Oracle Linux 8.4 Security Target

April 12, 2023

v1.12

Prepared By:
Acumen Security
2400 Research Blvd Suite 395
Rockville, MD, 20850
www.acumensecurity.net

Prepared for:
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
USA

Trademarks

Oracle Linux and the Oracle logo are trademarks or registered trademarks of Oracle Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Intel, Xeon, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Legal Notice

Table Of Contents

Revision History

Version	Date	Description
0.1	7/22/2021	Initial Draft.
0.2	7/26/2021	Updating processor information.
0.3	7/30/2021	Finalization of draft
0.4	9/2/2021	Incorporating customer comments
0.5	9/21/2021	Adding CAVPs
0.6	9/22/2021	Finalizing document
0.7	10/5/2021	Addressing ST review comments
0.8	11/8/2021	Minor updates to the ST and addressing OR comments.
0.9	11/18/2021	Addressing internal QA comments.
1.0	02/02/2022	Removing SSHPKG.
1.1	02/02/2022	Minor updates based on evaluator review.
1.2	02/09/2022	Addressing ST review comments.
1.3	03/31/2022	Addressing ST Comments.
1.4	04/13/2022	Addressing ST Comments.
1.5	5/13/2022	Updates to FCS_STO_EXT.1.
1.6	7/19/2022	Addressed internal ORs.
1.7	8/12/2022	Reinstate SSHPKG.
1.8	11/07/2022	Address internal comments to incorporate information associated with the SSHPKG.
1.9	11/14/2022	Address internal ORs.
1.10	1/25/2023	Address certifier ORs.
1.11	3/17/2023	Update CAVP certificate numbers.
1.12	4/12/2023	Update documentation references.

1 Security Target Introduction

1.1 Security Target and TOE Reference

This section provides information needed to identify and control this ST and its TOE.

Category	Identifier
ST Title	Oracle Linux 8.4 Security Target
ST Version	1.12
ST Date	April 12, 2023
ST Author	Acumen Security, LLC.
TOE Identifier	Oracle Linux 8.4 with the following package updates: Note: Dependencies for additional packages will be installed automatically.

kernel-uek	5.4.17-2136.312.3.4.el8uek
openssl	1:1.1.1k-7.el8_6
platform-python	3.6.8-47.0.1.el8_6
expat	2.2.5-8.0.1.el8_6.3
file	5.33-20.el8
glibc	2.28-189.5.0.1.el8_6
gnutls	3.6.16-5.el8_6
grub2-common	2.02-123.0.10.el8_6.8
nettle	3.4.1-7.el8
libsolv	07.20-1.el8
libtirpc	1.1.4-6.el8
libxml2	2.9.7-13.el8_6.1
lz4-libs	1.8.3-3.el8_4
nss	3.79.0-10.el8_6
polkit	0.115-13.0.1.el8_5.2
sssd-common	2.6.2-4.0.2.el8_6.1
vim-minimal	2:8.0.1763-19.0.1.el8_6.4
bind-export-libs	32:9.11.36-3.el8_6.1
c-ares	1.13.0-6.el8
cpio	2.12-11.el8
cryptsetup-libs	2.3.7-2.el8
curl	7.61.1-22.el8_6.4
cyrus-sasl-lib	2.1.27-6.el8_5
dnf	4.7.0-8.0.1.el8
libgcc	8.5.0-10.1.0.1.el8_6
libgomp	8.5.0-10.1.0.1.el8_6
libssh	0.9.6.3.el8
libstdc++	8.5.0-10.1.0.1.el8_6
glib2	2.56.4-158.el8_6.1

Category	Identifier	
	gnupg2	2.2.20-3.el8_6
	gzip	1.9-13.el8_5
	json-c	0.13.1-3.el8
	kpartx	0.8.4-22.el8_6.2
	libarchive	3.3.3-3.el8_5
	libgcrypt	1.8.5-7.el8_6
	libksba	1.3.5-8.el8_6
	lua-libs	5.3.4-12.el8
	microcode_ctl	4:20220207-1.20220510.1.0.1.el8_6
	ncurses	6.1-9.20180224.el8
	openssh	8.0p1-13.el8
	pcre	8.42-6.el8
	pcre2	10.32-3.el8_6
	rpm	4.14.3-24.el8_6
	rsyslog	8.2102.0-7.el8_6.1
	shim	15.6-1.0.3.el8
	systemd	239-58.0.1.el8.6.8
	sqlite-libs	3.26.0-16.el8_6
	udisks2	2.9.0-9.el8
	xz	5.2.4-4.el8_6
	zlib	1.2.11-19.el8_6
	NetworkManager	1:1.36.0-9.0.1.el8_6
	kexec-tools	2.0.20-68.0.3.el8
	libsepol	2.9-3.el8
	platform-python-pip	9.0.3-22.el8
	libsss_autofs	2.6.2-4.0.2.el8_6.1
	libsss_sudo	2.6.2-4.0.2.el8_6.1
	sssd-nfs-idmap	2.6.2-4.0.2.el8_6.1
	python3-pip-wheel	9.0.3-22.el8
TOE Software Version	8.4	
TOE Kernel	UEK Kernel	
TOE Developer	Oracle Corporation	
Key Words	Operating System, Oracle, Linux 8.4	

Table 1 - TOE/ST Identification

1.2 TOE Overview

Oracle Linux 8.4 (herein referred to as the TOE) is a Linux-based operating system. Oracle Linux is a general purpose, multi-user, multi-tasking Linux based operating system. It provides a platform for a variety of applications.

1.2.1 TOE Product Type

The TOE type is a Linux-based general-purpose operating system which supports secure remote login and other secure network services over an untrusted network using Secure Shell (SSH). It satisfies all the criterion to meet the Protection Profile for General Purpose Operating Systems Version 4.2.1 [GPOSPP] and the Functional Package for SSH Version 1.0 [PKG_SSH_V1.0].

1.3 TOE Architecture

1.3.1 Physical Boundaries

The evaluated configuration includes the general-purpose hardware with the following processors:

- Oracle Linux 8.4 on AMD EPYC 7551
- Oracle Linux 8.4 on Intel Skylake Xeon Platinum 8167M

The Target of Evaluation is based on the following system software:

- Oracle Linux 8.4

The TOE and its documentation are supplied on ISO images distributed via the Oracle Linux web site.

In addition to the installation media, the following documentation is provided:

- Evaluated Configuration Guide published by Oracle at the end of the evaluation
- Manual pages for all applications, configuration files and system calls

1.3.2 Logical Scope of the TOE

The TOE implements the following security functional requirements from [GPOSPP] and [SSHPKG] as listed below:

1.3.2.1 Audit Data Generation (FAU)

The TOE generates audit events for all start-up and shut-down functions, and all auditable events as specified in Section 6.2.1. The TOE leverages the Lightweight Audit Framework (LAF) audit system. Audit events are generated for the following audit functions:

- Start-up and shut-down of the audit functions;
- 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 contains the date and time of the event, type of event, subject identity (if applicable), and outcome (success or failure) of the event.

1.3.2.2 Cryptographic Support (FCS)

The TOE provides cryptographic support for the services described in Table 2. The TOE leverages the Oracle Linux 8.4 OpenSSL with AESNI, SHA1 AVX, SHA2 ASM cryptographic library for SSHv2 and TLS v1.2 related cryptographic operations. The related CAVP validation details are provided in Table 3.

The TOE provides an interface for the protection of stored credentials and includes AES-CBC and AES-GCM with key sizes of 128 and 256 bits along with SHA1, SHA-256, SHA-384, and SHA-512. The related CAVP validation details are provided in Table 3.

The cryptographic services provided by the TOE are described below.

Cryptographic Method	Usage
FCS_CKM.1 Cryptographic Key Generation (Refined)	• RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3. • RSA Key sizes supported are 2048 bits, 3072 bits, and 4096 bits. • 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. • Elliptic NIST curves supported are: P-256, P-384 and P-521. • FFC scheme using cryptographic key sizes of 2048 bits or greater.
FCS_CKM.2 Cryptographic Key Establishment (Refined)	• 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 conforming to NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography.
FCS_CKM_EXT.4 Cryptographic Key Destruction	• For volatile memory, the destruction shall be executed by removal of power to the memory. • For non-volatile memory, destruction consists of the invocation of an interface provided by the underlying platform that instructs the underlying platform to destroy the abstraction that represents the key.
FCS_COP.1(1) Cryptographic Operation - Encryption/Decryption (Refined)	• AES-CBC (as defined in NIST SP 800-38A) • AES-CTR (as defined in NIST SP 800-38A) • AES-GCM (as defined in NIST SP 800-38D)

Cryptographic Method	Usage
	• AES key sizes supported are 128 bits and 256 bits
FCS_COP.1(2) Cryptographic Operation - Hashing (Refined)	• Cryptographic hashing services conforming to FIPS Pub 180-4. • Hashing algorithms supported are: SHA-1, SHA-256, SHA-384 and SHA-512. • Message digest sizes supported are 160 bits, 256 bits, 384 bits and 512 bits.
FCS_COP.1(3) Cryptographic Operation - Signing (Refined)	• RSA digital signature algorithm conforming to FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 4. • ECDSA schemes using "NIST curves" P-256, P-384 and [P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5 • RSA key sizes supported are: 2048, 3072, and 4096 bits.
FCS_COP.1(4) Cryptographic Operation - Keyed-Hash Message Authentication (Refined)	• Keyed-hash message authentication services in conforming to FIPS Pub 198-1 The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard. • Keyed hash algorithm authentication services in accordance with the following specified cryptographic algorithms: SHA-1, SHA-256, SHA-384 and SHA-512. • Key sizes supported are: 112 bits. • Message digest sizes supported are: 160 bits, 256 bits, 384 bits and 512 bits.
FCS_RBG_EXT.1 Random Bit Generation	• Random number generation conforming to NIST Special Publication 800-90A. • The TOE leverages CTR_DRBG (AES). • The deterministic RBG used by the OS is seeded by an entropy source that accumulates entropy from a 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.
FCS_STO_EXT.1 Storage of Sensitive Data	• The OS implements functionality to encrypt sensitive data stored in non-volatile storage and provides interfaces to applications to invoke the functionality.

Cryptographic Method	Usage
FCS_SSH_EXT.1 SSH Protocol	• SSH protocol that complies with RFCs 4251, 4252, 4253, 4254 and 6668 as a client and server. • The TOE supports password-based authentication (RFC 4252) and public key authentication (RFC 4252). • The following public key algorithm is supported: rsa-sha2-256 (RFC 8332), rsa-sha2-512 (RFC 8332), ecdsa-sha2-nistp256 (RFC 5656), ecdsa-sha2-nistp384 (RFC 5656), and ecdsa-sha2-nistp521 (RFC 5656). • The SSH client shall ensure that, as described in RFC 4253, packets greater than 262144 bytes in an SSH transport connection are dropped. • The TOE supports the following encryption algorithms: aes128-ctr (RFC 4344), aes256-ctr (RFC 4344), aes128-cbc (RFC 4253), aes256-cbc (RFC 4253). • The TOE supports the following data integrity MAC algorithms: hmac-sha2-256 (RFC 6668), and hmac-sha2-512 (RFC 6668). • The TOE supports the following key exchange algorithm: ecdh-sha2-nistp256 (RFC 5656), ecdh-sha2-nistp384 (RFC 5656), and ecdh-sha2-nistp521 (RFC 5656). • The TOE uses SSH KDF as defined in • RFC 4253 (Section 7.2), and RFC 5656 (Section 4), • to derive the following cryptographic keys from a shared secret. • The TOE supports a rekey of the session keys occurs when any of the following thresholds are met: one hour of connection time, no more than one gigabyte of transmitted data, or no more than one gigabyte of received data.
FCS_SSHC_EXT.1 SSH Protocol - Client FCS_SSHS_EXT.1 SSH Protocol - Server	• The TOE shall authenticate its peer (SSH server) using a local database by associating each host name with a public key corresponding to the following: rsa-sha2-256 (RFC 8332), rsa-sha2-512 (RFC 8332), ecdsa-sha2-nistp256 (RFC 5656), ecdsa-sha2-nistp384 (RFC 5656), and ecdsa-sha2-nistp521 (RFC 5656) as described in RFC 4251 section 4.1. • The TSF shall authenticate itself to its peer (SSH Client) using: rsa-sha2-256 (RFC 8332), rsa-sha2-512 (RFC 8332), ecdsa-sha2-nistp256 (RFC 5656),

Cryptographic Method	Usage
	ecdsa-sha2-nistp384 (RFC 5656), and ecdsa-sha2-nistp521 (RFC 5656).
FCS_TLSC_EXT.1 TLS Client Protocol	• The TOE supports TLS v1.2 protocol • Supports the following cipher suites in the evaluated configuration: • 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_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_GCM_SHA384 as defined in RFC 5289
FCS_TLSC_EXT.2 TLS Client Protocol	• The OS shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: secp256r1, secp384r1, and secp521r1.

Table 2 - TOE Cryptographic Protocols

Each of these cryptographic algorithms have been validated for conformance to the requirements specified in their respective standards, as identified below.

Algorithms	Standards	Implementation	CAVP Certificates #	Processors
AES	• AES-CBC (as defined in NIST SP 800-38A) • AES-GCM (as defined in NIST SP 800-38D)	OpenSSL (64 bit) R8-8.6	A3381, A3382, A3383, A3392,	• AMD EPYC 7551

Algorithms	Standards	Implementation	CAVP Certificates #	Processors
	• AES-CTR (as defined in NIST SP 800-38A)		A3393, A3394, A3395, A3396, A3397, A3398, A3399, A3400	• Intel Xeon Platinum 8167M
RSA	• FIPS PUB 186-4 Digital Signature Standard (DSS), Appendix B.3.	• OpenSSL (64 bit) R8-8.6	A3401, A3402, A3403, A3404	• AMD EPYC 7551 • Intel Xeon Platinum 8167M
ECDSA	• FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	• OpenSSL (64 bit) R8-8.6	A3401, A3402, A3403, A3404	• AMD EPYC 7551 • Intel Xeon Platinum 8167M
ECDH	• NIST Special Publication 800-56A Revision 3	• OpenSSL (64 bit) R8-8.6	A3401, A3402, A3403, A3404	• AMD EPYC 7551 • Intel Xeon Platinum 8167M
DH	• Complies with RFC 3526	• OpenSSL (64 bit) R8-8.6	A3406	• AMD EPYC 7551 • Intel Xeon Platinum 8167M
KAS/CVL ECC	• NIST Special Publication 800-56A	• OpenSSL (64 bit) R8-8.6	A3401, A3402, A3403, A3404	• AMD EPYC 7551 • Intel Xeon Platinum 8167M
KAS/CVL FCC	• NIST Special Publication 800-56A	• OpenSSL (64 bit) R8-8.6	A3406	• AMD EPYC 7551 • Intel Xeon Platinum 8167M
HMAC	• Keyed-hash message authentication services in conforming to FIPS Pub 198-1 The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard	OpenSSL (64 bit) R8-8.6	A3401, A3402, A3403, A3404	• AMD EPYC 7551 • Intel Xeon Platinum 8167M
SHS	• NIST FIPS Pub 180-4.	OpenSSL (64 bit) R8-8.6	A3401, A3402, A3403, A3404	• AMD EPYC 7551 • Intel Xeon Platinum 8167M
DRBG	• Random number generation conforming to NIST Special Publication 800-90A.	• OpenSSL (64 bit) R8-8.6	A3381, A3382, A3383	• AMD EPYC 7551

Algorithms	Standards	Implementation	CAVP Certificates #	Processors
				• Intel Xeon Platinum 8167M
CVL SSH v2	• KDF 800-135	• OpenSSL (64 bit) • R8-8.6	A3385, A3386, A3387, A3388	• AMD EPYC 7551 • Intel Xeon Platinum 8167M •
CVL TLS v1.2	• KDF 800-135	• OpenSSL (64 bit) R8-8.6	A3401, A3402, A3403, A3404	• AMD EPYC 7551 • Intel Xeon Platinum 8167M

Table 3 - CAVP Algorithm Testing References

1.3.2.3 User Data Protection (FDP)

The TOE implements access controls which prevents unprivileged users from accessing files and directories owned by other users. The TOE provides an interface which allows VPN client to protect all IP traffic using IPSEC protocol.

1.3.2.4 Identification and Authentication (FIA)

All users must be authenticated to the TOE prior to carrying out any management actions. The TOE supports password-based authentication and public key based authentication. The OS disables user accounts after a configurable number of unsuccessful authentication attempts.

1.3.2.5 Security Management (FMT)

The TOE is capable of performing management functions. The administrator has full access to carry-out all management functions and the user has limited privilege.

1.3.2.6 Protection of the TSF (FPT)

The TOE implements the following protection of TSF data:

- Access Controls
- Address Space Layout Randomization
- Stack buffer overflow protection using stack canaries.
- Verification of integrity of the bootchain
- Trusted software updates

1.3.2.7 Trusted Path/Channels

The TOE supports TLS v1.2 and SSH v2 for trusted channel implementation. The TOE supports remote CLI using SSH v2 for secure remote administration.

1.4 Excluded Functionality

The following interfaces are not included as part of the evaluated configuration. All interfaces below are disabled in the evaluated configuration:

Functions	Exclusion discussion
GUI	A graphical user interface for system administration or any other operation is not included in the evaluated configuration.
LSM Support	The mandatory access control functionality offered by the Linux Security Module (LSM) framework found in the Linux kernel is not assessed by the evaluation and disabled in the evaluated configuration. All LSM modules such as SELinux, AppArmor, SMACK and others are not assessed as part of the evaluation. The evaluated configuration enables aspects of the LSM though.
GSS-API Security Mechanisms	The GSS-API is used to secure the connection between different audit daemons. The security mechanisms used by the GSS-API, however, are disabled in the evaluated configuration.

Table 4 - Excluded Functionality

1.5 TOE Documentation

The following documents are available in PDF formats.

Documentation	Version	Date
Oracle Linux 8.4 Common Criteria Guidance Document	Version 0.8	12 April, 2023
Oracle Linux 8 Installing Oracle Linux	F13930-24	August 2022
Oracle Linux 8 Enhancing System Security	F22907-21	May 2022
Oracle Linux Connecting to Remote Systems with OpenSSH	F22963-09	June 2022
Oracle Linux 8 Setting Up System Users and Authentication	F21455-09	November 2022

Table 5 - TOE Documentation

1.6 Other References

- Protection Profile for General Purpose Operating Systems, Version 4.2.1 [GPOSPP]

- Functional Package for Secure Shell (SSH), Version 1.0 [SSHPKG]

2 Conformance Claims

2.1 CC Conformance

This TOE is conformant to:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017: Part 3 extended

2.2 Protection Profile Conformance

This TOE is conformant to:

- Protection Profile for General Purpose Operating Systems, Version 4.2.1 [GPOSPP]
- Functional Package for Secure Shell (SSH), Version 1.0 [SSHPKG]

2.3 Conformance Rationale

This Security Target provides exact conformance to [GPOSPP] and [SSHPKG]. The security problem definition, security objectives and security requirements in this Security Target are all taken from the Protection Profile performing only operations defined there.

2.3.1 Technical Decisions

All NIAP Technical Decisions (TDs) issued to date that are applicable to [GPOSPP] and [SSHPKG] have been addressed. The following tables identify all applicable TDs:

Identifier	Applicable	Exclusion Rationale (if applicable)
TD0715: Update to FIA_X509_EXT.1 for Exception Processing and Test Conditions	Yes	
TD0680: Conformance Claims section updated to allow for MOD_WLAN_CLI_v1.0	Yes	
TD0649: Conformance claims for OS PP v4.2.1	Yes	
TD0630: FCS_COP.1 requirements for Secure Shell	Yes	
TD0600: Conformance claim sections updated to allow for MOD_VPNC_V2.3	Yes	
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 Update	Yes	
TD0365 – FCS_CKM_EXT.4 selections	Yes	

Table 6 - PP_OS_V4.2.1 Technical Decisions

Identifier	Applicable	Exclusion Rationale (if applicable)
TD0695: Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package.	Yes	
TD0694: FCS_SSH_EXT.1.3 Inconsistency	Yes	
TD0682: Addressing Ambiguity in GCS_SSHS_EXT.1 Tests	Yes	

Table 7 - PKG_SSH_V1.0 Technical Decisions

3 Security Problem Definition

The security problem definition has been taken from [GPOSPP] and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any organizational security policies that the TOE is expected to enforce.

3.1 Threats

The following threats are drawn directly from the [GPOSPP].

ID	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.
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 8 - Threats

3.2 Assumptions

The following assumptions are drawn directly from the [GPOSPP].

ID	Assumption
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 9 - Assumptions

3.3 Organizational Security Policies

The [GPOSPP] and [SSHPKG] do not define any OSPs.

4 Security Objectives

The security objectives for the TOE have been taken from [GPOSPP] and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the TOE

The following subsections describe objectives for the TOE.

ID	Objective for the Operation Environment
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.
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 10 - Security Objectives for the TOE

4.2 Security Objectives for the Operational Environment

The following security objectives for the operational environment assist the TOE in correctly providing its security functionality. These track with the assumptions about the environment.

ID	Objective for the Operation Environment
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 11 - Objectives for the Operational Environment

4.3 Rationale for Security Objectives

The following section describes how the assumptions, threats, and organizational security policies map to the security objectives.

Threat, Assumption, or OSP	Security Objectives	Rationale
T.NETWORK_ATTACK	O.PROTECTED_COMMS, O.INTEGRITY, O.MANAGEMENT O.ACCOUNTABILITY	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, Assumption, or OSP	Security Objectives	Rationale
T.NETWORK_EAVESDROP	O.PROTECTED_COMMS, O.MANAGEMENT	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.
T.LOCAL_ATTACK	O.INTEGRITY O.ACCOUNTABILITY	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.
T.LIMITED_PHYSICAL_ACCESS	O.PROTECTED_STORAGE	The objective O.PROTECTED_STORAGE protects against unauthorized attempts to access physical storage used by the TOE.
A.PLATFORM OE.PLATFORM	OE.PLATFORM	The operational environment objective OE.PLATFORM is realized through A.PLATFORM.
A.PROPER_USER	OE.PROPER_USER	The operational environment objective OE.PROPER_USER is realized through A.PROPER_USER.
A.PROPER_ADMIN	OE.PROPER_ADMIN	The operational environment objective OE.PROPER_ADMIN is realized through A.PROPER_ADMIN.

Table 12 - Rationale for Security Objectives

5 Extended Security Functional Components

Requirements	Descriptions
FCS_RBG_EXT.1	Random Bit Generation
FCS_STO_EXT.1	Storage of Sensitive Data
FCS_SSH_EXT.1	SSH Protocol
FCS_SSHC_EXT.1	SSH Protocol - Client
FCS_SSHS_EXT.1	SSH Protocol - Server
FCS_TLSC_EXT.1	TLS Client Protocol
FDP_IFC_EXT.1	Information flow control
FDP_ACF_EXT.1	Access Controls for Protecting User Data
FIA_X509_EXT.1	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FMT_MOF_EXT.1	Management of security functions behavior
FMT_SMF_EXT.1	Specification of Management Functions
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

Table 13 - Extended Security Functional Components

5.1 Extended Security Functional Components Rationale

The definition of all SFRs with the appendix of "_EXT" is supplied by the protection profile. All extended security functional components are derived directly from the [GPOSPP] and [SSHPKG] and applied verbatim. Please refer to Section 9 Annex B - Extended Security Functional Components.

6 Security Requirements

This section identifies the Security Functional Requirements for the TOE. The Security Functional Requirements included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, dated: April 2017 and all international interpretations.

Requirements	Descriptions
FAU_GEN.1	Audit Data Generation (Refined)
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
FCS_SSH_EXT.1	SSH Protocol
FCS_SSHC_EXT.1	SSH Protocol - Client
FCS_SSHS_EXT.1	SSH Protocol - Server
FCS_TLSC_EXT.1	TLS Client Protocol
FCS_TLSC_EXT.2	TLS Client Protocol
FDP_IFC_EXT.1	Information flow control
FDP_ACF_EXT.1	Access Controls for Protecting User Data
FIA_AFL.1	Authentication Failure Management (Refined
FIA_UAU.5	Multiple Authentication Mechanisms (Refined)
FIA_X509_EXT.1	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FMT_MOF_EXT.1	Management of security functions behavior
FMT_SMF_EXT.1	Specification of Management Functions
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
FTP_TRP.1	Trusted Path

Table 14 - SFRs

6.1 Conventions

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the following font conventions to identify the operations defined by the CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with <u>underlined</u> text;
- Iteration: Indicated by appending the SFR name with a sequential number in parentheses, e.g. FCS_COP.1(1).
- Where operations were completed in the PP or FP itself, the formatting used in the PP or FP has been retained.
- Extended SFRs are identified by the addition of "EXT" after the requirement name.

Extended SFRs are identified by having a label 'EXT' after the requirement name. Formatting conventions outside of operations matches the formatting specified within the PP or FP.

6.2 Security Functional requirements

6.2.1 Security Audit (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 shut-down 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);**
   - **[*<u>no other specifically defined auditable events</u>*]**

   ]].

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 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, [*User identity (if applicable)*].

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: [

- ***RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3,***
- ***ECC schemes using "NIST curves" P-256, P-384 and [P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4,***
- ***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 Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”,***

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

6.2.2.3 FCS_CKM_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*
- *For non-volatile memory that consists of [the invocation of an interface provided by the underlying platform that [*
    - *instructs the underlying platform to destroy the abstraction that represents the key]*

  ]

].

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),***
- ***AES-CTR (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 using "NIST curves" P-256, P-384 and [P-521] that meet the following: 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 [*112 bits used in HMAC*] **and message digest sizes [*160 bits, 256 bits, 384 bits, 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 [

- 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_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1 The TOE shall implement *SSH* acting as a [*client, server*] in accordance with that complies with RFCs 4251, 4252, 4253, 4254, [*4344, 5656,6668, 8332*] and [*no other standard*].

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

- *"password" (RFC 4252),*
- *"publickey" (RFC 4252): [*
    - *rsa-sha2-256 (RFC 8332)*
    - *rsa-sha2-512 (RFC 8332)*
    - *ecdsa-sha2-nistp256 (RFC 5656),*
    - *ecdsa-sha2-nistp384 (RFC 5656),*
    - *ecdsa-sha2-nistp521 (RFC 5656)*

    ]

    ] and no other methods.

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*262144*] in an SSH transport connection are dropped.

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

- *aes128-ctr (RFC 4344),*

- *aes256-ctr (RFC 4344),*
- *aes128-cbc (RFC 4253),*
- *aes256-cbc (RFC 4253),*

] and no other mechanisms.

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

- *hmac-sha2-256 (RFC 6668),*
- *hmac-sha2-512 (RFC 6668),*

] and no other mechanisms.

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

[

- *ecdh-sha2-nistp256 (RFC 5656),*
- *ecdh-sha2-nistp384 (RFC 5656),*
- *ecdh-sha2-nistp521 (RFC 5656)*

] and no other mechanisms.

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

- *RFC 4253 (Section 7.2),*
- *RFC 5656 (Section 4),*

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

FCS_SSH_EXT.1.8 The TSF shall ensure that [

- *a rekey of the session keys*
  ] occurs when any of the following thresholds are met:
- one hour connection time
- no more than one gigabyte of transmitted data, or
- no more than one gigabyte of received data.

6.2.2.11 FCS_SSHC_EXT.1 SSH Protocol - Client

FCS_SSHC_EXT.1.1 The TSF shall authenticate its peer (SSH server) using: [

- *using a local database by associating each host name with a public key corresponding to the following list:*
  [
  - *rsa-sha2-256 (RFC 8332),*
  - *rsa-sha2-512 (RFC 8332),*
  - *ecdsa-sha2-nistp256 (RFC 5656),*
  - *ecdsa-sha2-nistp384 (RFC 5656),*
  - *ecdsa-sha2-nistp521 (RFC 5656)*

  ]

as described in RFC 4251 section 4.1.

6.2.2.12 FCS_SSHS_EXT.1 SSH Protocol – Server

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

- *rsa-sha2-256 (RFC 8332),*
- *rsa-sha2-512 (RFC 8332),*
- *ecdsa-sha2-nistp256 (RFC 5656),*
- *ecdsa-sha2-nistp384 (RFC 5656),*
- *ecdsa-sha2-nistp521 (RFC 5656)*

6.2.2.13 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_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_GCM_SHA384 as defined in RFC 5289*

].

FCS_TLSC_EXT.1.2

The OS shall verify that the presented identifier matches the reference identifier per RFC 6125.

FCS_TLSC_EXT.1.3

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

6.2.2.14 FCS_TLSC_EXT.2 TLC Client Protocol

FCS_TLSC_EXT.2.1 The **OS** shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [secp256r1, secp384r1, secp521r1].

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.

6.2.3.2 FDP_IFC_EXT.1 Information flow control

FDP_IFC_EXT.1.1 The OS shall [*provide an interface which allows a VPN client to protect all IP traffic using IPsec]* with the exception of IP traffic required to establish the VPN connection and [no other traffic].

Application Note: Typically, the traffic required to establish the VPN connection

6.2.4 Identification and Authentication (FIA)

6.2.4.1 FIA_AFL.1 Authentication Failure Management (Refined)

FIA_AFL.1.1 The **OS** shall detect when [

- *an Administrator configurable positive integer within [1-999]*

] 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 for an account 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,***
- ***for use in SSH only, SSH public key-based authentication as specified by the Functional Package for Secure Shell [SSHPKG]***

] to support user authentication.

FIA_UAU.5.2 The **OS** shall authenticate any user's claimed identity according to the [*authentication on the local console is based on user name and password, authentication via the SSHv2 protocol first performs the certificate-based authentication which is followed by the user name and password authentication if the certificate-based authentication was unsuccessful*].

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 [CRL as specified in RFC 8603] 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.*]

NOTE: TD0604 has been applied.

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 X.509 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 [*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	

Management Function	**Administrator**	**User**
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 [*limiting number of attempts during a time period*]	X	
Configure host-based firewall	X	
Configure name/address of directory server with which to bind		
Configure name/address of remote management server from which to receive management settings		
Configure name/address of audit/logging server to which to send audit/logging records	X	
Configure audit rules	X	
Configure name/address of network time server	X	
Enable/disable automatic software update	X	
Configure WiFi interface		
Enable/disable Bluetooth interface		
Enable/disable [*no other devices*]	X	
No other management functions	X	

Table 15 - Specification of Management Functions

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 *[32 bits*] of entropy except for [*the Linux kernel, non-Position-Independent-Executable applications, non-Position-Independent-Code shared libraries].*

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 <u>and shall use a digital signature scheme specified in FCS_COP.1(3) to validate the authenticity of the response</u>.

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 communication

FTP_ITC_EXT.1.1 The OS shall use [

- *<u>TLS as conforming to FCS_TLSC_EXT.1,</u>*
- *<u>SSH as conforming to the Functional Package for Secure Shell</u>*

] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: [*<u>server</u>*] 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 [remote, *<u>local</u>*] users that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from modification and disclosure.

FTP_TRP.1.2 The OS shall permit [*<u>the TSF, local users, remote users</u>*] 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.

6.3 TOE SFR Dependencies Rationale for SFRs

[GPOSPP] and [SSHPKG] contain all the requirements claimed in this Security Target. As such, the dependencies are not applicable since the PP and FP have been approved.

6.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from [GPOSPP] which are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in the table below.

Assurance Class	Components	Components Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documentation	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative Procedures
Life-Cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM Coverage
	ALC_TSU_EXT.1	Timely Security Updates
Tests	ATE_IND.1	Independent Testing – Conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Survey

Table 16 - Security Assurance Requirements

6.5 Rationale for Security Assurance Requirements

The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes). The development evidence also contains a tracing of the interfaces to the SFRs described in this ST.

6.6 Assurance Measures

The TOE satisfies the identified assurance requirements. This section identifies the Assurance Measures applied by Oracle to satisfy the assurance requirements. The table below lists the details.

SAR Component	How the SAR will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes).
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1 ALC_CMS.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.

SAR Component	How the SAR will be met
ALC_TSU_EXT.1	The security updates are flagged as Critical or High based on the CSS ratings and should be available to the public within 24 hours of the fix has been finalized. Oracle uses the utilizes CVSS 3.0 specification for scoring CVEs. Any low severity CVEs will be evaluated in the next release based on priority. For the kernel, there will be quarterly release for the UEK. Any low severity may be addressed in the next major release. In addition, there is also a monthly errata for UEK where pending high level security issues can be consolidated. To report, security vulnerabilities, users should follow the process outline in the following website: https://www.oracle.com/corporate/security-practices/assurance/vulnerability/reporting.html The following webpage provides links to published Errata where users can track any vulnerabilities. https://linux.oracle.com/security/ If there is a publicly known vulnerability, users can track progress on the remediation progress from the following link: https://linux.oracle.com/security One can search for CVEs or Oracle Linux 8.4 Security Errata. Users can sign up to the mailing list to be notified of security updates: https://oss.oracle.com/mailman/listinfo/el-errata to receive updates. Oracle customers and partners should use the "My Oracle Support to submit a service request for any security vulnerabilities that they may have discovered in the Oracle product. All other users, should submit an email to secalert_us@oracle.com with their observations. All users are strongly recommended to use email encryption using Oracle encryption key when contacting Oracle Security. Oracle works closely with the research community who find vulnerabilities and work with Oracle so that the security fixes can be issued to all customers.
ATE_IND.1	Oracle will provide the TOE for testing.
AVA_VAN.1	Oracle will provide the TOE for testing.

Table 17 - TOE Security Assurance Measures

7 TOE Summary Specification

This chapter identifies and describes how the Security Functional Requirements and Security Assurance Requirements identified above are met by the TOE.

TOE SFRs	Rationale
FAU_GEN.1 and FAU_GEN.2	The TOE leverages the Lightweight Audit Framework (LAF) audit system. Audit events are generated for the following audit functions: • Start-up and shut-down of the audit functions; • 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 contains the following information: Date and time of the event, type of event, subject identity (if applicable), and outcome (success or failure) of the event The audit trail is stored in files which are only accessible by administrators. Once the audit files are full, the administrator would be notified. Once the audit trail is full, the audit daemon will not allow new audit events from the kernel. The kernel buffer must be cleared before new audit events are allowed.
FCS_CKM.1	The TOE supports RSA key sizes of 2048, 3072, and 4096 bits for key generation conforming to FIPS PUB 186-4 Digital Signature Standard (DSS), Appendix B.3. The RSA keys are used in support of digital signatures for both TLS and SSH communications. The TOE supports 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. ECDSA is used in support of TLS and SSH communications. FFC Schemes using safe primes that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes" for use in TLS. Please refer to Table#3 Cryptographic Algorithm Certificates for NIST CAVPs for RSA, and ECDSA.
FCS_CKM.2	The TOE supports Cryptographic Key Establishment using the following schemes: • 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".

TOE SFRs	Rationale
	• Finite field-based key establishment conforming to NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography". Please refer to Table#3 Cryptographic Algorithm Certificates for NIST CAVPs for ECDSA and KAS/CVL FCC.
FCS_CKM_EXT.4	For volatile memory, the destruction shall be executed by removal of power to the memory. For non-volatile memory, the destruction consists of the invocation of an interface provided by the underlying platform that instructs the underlying platform to destroy the abstraction that represents the key. Symmetric key material and Diffie-Hellman / EC Diffie-Hellman public and private keys are derived using the SSH KDF and stored in volatile memory. Asymmetric key material is stored on hard disk. The /etc/ssh directory contains the host keys which are generated using ssh-keygen. The $HOME/.ssh contains user keys and are generated using ssh-keygen. Authorized public keys are generated remotely and input into the TOE. Symmetric session keys for TLS are derived from the TLS KDF or input through RSA key wrap.
FCS_COP.1(1)	The TOE supports AES encryption and decryption conforming to • CBC as specified in NIST SP 800-38A • CTR as specified in NIST SP 800-38A • GCM as specified in NIST SP 800-38D The AES key size supported are 128 bits and 256 bits and the AES modes supported are: CBC and GCM. The SSH software shall perform encryption/decryption services for data in accordance with a specified cryptographic algorithm AES-CTR and AES-CBC (as defined in NIST SP 800-38A) mode and cryptographic key sizes of 128-bits, and 256-bits. The TSF provides unique counter values for the AES-CTR algorithm. The OpenSSH module uses the OpenSSL module which does the AES CTR for SSH. A normal sequence of events would be to call ssh_aes_ctr_init() when a session is started, multiple calls to ssh_aes_ctr() to encrypt packets, and ssh_aes_ctr_cleanup to close out a session and free memory. The ssh_aes_ctr_init() function accepts a key and iv for the session (the iv is used as the initial value for the ctr). If the calling program (ssh or sshd) supplies an IV (ctr), it is used as the initial value for the counter, otherwise 0 is the initial value used. As encryption is done the with the ssh_aes_ctr function, the ssh_ctr_inc is called to increment the value of the counter by 1. Because the counter value is 128 bits (16 bytes), there is no direct instruction to add 1 to it, so the ssh_ctr_inc function does a loop to increment the value byte-by-byte and handles carries from low-order bytes

<table>
<tr><th>TOE SFRs</th><th>Rationale</th></tr>
<tr><td></td><td>to high-order bytes.
Since the counter is 128 bits, it would take a HUGE amount of time before a ctr value is re-used with a specific key because of roll-over.

Please refer to Table#3 Cryptographic Algorithm Certificates for NIST CAVPs for AES.</td></tr>
<tr><td>FCS_COP.1(2)</td><td>The TOE supports Cryptographic hashing services conforming to FIPS Pub 180-4. The hashing algorithms are used for signature services and HMAC services.

The following hashing algorithms supported: SHA-1, SHA-256, SHA-384 and SHA-512.
The message digest sizes supported are: 160 bits, 256 bits, 384 bits and 512 bits.

Please refer to Table #3 Cryptographic Algorithm Certificates for NIST CAVPs SHS.</td></tr>
<tr><td>FCS_COP.1(3)</td><td>The TOE provides Cryptographic signature generation and verification in accordance with the following cryptographic algorithms:
RSA digital signature algorithm conforming to FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 4.
The RSA key sizes supported are: 2048, and 3072, and 4096 bits.
ECDSA schemes using "NIST curves" P-256, P-384 and [P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5
The Elliptical curve key size supported is 256 bits.
Please refer to Table #3 Cryptographic Algorithm Certificates for NIST CAVPs for RSA.</td></tr>
<tr><td>FCS_COP.1(4)</td><td>The TOE supports Keyed-hash message authentication conforming to the Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard with the following algorithms:
Keyed hash algorithm authentication services in accordance with the following specified cryptographic algorithms: SHA-1, SHA-256, SHA-384 and SHA-512.
Key sizes supported are: 112 bits.
HMAC algorithms is used in support of TLS and SSH sessions.<table>
<tr><th>HMAC Algorithms</th><th>Hash Functions</th><th>Block Size</th><th>Key lengths</th><th>MAC lengths</th></tr>
<tr><td>HMAC-SHA-1</td><td>SHA-1</td><td>512 bits</td><td>160 bits</td><td>160 bits</td></tr>
<tr><td>HMAC-SHA-256</td><td>SHA-256</td><td>512 bits</td><td>256 bits</td><td>256 bits</td></tr>
<tr><td>HMAC-SHA-384</td><td>SHA-384</td><td>1024 bits</td><td>384 bits</td><td>384 bits</td></tr>
</table></td></tr>
</table>

TOE SFRs	Rationale
	<table><tr><td>HMAC-SHA-512</td><td>SHA-512</td><td>1024 bits</td><td>512 bits</td><td>512 bits</td></tr></table> Please refer to Table #3 Cryptographic Algorithm Certificates for NIST CAVPs for HMAC.
FCS_RBG_EXT.1	The TOE uses the following DRBG conforming to NIST Special Publication 800-90A: • CTR_DRBG(AES) The TOE leverages CTR_DRBG (AES). The deterministic RBG used by the OS is seeded by an entropy source that accumulates entropy from a 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. Please refer to Table #3 Cryptographic Algorithm Certificates for NIST CAVPs for DRBG.
FCS_STO_EXT.1	The TOE includes the Oracle Linux 8.4 OpenSSL which provides an interface to end-users to securely store sensitive data on the filesystem. OpenSSL provides file encryption services using AES-CBC and AES-GCM with 128 and 256 bit key sizes.
FCS_SSH_EXT.1	The SSH software shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254,4344,5656, 6668, and 8332 as a client and server.
FCS_SSH_EXT.1.2	The TOE supports password-based authentication (RFC 4252), and public key authentication (RFC 4252). The following public key algorithms are supported for authentication: rsa-sha2-256 (RFC 8332), rsa-sha2-512 (RFC 8332), ecdsa-sha2-nistp256 (RFC 5656), ecdsa-sha2-nistp384 (RFC 5656), and ecdsa-sha2-nistp521 (RFC 5656). This list conforms to FCS_SSH_EXT.1.2.
FCS_SSH_EXT.1.3	The TOE ensures that SSH packets that exceed 262144 bytes are dropped at the application layer per RFC 4253. This large packet size is typical for Linux implementations. Once SSH packets are received, it is verified that it contains the packet length, padding length, payload and random padding. Once the packet information has been verified then the packet is decrypted. The packets are stored in a buffer. If the packet size is larger than permitted, the SSH packets are dropped, and the connection is terminated.
FCS_SSH_EXT.1.4	The TOE supports the following encryption algorithms: aes128-ctr (RFC 4344), aes256-ctr (RFC 4344), aes128-cbc (RFC 4253), and aes256-cbc (RFC 4253). Optional characteristics are not supported. The encryption algorithms specified are identical to those listed for the component.

TOE SFRs	Rationale
FCS_SSH_EXT.1.5	The TOE supports the following data integrity HMAC algorithms: hmac-sha2-256 (RFC 6668) and hmac-sha2-512 (RFC 6668) Optional characteristics are not supported. The encryption algorithms specified are identical to those listed for the component.
FCS_SSH_EXT.1.6	The TOE supports the following key exchange algorithms: ecdh-sha2-nistp256 (RFC 5656), ecdh-sha2-nistp384 (RFC 5656), and ecdh-sha2-nistp521 (RFC 5656). The key exchange algorithms specified are identical to those listed for the component.
FCS_SSH_EXT.1.7	The TOE uses SSH KDF as defined in RFC 4253 (Section 7.2), and RFC 5656 (Section 4) to derive the following cryptographic keys from a shared secret: session keys.
FCS_SSH_EXT.1.8	The TOE ensures that a rekey of the session keys occurs when any of the following thresholds are met: one hour connection time and no more than one gigabyte of transmitted data or no more than one gigabyte of received data.
FCS_SSHC_EXT.1	The TOE shall authenticate its peer (SSH server) using a local database by associating each host name with a public key corresponding to the following: rsa-sha2-256 (RFC 8332), rsa-sha2-512 (RFC 8332), ecdsa-sha2-nistp256 (RFC 5656), ecdsa-sha2-nistp384 (RFC 5656), and ecdsa-sha2-nistp521 (RFC 5656) as described in RFC 4251 section 4.1.
FCS_SSHS_EXT.1	The TOE shall authenticate itself to its peer (SSH Client) using: rsa-sha2-256 (RFC 8332), rsa-sha2-512 (RFC 8332), ecdsa-sha2-nistp256 (RFC 5656), ecdsa-sha2-nistp384 (RFC 5656), and ecdsa-sha2-nistp521 (RFC 5656).
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_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_GCM_SHA384 as defined in RFC 5289* The cipher suites specified are identical to those listed for this component.
FCS_TLSC_EXT.1.2	The OS verifies that the presented identifier matches the reference identifier according to RFC 6125. The following reference identifiers are to be verified during the TLS channel establishment:

TOE SFRs	Rationale
	• DNS host name or IP address found in Common Name of the X.509 certificate. Wild cards are supported. • DNS host name found in the SAN for DNS names of the X.509 certificate. The TOE does not support URI reference identifiers, SRV reference identifiers, or certificate pinning.
FCS_TLSC_EXT.1.3	The OS establishes a trusted channel if the peer certificate is valid.
FCS_TLSC_EXT.2.1	The OS, by default, presents the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: secp256r1, secp384r1, and secp521r1.
FDP_ACF_EXT.1	The TOE provides support for POSIX type access control lists. ACL's can be used with the following file systems: • ext4 • XFS • OCSFS2 An ACL consists of a set of rules that specify how a specific user or group can access the file or directory with which the ACL is associated. A regular ACL entry specifies access information for a single file or directory. A default ACL entry is set on directories only and specifies default access information for any file within the directory that does not have an access ACL. Users can configure ACLs that define access rights for more than just a single user or group, and specify rights for programs, processes, files, and directories. If you set a default ACL on a directory, its descendants inherit the same rights automatically.
FDP_IFC_EXT.1	The TOE provides the XFRM framework with the XFRM netlink interface and it also provides the TUN/TAP interface for supporting user-space VPN clients operating at ISO/OSI level 2 or 3. Only IP traffic goes through the VPN and other traffic (DNS, etc) do not go through the VPN.
FIA_AFL.1	The TOE will detect when an administrator configurable integer within 1-999 unsuccessful authentication attempts for authentication based on user name and password occur related to authentication on local console, and password-based authentication via SSH v2 protocol. Once the specified number of unsuccessful authentication attempts for an account has been met, the OS shall lock the account.
FIA_UAU.5	The TOE supports authentication based on username and password and public key-based authentication. The TOE leverages the Pluggable Authentication Module (PAM) authentication mechanism. For password-based authentication, when the user provides the correct username and password, this is compared to the known user database and

TOE SFRs	Rationale
	if they match then the user is granted access. Otherwise, the user will not be granted access to the TOE. 'When using key-based authentication, the user must generate an RSA key pair. If the user uses public key-based authentication, the presented key is compared to the user's stored key. If the comparison is successful, then the user is granted access to the TOE. If the public key based authentication is unsuccessful, the user is prompted for a username and password.
FIA_X509_EXT.1	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, 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 TOE supports CRL as specified by RFC 5759. The OS shall validate the extendedKeyUsage field according to the following rules: o 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. o 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. o 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. o 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. o 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. o 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. (conditional) A Security Administrator can configure the TSF to use OCSP or CRL for revocation checking. The TOE validates X.509 certificates when presented as part of a TLS Server Hello during a handshake.

<table>
<tr><th>TOE SFRs</th><th>Rationale</th></tr>
<tr><td></td><td></td></tr>
<tr><td>FIA_X509_EXT.1.2</td><td>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.</td></tr>
<tr><td>FIA_X509_EXT.2</td><td>The TSF uses X.509v3 certificates for TLS connections only.</td></tr>
<tr><td>FMT_MOF_EXT.1</td><td>All management activities are restricted to the root user. Privileges to perform administrative actions are maintained by the TOE. These privileges are separated into privileges to act on data or access functionality in user space and in kernel space.

Functionality accessible in user space are applications that can be invoked by users. Also, data accessible in user space is either data maintained with an application or data stored in persistent or transient storage objects. Privileges are controlled by permissions to invoke applications and to access data. For example, the configuration files including the user databases of /etc/passwd and /etc/shadow are accessible to the root user only. Due to privileges being controlled by permissions, this prevents users from performing management functions that they do not have access to.</td></tr>
<tr><td>FMT_SMF_EXT.1</td><td>The TOE maintains the following roles: Administrator and User

The management functions are listed below:
<table>
<tr><th>Management Function</th><th>Administrator</th><th>User</th></tr>
<tr><td>Enable/disable [session timeout]</td><td>X</td><td></td></tr>
<tr><td>Configure [session] inactivity timeout</td><td>X</td><td></td></tr>
<tr><td>Configure local audit storage capacity</td><td>X</td><td></td></tr>
<tr><td>Configure minimum password Length</td><td>X</td><td></td></tr>
<tr><td>Configure minimum number of special characters in password</td><td>X</td><td></td></tr>
<tr><td>Configure minimum number of numeric characters in password</td><td>X</td><td></td></tr>
<tr><td>Configure minimum number of uppercase characters in password</td><td>X</td><td></td></tr>
<tr><td>Configure minimum number of lowercase characters in password</td><td>X</td><td></td></tr>
<tr><td>Configure lockout policy for unsuccessful authentication attempts through [limiting number of attempts during a time period]</td><td>X</td><td></td></tr>
<tr><td>Configure host-based firewall</td><td>X</td><td></td></tr>
</table></td></tr>
</table>

TOE SFRs	Rationale			
	Configure name/address of directory server with which to bind			
	Configure name/address of remote management server from which to receive management settings			
	Configure name/address of audit/logging server to which to send audit/logging records	X		
	Configure audit rules	X		
	Configure name/address of network time server	X		
	Enable/disable automatic software update	X		
	Configure WiFi interface			
	Enable/disable Bluetooth interface			
	Enable/disable [*no other devices*]	X		
	No other management functions	X		
FPT_ACF_EXT.1	The OS implements access control to the following security relevant data: • /lib/modules: contains Kernel modules and device drivers • /var/log/audit: contains audit data • /lib, /lib64, /usr/lib and /usr/lib64 contains shared libraries • /bin, /sbin, /usr/bin, and /usr/sbin contains system executables. • /etc: contains system configuration files. This access control prohibits unprivileged users from reading security audit logs and system-wide credential repositories.			
FPT_ASLR_EXT.1	The TOE always randomizes process address memory locations with 32 bits of entropy except for the Linux kernel, non-Position-Independent-Executable applications, non-Position-Independent-Code shared libraries.			
FPT_SBOP_EXT.1	The OS implements compiler flag stack-based buffer overflow protections. Application developers should use the following compiler options as best practice when developing applications invoking the gcc compiler and linker.			

TOE SFRs	Rationale

	The stack-protector-strong flag has been developed to broaden the scope of the stack protection without extending it to every function in the program.
	
	-fstack-protector-strong --param=ssp-buffer-size=4
	
	ASLR improves executable security in terms of memory randomization and access protection.
	
	-fpie -Wl,-pie
	
	The following library comes from the brotli package. The functions do not have an array on the stack, so they do not need stack protection.
	- /usr/lib64/libbrotlicommon.so.1.0.6
	
	The following library comes from the coreutils package. The functions do not have an array on the stack, so they do not need stack protection.
	- /usr/libexec/coreutils/libstdbuf.so
	
	The following libraries come from the fwupd package. The functions do not have an array on the stack, so they do not need stack protection.
	- /usr/lib64/fwupd-plugins-3/libfu_plugin_invalid.so
	- /usr/lib64/fwupd-plugins-3/libfu_plugin_iommu.so
	- /usr/lib64/fwupd-plugins-3/libfu_plugin_pci_bcr.so
	- /usr/lib64/fwupd-plugins-3/libfu_plugin_test.so
	- /usr/lib64/fwupd-plugins-3/libfu_plugin_uefi_recovery.so
	- /usr/lib64/fwupd-plugins-3/libfu_plugin_upower.so
	
	The following libraries come from the gawk package. The functions do not have an array on the stack, so they do not need stack protection.
	- /usr/lib64/gawk/readdir.so
	- /usr/lib64/gawk/revtwoway.so
	
	The following library comes from the glib2 package. The library only has a couple functions, none of which need stack protection.
	- /usr/lib64/libgthread-2.0.so.0.5600.4
	
	The following libraries come from the glibc package.
	- glibc has special needs:
	  - /usr/lib64/ld-2.28.so
	  - /usr/lib64/libmvec-2.28.so
	- The following are data tables for character set conversion in glibc:
	  - /usr/lib64/gconv/libCNS.so
	  - /usr/lib64/gconv/libGB.so
	  - /usr/lib64/gconv/libISOIR165.so
	  - /usr/lib64/gconv/libJIS.so
	  - /usr/lib64/gconv/libJISX0213.so

TOE SFRs	Rationale
	- /usr/lib64/gconv/libKSC.so The following libraries come from the iptables package. - The following libraries come from iptables package. The functions are simple and don't need stack protection. - /usr/lib64/xtables/libip6t_ah.so - /usr/lib64/xtables/libip6t_DNPT.so - /usr/lib64/xtables/libip6t_eui64.so - /usr/lib64/xtables/libip6t_frag.so - /usr/lib64/xtables/libip6t_hl.so - /usr/lib64/xtables/libip6t_HL.so - /usr/lib64/xtables/libip6t_ipv6header.so - /usr/lib64/xtables/libip6t_LOG.so - /usr/lib64/xtables/libip6t_REJECT.so - /usr/lib64/xtables/libip6t_rt.so - /usr/lib64/xtables/libip6t_SNAT.so - /usr/lib64/xtables/libip6t_SNPT.so - /usr/lib64/xtables/libipt_ah.so - /usr/lib64/xtables/libipt_CLUSTERIP.so - /usr/lib64/xtables/libipt_ECN.so - /usr/lib64/xtables/libipt_LOG.so - /usr/lib64/xtables/libipt_REJECT.so - /usr/lib64/xtables/libipt_ttl.so - /usr/lib64/xtables/libipt_TTL.so - /usr/lib64/xtables/libipt_ULOG.so - /usr/lib64/xtables/libxt_addrtype.so - /usr/lib64/xtables/libxt_AUDIT.so - /usr/lib64/xtables/libxt_cgroup.so - /usr/lib64/xtables/libxt_CHECKSUM.so - /usr/lib64/xtables/libxt_cluster.so - /usr/lib64/xtables/libxt_connbytes.so - /usr/lib64/xtables/libxt_connlimit.so - /usr/lib64/xtables/libxt_connmark.so - /usr/lib64/xtables/libxt_CONNMARK.so - /usr/lib64/xtables/libxt_CONNSECMARK.so - /usr/lib64/xtables/libxt_cpu.so - /usr/lib64/xtables/libxt_dccp.so - /usr/lib64/xtables/libxt_dscp.so - /usr/lib64/xtables/libxt_DSCP.so - /usr/lib64/xtables/libxt_ecn.so - /usr/lib64/xtables/libxt_esp.so - /usr/lib64/xtables/libxt_helper.so - /usr/lib64/xtables/libxt_HMARK.so - /usr/lib64/xtables/libxt_IDLETIMER.so - /usr/lib64/xtables/libxt_LED.so - /usr/lib64/xtables/libxt_length.so - /usr/lib64/xtables/libxt_mac.so

TOE SFRs	Rationale
	- /usr/lib64/xtables/libxt_mark.so - /usr/lib64/xtables/libxt_multiport.so - /usr/lib64/xtables/libxt_nfacct.so - /usr/lib64/xtables/libxt_NFLOG.so - /usr/lib64/xtables/libxt_NFQUEUE.so - /usr/lib64/xtables/libxt_osf.so - /usr/lib64/xtables/libxt_physdev.so - /usr/lib64/xtables/libxt_pkttype.so - /usr/lib64/xtables/libxt_policy.so - /usr/lib64/xtables/libxt_quota.so - /usr/lib64/xtables/libxt_recent.so - /usr/lib64/xtables/libxt_rpfilter.so - /usr/lib64/xtables/libxt_sctp.so - /usr/lib64/xtables/libxt_SECMARK.so - /usr/lib64/xtables/libxt_socket.so - /usr/lib64/xtables/libxt_standard.so - /usr/lib64/xtables/libxt_statistic.so - /usr/lib64/xtables/libxt_SYNPROXY.so - /usr/lib64/xtables/libxt_tcpmss.so - /usr/lib64/xtables/libxt_TCPMSS.so - /usr/lib64/xtables/libxt_TEE.so - /usr/lib64/xtables/libxt_tos.so - /usr/lib64/xtables/libxt_TOS.so - /usr/lib64/xtables/libxt_TPROXY.so - /usr/lib64/xtables/libxt_TRACE.so - /usr/lib64/xtables/libxt_udp.so - /usr/lib64/libiptc.so.0.0.0 - /usr/lib64/xtables/libebt_arpreply.so - /usr/lib64/xtables/libebt_redirect.so - /usr/lib64/xtables/libebt_dnat.so - /usr/lib64/xtables/libebt_snat.so - /usr/lib64/xtables/libxt_ipcomp.so - /usr/lib64/xtables/libip6t_srh.so - /usr/lib64/xtables/libxt_ipvs.so The following library comes from the libblockdev package. The functions do not have an array on the stack, so they do not need stack protection: - /usr/lib64/libbd_part_err.so.2.0.0 The following libraries come from the ncurses package. The functions do not have an array on the stack, so they do not need stack protection: - /usr/lib64/libpanel.so.6.1 - /usr/lib64/libpanelw.so.6.1 The following library comes from the nspr package. The functions are simple and do not have an array on the stack, so they do not need stack protection: -/usr/lib64/libplc4.so

TOE SFRs	Rationale
	The following libraries come from the pam package. The functions are simple and do not have an array on the stack, so they do not need stack protection: - /usr/lib64/security/pam_deny.so - /usr/lib64/security/pam_postgresok.so The following libraries come from the libldb package. The functions do not have an array on the stack, so they do not need stack protection: - /usr/lib64/ldb/modules/ldb/ldb.so - /usr/lib64/ldb/modules/ldb/mdb.so - /usr/lib64/ldb/modules/ldb/skel.so - /usr/lib64/ldb/modules/ldb/tdb.so The following library comes from the openssl package. The functions do not have an array on the stack, so they do not need stack protection: - /usr/lib64/engines-1.1/capi.so The following library comes from the python3 package. The library has a single simple function, no stack protection is needed: - /usr/lib64/libpython3.so The following are kernel modules that are hand-written assembler - /usr/lib/modules/4.18.0-305.el8.x86_64/vdso/vdso32.so - /usr/lib/modules/4.18.0-305.el8.x86_64/vdso/vdso64.so The following are kernel-uek modules that are hand-written assembler. - /usr/lib/modules/5.4.17-2136.312.3.4.el8uek.x86_64/vdso/vdso32.so - /usr/lib/modules/5.4.17-2136.312.3.4.el8uek.x86_64/vdso/vdso64.so The following libraries come from libgcc package which has special needs. - /usr/lib64/libgcc_s-8-20200928.so.1 The following library comes from the ding-libs package. The functions do not have an array on the stack, so they do not need stack protection: - /usr/lib64/libref_array.so.1.2.1
FPT_TST_EXT.1	When the OS boots, it performs the following operations: The computer's BIOS performs a power-on self-test (POST), and then locates and initializes any peripheral devices including the hard disk. The BIOS reads the Master Boot Record (MBR) into memory from the boot device. (For GUID Partition Table (GPT) disks, this MBR is the protective MBR on the first sector of the disk.) The MBR stores information about the organization of partitions on that device. On a computer with x86 architecture, the MBR occupies the first 512 bytes of the boot device. The first 446 bytes contain boot code that points to the boot loader program, which can be on the same device or on another

TOE SFRs	Rationale
	device. The next 64 bytes contain the partition table. The final two bytes are the boot signature, which is used for error detection. The default boot loader program used on Oracle Linux is GRUB 2, which stands for Grand Unified Bootloader version 2. When Secure Boot is used there are two stages of bootloaders. The first stage bootloader starts and verifies the keys for GRUB2. Once the keys are verified GRUB2 is loaded. The boot loader loads the vmlinuz kernel image file into memory and extracts the contents of the initramfs image file into a temporary, memory-based file system (tmpfs). The kernel loads the driver modules from the initramfs file system that are needed to access the root file system. The kernel starts the systemd process with a process ID of 1 (PID 1). systemd is the ancestor of all processes on a system. systemd reads its configuration from files in the /etc/systemd directory. The /etc/systemd/system.conf file controls how systemd handles system initialization. During this process systemd mounts file systems, saves entropy, and starts system logging, and cron daemons. As a final step, the kernel executes /sbin/init. The OS uses Unified Extensible Firmware Interface (UEFI) Secure Boot technology to ensure the system firmware checks whether the system boot loader is signed with an authorized cryptographic key. The first-stage boot loader, shim.efi, is signed by a UEFI private key and authenticated by a public key, signed by a certificate authority (CA), stored in the firmware database. This boot loader also contains the Oracle public key, which is used to authenticate the GRUB 2 boot loader and the Oracle kernel. The kernel contains public keys to authenticate drivers and modules. Kernel Boot process • The kernel will carry out the following actions as part of the boot process: • Setup functions will be initialized and configure the hardware devices, then the kernel will be loaded into memory function. • Memory management will be initialized. • Kernel mode stack for process 0 is set. • The provisional Page Tables paging will be enabled. • Exception handlers would be set. The kernel will then complete the kernel initialization by initializing Page Tables, Memory Handling Data Structures, the SLUB allocator, system date, and system time.

TOE SFRs	Rationale
	Once the kernel boot process is complete, the user space would be started up. The root file must be available along with the loading of applications and daemons. All other setup and configuration process to get the system operational would be carried out. The software is cryptographically verified (integrity tested) using HMAC-SHA-256. The HMAC value is computed at build time and stored in the hmac file. The value is recalculated at runtime and compared against the stored value. If the comparison succeeds, then the remaining power-up self-test (consisting of the algorithm-specific Known Answer Tests) are performed. On successful completion of the power-up tests, the module becomes operational and crypto services are available. If any of the tests fails module transitions to error state and subsequent calls to the Module will fail - thus no further cryptographic operations will be possible.
FPT_TUD_EXT.1 FPT_TUD_EXT.2	The TOE software is delivered and installed using Red Hat Packages (RPMs). An Oracle certificate is used to verify the RPM during installation of an RPM. The Oracle certificate is installed on the system at the time of installation. The TOE leverages 2048 bit RSA digital signature mechanism for signing and verification of packages/updates**.** SHA-256 used for integrity verification. If the signature verification is successful, then the RPM package is installed. Otherwise, it fails the installation. The administrator must download the RPM from the Oracle download center. To obtain updates, the OS pulls the latest update lists from Oracle servers nightly and either installs new RPMs automatically or informs the administrator about the presence of update RPMs, depending on the system configuration. The installation of these updates follows the signature verification procedure discussed above.
FTP_ITC_EXT.1	The TOE supports TLS v1.2 and SSH v2 for trusted channel implementation. Further details on the implementation of these protocols is provided in FCS_TLSC_EXT.1 and FCS_SSHC_EXT.1.
FTP_TRP.1	TOE supports remote CLI using SSH v2 for secure remote administration. Administration via the local console is also supported. This access is logically distinct from other communication paths and is authenticated by the user prior to access being granted to administrate the OS. Data is protected from modification and disclosure through physical security. Local and remote access to the trusted path is initiated by the user or TSF. No other methods to administer the TOE are available.

Table 18 - TOE Summary Specification SFR Description

ALC_TSU_EXT.1	The information as to how the end-user devices are updated to address security issues in a timely manner is described in section 6.6, table 17 above.

Table 19 - TOE Summary Specification SAR Description

8 Annex A: References

Identifiers	Descriptions
[CC_PART1]	Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-001
[CC_PART2]	Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-002
[CC_PART3]	Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-003
[CEM]	Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated September 2012, version 3.1, Revision 5, CCMB-2017-04-004
[800-38A]	NIST Special Publication 800-38A Recommendation for Block 2001 Edition Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001
[800-56A]	NIST Special Publication 800-56A Rev 2, May 2013
[800-56B]	NIST Special Publication 800-56B Recommendation for Pair-Wise, August 2009
[800-38A]	[NIST Special Publication 800-38A Recommendation for Block 2001 Edition Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001
[800-38D]	NIST Special Publication 800-38D Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007.

Table 20 - Annex A: References

9 Annex B - Extended Security Functional Components

Requirements	Descriptions
FCS_CKM_EXT.4	Cryptographic Key Destruction
FCS_RBG_EXT.1	Random Bit Generation
FCS_STO_EXT.1	Storage of Sensitive Data
FCS_TLSC_EXT.1	TLS Client Protocol
FCS_TLSC_EXT.2	TLS Client Curves Allowed
FCS_SSH_EXT.1	SSH Protocol
FCS_SSHC_EXT.1	SSH Protocol - Client
FCS_SSHS_EXT.1	SSH Protocol - Server
FDP_IFC_EXT.1	Information flow control
FDP_ACF_EXT.1	Access Controls for Protecting User Data
FIA_X509_EXT.1	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FMT_MOF_EXT.1	Management of security functions behavior
FMT_SMF_EXT.1	Specification of Management Functions
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

Table 21 - Extended Security Functional Components

9.1 Cryptographic Support (FCS)

9.1.1 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]*

]

] .

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

NOTE: TD0365 has been applied.

9.1.2 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.

9.1.3 FCS_STO_EXT.1 Storage of Sensitive Data

FCS_STO_EXT.1 The OS shall implement functionality to encrypt sensitive data stored in non-volatile storage and provide interfaces to applications to invoke this functionality.

9.1.4 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: [**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.

9.1.5 FCS_TLSC_EXT.2 TLS Client Protocol

FCS_TLSC_EXT.2.1 The OS shall present the Supported Groups Extension in the Client Hello with the following supported groups: [**selection**: *secp256r1*, *secp384r1*, *secp521r1*].

9.1.6 FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1 FCS_SSH_EXT.1.1 The TOE shall implement SSH acting as a [selection: client, server] in accordance with that complies with RFCs 4251, 4252, 4253, 4254, [selection: 4256, 4344, 5647, 5656, 6187, 6668, 8268, 8308, 8332, 8709, 8731, no other RFCs] and [no other standard].

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [selection: “password” (RFC 4252), “keyboard-interactive” (RFC 4256), “publickey” (RFC 4252): [selection: ssh-rsa (RFC 4253), rsa-sha2-256 (RFC 8332), rsa-sha2-512 (RFC 8332), ecdsa-sha2-nistp256 (RFC

5656), ecdsa-sha2-nistp384 (RFC 5656), ecdsa-sha2-nistp521 (RFC 5656), ssh-ed25519 (RFC 8709), ssh-ed448 (RFC 8709), x509v3-ecdsa-sha2-nistp256 (RFC 6187), x509v3-ecdsa-sha2-nistp384 (RFC 6187), x509v3-ecdsa-sha2-nistp521 (RFC 6187), x509v3-rsa2048-sha256 (RFC 6187)]] and no other methods.

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: number of bytes between 35,000 and 1 GB (inclusive)] in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4 The TSF shall protect data in transit from unauthorised disclosure using the following mechanisms: [selection: aes128-ctr (RFC 4344), aes256-ctr (RFC 4344), aes128-cbc (RFC 4253), aes256-cbc (RFC 4253), AEAD_AES_128_GCM (RFC 5647), AEAD_AES_256_GCM (RFC 5647), aes128-gcm@openssh.com (RFC 5647), aes256-gcm@openssh.com (RFC 5647)] and no other mechanisms.

FCS_SSH_EXT.1.5 The TSF shall protect data in transit from modification, deletion, and insertion using: [selection: hmac-sha2-256 (RFC 6668), hmac-sha2-512 (RFC 6668), AEAD_AES_128_GCM (RFC 5647), AEAD_AES_256_GCM (RFC 5647), implicit] and no other mechanisms.

FCS_SSH_EXT.1.6 The TSF shall establish a shared secret with its peer using: [selection: diffie-hellman-group14-sha256 (RFC 8268), diffie-hellman-group15-sha512 (RFC 8268), diffie-hellman-group16-sha512 (RFC 8268), diffie-hellman-group17-sha512 (RFC 8268), diffie-hellman-group18-sha512 (RFC 8268), ecdh-sha2-nistp256 (RFC 5656), ecdh-sha2-nistp384 (RFC 5656), ecdh-sha2-nistp521 (RFC 5656), curve25519-sha256 (RFC 8731), curve448-sha512 (RFC 8731)] and no other mechanisms.

FCS_SSH_EXT.1.7 The TSF shall use SSH KDF as defined in [selection: RFC 4253 (Section 7.2), RFC 5656 (Section 4)] to derive the following cryptographic keys from a shared secret: session keys.

FCS_SSH_EXT.1.8 The TSF shall ensure that [selection: a rekey of the session keys, connection termination] occurs when any of the following thresholds are met: one hour connection time no more than one gigabyte of transmitted data, or no more than one gigabyte of received data.

9.1.7 FCS_SSHC_EXT.1 SSH Protocol - Client

FCS_SSHC_EXT.1.1 FCS_SSHC_EXT.1.1 The TSF shall authenticate its peer (SSH server) using: [selection: using a local database by associating each host name with a public key corresponding to the following list: [selection: ssh-rsa (RFC 4253), rsa-sha2-256 (RFC 8332), rsa-sha2-512 (RFC 8332), ecdsa-sha2-nistp256 (RFC 5656), ecdsa-sha2-nistp384 (RFC 5656), ecdsa-sha2-nistp521 (RFC 5656), ssh-ed25519 (RFC 8709), ssh-ed448 (RFC 8709)] , a list of trusted certification authorities when the public key is in the following formats: [selection: x509v3-ecdsa-sha2-nistp256 (RFC 6187), x509v3-ecdsa-sha2-nistp384 (RFC 6187), x509v3-ecdsa-sha2-nistp521 (RFC 6187), x509v3-rsa2048-sha256 (RFC 6187)]] as described in RFC 4251 section 4.1.

9.1.8 FCS_SSHS_EXT.1 SSH Protocol - Server

FCS_SSHS_EXT.1.1 FCS_SSHS_EXT.1.1 The TSF shall authenticate itself to its peer (SSH Client) using: [selection: ssh-rsa (RFC 4253), rsa-sha2-256 (RFC 8332), rsa-sha2-512 (RFC 8332), ecdsa-sha2-nistp256 (RFC 5656), ecdsa-sha2-nistp384 (RFC 5656), ecdsa-sha2-nistp521 (RFC 5656), x509v3-ecdsa-sha2-nistp256 (RFC 6187), x509v3-ecdsa-sha2-nistp384 (RFC 6187), x509v3-ecdsa-sha2-nistp521 (RFC 6187), x509v3-rsa2048-sha256 (RFC 6187), ssh-ed25519 (RFC 8709), ssh-ed448 (RFC 8709)].

9.2 User Data Protection (FDP)

9.2.1 FDP_IFC_EXT.1 Information flow control

FDP_IFC_EXT.1.1 The OS shall [**selection**:

- *provide an interface which allows a VPN client to protect all IP traffic using IPsec*,
- *provide a VPN client which can protects all IP traffic using IPsec*

] with the exception of IP traffic required to establish the VPN connection and [**selection**: *signed updates directly from the OS vendor*, *no other traffic*].

9.2.2 FDP_ACF_EXT.1 Access Controls for Protecting User Data

FDP_ACF_EXT.1 The OS shall implement access controls which can prohibit unprivileged users from accessing files and directories owned by other users.

9.3 Identification and Authentication (FIA)

9.3.1 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.

9.3.2 FIA_X509_EXT.2 X.509 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**: other protocols]*, *no other protocols*] connections.

9.4 Security Management (FMT)

9.4.1 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.

9.4.2 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 remote management server from which to receive management settings	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
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

9.5 Protection of the TSF (FPT)

9.5.1 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*]

.

9.5.2 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*].

9.5.3 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].

9.6 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 hardware-protected hash*

] .

9.6.1 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.

FPT_TUD_EXT.1.2 The OS shall [**selection**: cryptographically verify, invoke platform-provided functionality to cryptographically verify] updates to itself using a digital signature prior to installation using schemes specified in FCS_COP.1(3).

NOTE: TD0386 has been applied.

9.6.2 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.

FPT_TUD_EXT.2.2 The OS shall [**selection**: cryptographically verify, invoke platform-provided functionality to cryptographically verify] updates to itself using a digital signature prior to installation using schemes specified in FCS_COP.1(3).

9.7 Trusted Path/Channels (FTP)

9.7.1 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 Function 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.

10 Annex C - Extended Security Assurance Components

10.1 Life Cycle (ALC)

10.1.1 ALC_TSU_EXT.1 Timely Security Updates

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).