for credentials. Conformant OSes also provide access controls which allow users to keep their files private from other users of the same system.	
Security Functional Requirements:	FCS_COP.1(1)	Cryptographic Operation – Encryption/Decryption (Refined)
	FCS_RBG_EXT.1	Random Bit Generation
	FCS_STO_EXT.1	Storage of Sensitive Data
	FDP_ACF_EXT.1	Access Controls for Protecting User Data
Rationale:	FCS_STO_EXT.1 provides a mechanism by which the TOE can designate data as 'sensitive' and subsequently require it to be encrypted. FCS_COP.1(1) defines the symmetric algorithm used to encrypt and decrypt sensitive data. FCS_RBG_EXT.1 defines the random bit generator used to create the symmetric keys used to perform this encryption and decryption. FDP_ACF_EXT.1 enforces logical access control on stored data.	

Objective: **O.PROTECTED_ COMMS**	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant OSes provide mechanisms to create trusted channels for CSP and sensitive data. Both CSP and sensitive data should not be exposed outside of the platform.	
Security Functional Requirements:	FCS_CKM.1	Cryptographic Key Generation (Refined)
	FCS_CKM.2	Cryptographic Key Establishment (Refined)
	FCS_CKM_EXT.4	Cryptographic Key Destruction
	FCS_COP.1(1)	Cryptographic Operation – Encryption/Decryption (Refined)
	FCS_COP.1(2)	Cryptographic Operation – Hashing (Refined)
	FCS_COP.1(3)	Cryptographic Operation – Signing (Refined)
	FCS_COP.1(4)	Cryptographic Operation – Keyed-Hash Message Authentication (Refined)
	FCS_RBG_EXT.1	Random Bit Generation
	FCS_TLSC_EXT.1	TLC Client Protocol

	FIA_X509_EXT.1	X.509 Certificate Validation
	FIA_X509_EXT.2	X.509 Certificate Authentication
	FTP_ITC_EXT.1	Trusted Channel Communication
Rationale:	FCS_TLSC_EXT.1 defines the ability of the TOE to act as a TLS client as a method of enforcing protected communications. FCS_CKM.1, FCS_CKM.2, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), and FCS_RBG_EXT.1 define the cryptographic operations and key lifecycle activity used to support the establishment of protected communications. FIA_X509_EXT.1 defines how the TSF validates x.509 certificates as part of establishing protected communications. FIA_X509_EXT.2 defines the trusted communication protocols for which the TOE must perform certificate validation operations. FTP_ITC_EXT.1 defines the trusted communications channels supported by the TOE.	

6.4.3 Dependency Rationale

The [PP_OS_V4.2.1] contains all the requirements claimed in this Security Target. As such, the dependencies are not applicable since the PP have been approved.

6.4.4 Security Assurance Requirements Rationale

The TOE assurance requirements for this ST are consistent with the Security Assurance Requirements listed in the claimed Protection Profile. These assurance requirements were chosen in order to maintain consistency with the [PP_OS_V4.2.1].

7 TOE SUMMARY SPECIFICATION

This section provides a description of the security functions and assurance measures of the TOE that meet the TOE security requirements.

7.1 SECURITY AUDIT

SFR	Description
FAU_GEN.1	The STOP audit functionality is provided by a kernel-based logging system called System LOGger (slog). Slog assesses whether or not the record should be incorporated into the current slog file based on filter rules. If the record is to be incorporated, slog populates slog records into a plain text file. The possible auditable events include: • Authentication events (Success/Failure) • Use of privileged/special rights events (Successful and unsuccessful security, audit, and configuration changes) • Privilege or role escalation events (Success/Failure) Each audit record includes a date and time of event, type of event, outcome of the event, and the identity of the subject that caused the event. The user associated with the subject that caused the event is identified, where applicable. The outcome of the event is indicated.

7.2 CRYPTOGRAPHIC SUPPORT

SFR	Description
FCS_CKM.1	The TOE implements the following key generation techniques for TLS communications: The TOE implements ECC schemes as specified in FIPS 186-4, Digital Signature Standard (DSS), Appendix B.4 using NIST curves P-256, P-384, and P521 in support of ECDHE key exchange used in TLS communications. The TOE implements FFC Schemes using safe primes that meet NIST Special Publication 800-56A Revision 3 in support of DHE key exchange used in TLS communications. Diffie-Hellman Group 14 (2048-bit) key sizes are supported.
FCS_CKM.2	The TOE supports the following key establishment techniques for TLS communications: Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes

SFR	Description
	Using Discrete Logarithm Cryptography" with P-256, P-384 and P-521 curves. Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" Diffie-Hellman Group 14 (2048-bit) key sizes are supported.
FCS_CKM_EXT.4	For volatile memory, the destruction of keys shall be executed by the removal of power to the memory. The following keys are stored and managed in volatile memory and generated by the DRBG: • TLS (Diffie-Hellman Keys) • TLS Session Keys (AES)
FCS_COP.1(1)	The TOE implements AES data encryption/decryption in CBC and GCM mode as specified in FIPS 197 with 128-bit and 256-bit key sizes.
FCS_COP.1(2)	The TOE implements SHA-1, SHA-256, SHA-384 and SHA-512 as specified in FIPS Pub 180-4. The hashing algorithms are used for TLS signature and HMAC services. For TLS, only SHA-1, SHA-256 and SHA-384 hashes are supported.
FCS_COP.1(3)	The TOE provides cryptographic signature generation and verification services using RSA in accordance with FIPS PUB 186-4, "Digital Signature Standard (DSS)." The TOE supports 2048-bit and 4096-bit keys. The TOE also supports ECDSA with curves P-256, P-384 and P-521 RSA in accordance with FIPS PUB 186-4, "Digital Signature Standard (DSS).
FCS_COP.1(4)	The TOE performs keyed-hash message authentication services in accordance with FIPS Pub 198-1 The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard. The TOE supports SHA-1, SHA-256, SHA-384 and SHA-512 with key sizes of 160 bits, 256 bits, 384 bits and 512 bits.
FCS_RBG_EXT.1	The TOE performs random bit generation using a CTR_DRBG as specified in NIST SP 800-90A.
FCS_STO_EXT.1	The TOE includes the OpenSSL library for encrypting sensitive data. The TOE allows the administrator to encrypt any data using the OpenSSL API, including credentials and keys. OpenSSL provides file encryption services using AES-128 or AES-256 in CBC mode.

SFR	Description
FCS_TLSC_EXT.1	The TOE implements TLSv1.2 supporting the following cipher suites: *TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,* *TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,* *TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,* *TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,* *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,* *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,* *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,* *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,* *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,* *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,* *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,* *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*]. The TOE establishes the reference identifier by parsing the DNS Name or IP address for the configured server. The reference identifier is matched against the SAN if present. If the SAN is not present, the referenced identifier is matched against the CN. The TOE supports wildcards in the DNS name of the server certificate. The TOE does not support URI reference identifiers, SRV reference identifiers, or certificate pinning.

7.3 USER DATA PROTECTION

SFR	Description
FDP_ACF_EXT.1	STOP invokes a Metapolicy to enforce access control. The Metapolicy is devised from the following sub-policies: Bell-LaPadula policy, Biba policy, RBAC policy, and DAC Policy. The Bell-LaPadula and Biba policies are implemented as a single policy and referred to as the BL/B policy. Each user is defined with a clearance. A clearance consists of a minimum label, a default label, and a maximum label. Every subject and object has a security label. A security label consists of three parts, one for each component of the security policy: • Role – a comma-separated list of roles

SFR	Description
	• BL/B – the name of a BL/B label representing the sensitivity level and categories and the integrity level and categories • DAC – the owner, group, and mode in the traditional Unix discretionary access control policy. The fields are separated by periods The Metapolicy queries each sub-policy, assembles a result from each sub-policy, and makes a final decision. The sub-policies are queried in the following order: 1) RBAC 2) BL/B 3) DAC **Role-Based Access Control (RBAC)** Roles consist of named lists of actions. A role is referred to by its name and may only be assigned to a user by an authorized administrator. The RBAC component of a label is a comma-separated list of roles. STOP defines a set of actions that the system can perform, such as open, delete, execute, or mount. These actions are referred to as requests, and they are all-inclusive; nothing happens on the system without one or more of these requests being either directly or indirectly performed. **Bell-LaPadula Biba (BL/B)** STOP combines the concepts of confidentiality and integrity into a single security control called the BL/B. While the Bell-LaPadula and Biba models both dictate read and write permissions, typical uses of the Bell-LaPadula model deal primarily with who is allowed to read data (i.e. confidentiality), and typical uses of the Biba model primarily controls who is allowed to write data (i.e. integrity). Bell-LaPadula and Biba models dictate how data is permitted to traverse domains. The Bell-LaPadula model protects confidentiality by enforcing the Write Up, Read Down (WURD) principle. If an object is at a high confidentiality level, then subjects at a lower level will not be able to read that object. Subjects will, however, be able to write to that object. Common implementations have data classified as Top Secret, Secret, and Unclassified. The Biba model is the mathematical inverse of the Bell-LaPadula model and protects integrity by enforcing the Read Up, Write Down (RUWD) principle. This is similar to the Bell-LaPadula model, but in reverse. An object at a high integrity level can only be written to by subjects at or above that integrity level. Subjects at lower integrity levels can read higher-integrity objects, but not write. Similarly, a subject at a high integrity level is allowed to write to levels at or below its own, but not above. **Discretionary Access Control (DAC)**

SFR	Description
	Unlike the BL/B settings, which a user might not be able to modify, a user will always be able to modify the DAC permissions on files owned by that user, within the constraints of the BL/B policy. The DAC policy labels objects with owner, group, and a set of permission modes (read, write, and execute), for the owner, group, and all other users. STOP implements a simplified version of the traditional UNIX model where only one user and group is associated with a subject or an object, and there are no access control lists. STOP also supports the restricted deletion flag, also known as the sticky bit, on directories. This is used to restrict deletion of objects inside the directory to the owner of that object. This is commonly used for `/tmp` directories where many users require full access, but should not be able to delete other user's files. Unlike the RBAC and BL/B components of a label, the DAC permissions are not named. The permissions are part of the label itself. The DAC component of a label has the following format: <user>.<group>.<mode> The user and group components are simply the names of users. STOP does not have the concept of groups in the traditional Unix sense. There is no user ID or group ID associated with a user. As such, setting the group to a user other than the owner enables the owner to share an object with other users who have their process DAC group label set to that group. Additionally, there is no concept of a root user. On a standard installation, there is no account that has full permissions to everything on the system; all accounts are subject to the mandatory and DAC policies. By default, a user's DAC owner and group will be set to the user's username.

7.4 IDENTIFICATION AND AUTHENTICATION

SFR	Description
FIA_AFL.1	The TOE will detect when an administrator configurable integer within 1 – unlimited unsuccessful authentication attempts for authentication based on username and password occur related to password-based authentication at the local console. Once the specified number of unsuccessful authentication attempts for an account has been met, the TOE locks the account.
FIA_UAU.5	The TOE supports authentication based on username and password at the local console. The TOE leverages the Pluggable Authentication Module (PAM) authentication mechanism. For password-based authentication, the username and password are compared to the set of credentials stored in the database. If the credentials match, then

SFR	Description
	the user is granted access to the TOE
FIA_X509_EXT.1 FIA_X509_EXT.2	The TOE validates X.509 certificates when presented during a TLS handshake. When an X.509 certificate is presented, the TOE verifies the certificate path, and certification validation process by verifying the following rules: • RFC 5280 certificate validation and certificate path validation. • The certificate path must terminate with a trusted CA certificate. • The OS shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates. • The TSF shall validate that any CA certificate includes caSigning purpose in the key usage field. • The OS shall validate the revocation status of the certificate using CRL as specified in RFC 5759. The validity is initiated by the opkg repository manager and performed by OpenSSL during connection establishment to the remote repository as well a package verification during installation. The package verification certificates are stored with the packages in the repository and are accessed during download/installation. If the package is not verified, it is not installed but discarded. The OS shall validate the extendedKeyUsage field according to the following rules: • Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field. • Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.

7.5 SECURITY MANAGEMENT

SFR	Description
FMT_MOF_EXT.1 FMT_SMF_EXT.1	The management functions identified in FMT_SMF_EXT.1 are restricted to administrators through the implementation of the RBAC, BL/B, and DAC access controls as described in section 0. The administrative functions are performed through the modification of the configuration file for the related function. Access to the configuration files is restricted based on the assigned security labels. The TOE allows administrators to perform the following

SFR	Description
	management functions: • Enable/disable session timeout • Configure session inactivity timeout • Configure local audit storage capacity • Configure minimum password length • Configure minimum number of special characters in password • Configure minimum number of numeric characters in password • Configure minimum number of uppercase characters in a password • Configure minimum number of lowercase characters in a password • Configure lockout policy for unsuccessful authentication attempts • Configure host-based firewall • Configure audit rules • Configure name/address of network time server • Configure network interfaces

7.6 PROTECTION OF THE TSF

SFR	Description
FPT_ACF_EXT.1	The OS implements access controls, as described in section 0, to protect the following relevant configuration files/locations: - Kernel Drivers/modules (`/xts/kernel.gz`) - Security Audit Logs (`/xts/slog`) - Shared Libraries (`/xts/lib32 & /xts/lib64`) - System executables (`/xts/bin`) - Kernel-based system configuration files (`/xts/cfg`) - Package-based system configuration files (PAM, etc.) (`/xts/etc` All file system objects are protected according to the mechanism described in section 7.3 of the ST. Configuration files are file system objects subject to the

SFR	Description
	systems security policy. The security label of the file system object defines the protection. Any security label can be obtained by an administrator typing: sec_label <pathname>
FPT_ASLR_EXT.1	The TOE always randomizes process address space memory locations with 34 bits of entropy. By default, there are no exceptions, however, authorized administrators may exempt any file, object, or binary using the `aslr_exempt_obj` command. The `aslr_exempt_obj` command instructs the program loader to turn off all address space randomization techniques regardless of how the program was compiled. This action can be used in rare cases where a predictable address space is required such as program debugging. Note, specifying this action within the null role effectively turns off ASLR for the entire system.

SFR	Description
FPT_SBOP_EXT.1	All OS binaries are compiled with the `fstack_protect_all` option to add a stack canary and associated verification code during the entry and exit of function frames to prevent stack-based buffer overflows. Package build harnesses will choose not to use the stack check protection in some cases and some binaries will not include it due to compiler, linker, and debugger optimization. The following executable files do not include the stack canary: /xts/bin • tic • getconf • ldconfig • ldd /xts/lib64 • libcrypt.a • libdl.a • libgmp.so.10.4.1 • libisl.so.15.0.0 • libm.a • libmpc.so.3.1.0 • libmpfr.so.4.1.3 • libpthread.a • libpthread_nonshared.a • libresolv.a • librt.a • libstdc++.a • libstdc++.so.6.0.25 • libthread_db-1.0.39.so
FPT_TST_EXT.1	As a result of power-on or reset, the Unified Extensible Firmware Interface (UEFI) firmware will load STOP's UEFI bootloader from the EFI system partition. The STOP UEFI bootloader is digitally signed with BAE System's own private vendor key. Before transferring execution to the STOP UEFI bootloader, the UEFI firmware will perform a digital signature check of the bootloader using BAE System's vendor key which must be pre-loaded into the UEFI firmware's secure boot keys configuration. The STOP UEFI bootloader will use UEFI services to obtain the memory map of the physical machine, and then use UEFI services to load the STOP kernel into memory at 1MB. The STOP UEFI bootloader is compiled with BAE System's public key in it, which it

SFR	Description
	will use to then perform a digital signature check of the STOP kernel. If the digital signature check succeeds, the bootloader will clear available physical memory other than the loaded kernel image and then transfer execution control to the kernel at the Kernel Startup entry point. Note that once the kernel begins execution, no UEFI services (including UEFI runtime services) will be used.
FPT_TUD_EXT.1 FPT_TUD_EXT.2	The TOE has the ability to check for updates to itself and application software by querying an enterprise hosted update repository. Update packages are cryptographically verified using RSA 4096 prior to installation.

7.7 TRUSTED PATH / CHANNELS

SFR	Description
FTP_ITC_EXT.1	The TOE provides a TLS Client protocol implementation which allows applications to protect communications with remote IT entities.
FTP_TRP.1	The TOE provides a trusted path with local users.

7.8 TIMELY SECURITY UPDATES

SAR	Description
ALC_TSU_EXT	STOP users can report security issues directly to BAE Systems via email to CSP support (csp@baesystems.com) or by phone (703-563-8124). For sensitive information, users may encrypt their email using Pretty Good Privacy (PGP). The public key for CSP Support is available at http://keyserver.pgp.com/vkd/DownloadKey.event?keyid=0xD7D0EA16C24B7039. Upon receipt of notification of a potential security issue from a STOP user, a STOP developer or support contact will immediately create a problem report in the issue tracking database. If a STOP user reports a potential security issue that is already known, the support contact captures the request in the issue tracking database and closes it as a duplicate of the previously-existing issue. Security issues are classified by severity based on the qualitative nature of the problem as well as the score assigned by the Common Vulnerability Scoring System (CVSS). The impact of a security issue will be determined by the description of the problem or the CVSS score, whichever results in a more severe impact. Within 10 business days of the confirmation of a security issue, STOP

SAR	Description
	engineering targets the issue for remediation in a specific release based on the following criteria: a) Critical security issues will be addressed in a minor release, and such minor release shall be issued within one week of resolution of a critical security problem. b) Important security issues will be addressed in a minor release which shall be issued within one month of resolution of an important security problem. c) Moderate impact security issues will be addressed in the next major release. d) Low-impact issues will be addressed as there is availability in the product roadmap. Within two weeks of the remediation (either interim or permanent) of a security issue, the security issue shall be reported to TOE users along with corrective action that TOE users can take to correct the issue or to mitigate the risk of the issue being exploited in their environment. Notifications of security issues are sent to users from the xts-support@baesystems.com account. The list of users is maintained as a contact in the xts-support@baesystems.com account. Any STOP user with an active support contract is entitled to receive updated releases, including releases containing security fixes. All security issue announcements will contain instructions for TOE users to request the update be provided to them.

8 ACRONYMS

The following acronyms are used in this ST:

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
ASLR	Address Space Layout Randomization
BL	Bell-LaPadula
B	Biba
BL/B	Bell-LaPadula-Biba
CA	Certificate Authority
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CC	Common Criteria
CN	Common Name
CPU	Central Processing Unit
CRL	Certificate Revocation List
CTR	Counter Mode
CVSS	Common Vulnerability Scoring System
DAC	Discretionary Access Control
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
DVD	Digital Video Disc
EAL	Evaluation Assurance Level
ECC	Elliptic Curve Cryptography
EKU	Extended Key Usage
EP	Extended Package
FFC	Finite-Field Cryptography

Acronym	Definition
FIPS	Federal Information Processing Standards
GB	Gigabyte
HMAC	Hash Message Authentication Code
IP	Internet Protocol
ISO	International Standards Organization
IT	Information Technology
JDK	Java Development Kit
MB	Megabyte
MLS	Multi-Level Security
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology (US)
OCSP	Online Certificate Status Protocol
OID	Object Identifier
OS	Operating System
OSP	Organizational Security Policy
PAM	Pluggable Authentication Module
PDF	Portable Document Format
PP	Protection Profile
RA	Registration Authority
RBAC	Role Based Access Control
RFC	Request for Comments
RSA	Rivest, Shamir and Adleman
RUWD	Read Up, Write Down
SAN	Subject Alternative Name
SFP	Security Function Policy
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SHS	Secure Hashing

Acronym	Definition
SP	Special Publication
SSL	Secure Sockets Layer
ST	Security Target
STOP	Secure Trusted Operating Program
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
UEFI	Unified Extensible Firmware Interface
URI	Uniform Resource Identifier
VPN	Virtual Private Network
WURD	Write Up, Read Down

Table 12 – Acronyms

9 APPENDIX A – CAVP CERTIFICATES & USAGE

The following algorithms are used by the TOE in the evaluated configuration. The vendor affirms that no source code changes were made to the cryptographic algorithm implementation prior to recompilation into the TOE.

Algorithm	Key Sizes / Digest	Usage	CAVP Cert #
AES-CBC	128-bit 256-bit	TLS Encryption/Decryption	AES 3264
AES-GCM	128-bit 256-bit	TLS Encryption/Decryption	AES 3264
RSA (186-4)	2048 3072	TLS Signature Generation/Verification	RSA 1664
ECDSA (186-4)	P-256 P-384 P-521	TLS Signature Generation/Verification	ECDSA 620
DH		Diffie-Hellman Key Establishment for TLS	N/A
SHS	SHA-1 SHA2-256 SHA2-384 SHA2-512	TLS Cryptographic Hashing	SHS 2702
HMAC	HMAC-SHA1 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-512	TLS Message Authentication	HMAC 2063
CTR DRBG	256-bit	Random Bit Generation	DRBG 723

Table 13 – Cryptographic Operations