ORACLE
Linux

FIPS 140-2 Non-Proprietary Security Policy

Oracle Linux 7 OpenSSH Server Cryptographic Module

FIPS 140-2 Level 1 Validation

Software Version: R7-7.8.0

Date: June 2nd, 2022

Title: Oracle Linux 7 OpenSSH Server Cryptographic Module Security Policy
DATE: June 2nd, 2022
Author: Oracle Security Evaluations – Global Product Security
Contributing Authors:
Oracle Linux Engineering
atsec information security

Oracle Corporation
World Headquarters
2300 Oracle Way
Austin, TX 78741
U.S.A.
Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

ORACLE

TABLE OF CONTENTS

List of Tables

List of Figures

1. Introduction

1.1 Overview

This document is the Security Policy for the Oracle Linux 7 OpenSSH Server Cryptographic Module by Oracle Corporation. Oracle Linux 7 OpenSSH Server Cryptographic Module is also referred to as "the Module" or "Module". This Security Policy specifies the security rules under which the module shall operate to meet the requirements of FIPS 140-2 Level 1. It also describes how the Oracle Linux 7 OpenSSH Server Cryptographic Module functions in order to meet the FIPS 140-2 requirements, and the actions that operators must take to maintain the security of the module.

This Security Policy describes the features and design of the Oracle Linux 7 OpenSSH Server Cryptographic Module using the terminology contained in the FIPS 140-2 specification. FIPS 140-2, Security Requirements for Cryptographic Module specifies the security requirements that will be satisfied by a cryptographic module utilized within a security system protecting sensitive but unclassified information. The NIST/CCCS Cryptographic Module Validation Program (CMVP) validates cryptographic module to FIPS 140-2. Validated products are accepted by the Federal agencies of both the USA and Canada for the protection of sensitive or designated information.

1.2 Document Organization

The FIPS 140-2 submission package contains:

- Oracle Linux 7 OpenSSH Server Cryptographic Module Non-Proprietary Security Policy
- Other supporting documentation as additional references

With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Validation Documentation is proprietary to Oracle and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Oracle.

2. Oracle Linux 7 OpenSSH Server Cryptographic Module

2.1 Functional Overview

The Oracle Linux 7 OpenSSH Server Cryptographic Module is a software module implementing the cryptographic support for the SSH protocol in the Oracle Linux user space.

The Oracle Linux 7 OpenSSH Server Cryptographic Module is distributed with Oracle Linux open-source distributions. The Module implements SSH protocol and acts as a server daemon providing SSH service.

2.2 FIPS 140-2 Validation Scope

The following table shows the security level for each of the eleven sections of the validation.

Security Requirements Section	Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles and Services and Authentication	1
Finite State Machine Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	3
Mitigation of Other Attacks	N/A

Table 1: FIPS 140-2 Security Requirements

3. Cryptographic Module Specification

3.1 Definition of the Cryptographic Module

The Oracle Linux 7 OpenSSH Server Cryptographic Module is a software-only multi-chip standalone module as defined by the requirements within FIPS PUB 140-2. The logical cryptographic boundary of the module consists of the application, library files and their integrity check HMAC files, which are delivered through the Oracle Linux Yum Public Server as listed below:

The module will use the Oracle Linux 7 OpenSSL Cryptographic Module (FIPS 140-2 Certificate #4170) as a bound module which provides the underlying cryptographic algorithms necessary for establishing and maintaining the SSH session. In addition the integrity check uses the cryptographic services provided by the Oracle Linux OpenSSL Cryptographic Module as used by the utility application of fipscheck using the HMAC-SHA-256 algorithm.

This requires a copy of a Cert. #4170 validated version of the Oracle Linux 7 OpenSSL Cryptographic Module to be installed on the system for the current module to operate.

The cryptographic Module combines a vertical stack of Oracle Linux components intended to limit the external interface each separate component may provide. The following software needs to be installed for the module to operate:

- Oracle Linux 7 OpenSSH Server Cryptographic Module with the version of the OpenSSH server RPM file openssh-server-7.4p1-21.0.3.el7.x86_64.rpm or openssh-server-7.4p1-21.0.3.el7.aarch64.rpm
- The bound module of OpenSSL with FIPS 140-2 Certificate #4170
- The contents of the fipscheck RPM package (fipscheck-1.4.1-6.el7.x86_64.rpm or fipscheck-1.4.1-6.el7.aarch64.rpm)
- The contents of the fipscheck-lib RPM package (fipscheck-lib-1.4.1-6.el7.x86_64.rpm or fipscheck-lib-1.4.1-6.el7.aarch64.rpm).

The OpenSSH server RPM package of the Module includes the binary files, integrity check HMAC files and Man Pages. Any application other than the OpenSSH server application delivered with the aforementioned OpenSSH RPM packet is not part of the Module. The FIPS certificate for this module will not be valid if any other application than the OpenSSH server application is used.

The files comprising the module are the following:

- /usr/sbin/sshd
- /usr/bin/fipscheck
- /usr/lib64/fipscheck/sshd.hmac
- /usr/lib64/fipscheck/fipscheck.hmac
- /usr/lib64/fipscheck/libfipscheck.so.1.2.1.hmac
- /usr/lib64/libfipscheck.so.1.2.1

Figure 1 shows the logical block diagram of the module executing in memory on the host system.

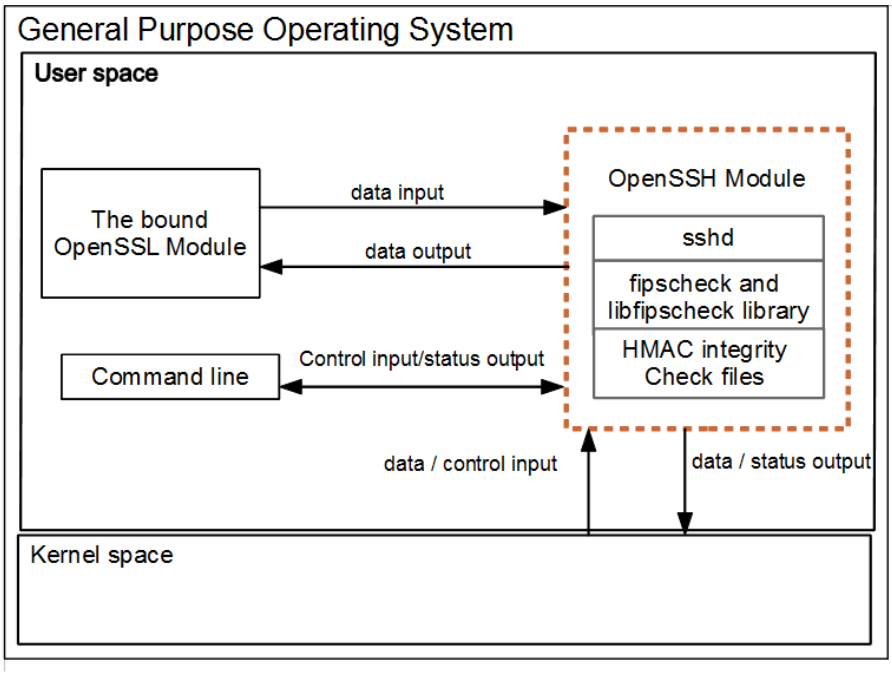

Figure 1: Oracle Linux 7 OpenSSH Server Logical Cryptographic Boundary

3.2 Definition of the Physical Cryptographic Boundary

The physical cryptographic boundary is defined as the hard enclosure of the host system on which it runs. See Figure 2 below. No components are excluded from the requirements of FIPS PUB 140-2.

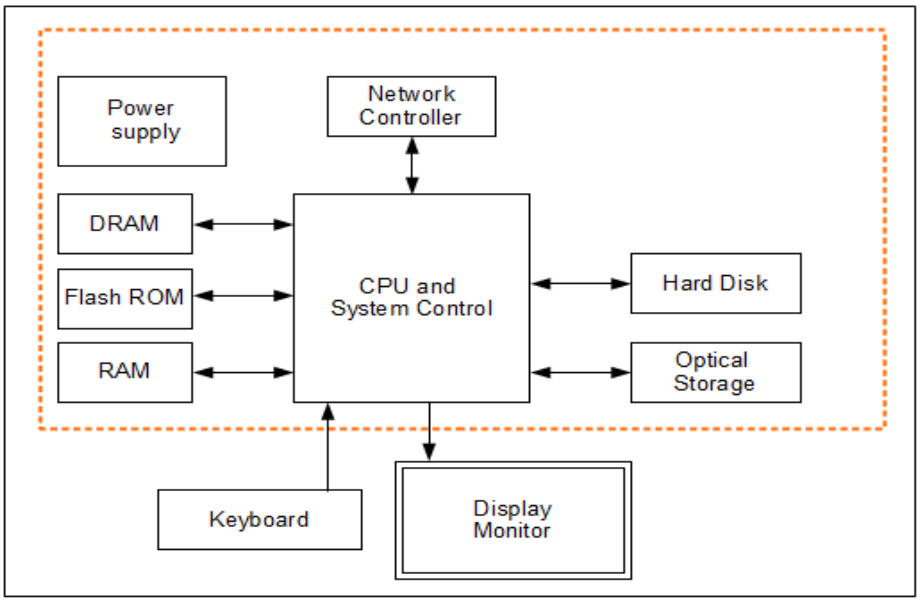

Figure 2: Oracle Linux 7 OpenSSH Server Hardware Block Diagram

3.3 Modes of Operation

The Module supports two modes of operation: FIPS approved and non-FIPS approved mode. The mode of operation is implicitly assumed depending on the services/security functions invoked. The Module turns to the FIPS approved mode after power-on self-tests succeed. The services available in FIPS mode can be found in section 7.2, Table 9.

3.4 Approved Security Functions from OpenSSH module

The Oracle Linux 7 OpenSSH Server Cryptographic Module contains the following FIPS Approved Algorithms:

Approved or Allowed Security Functions		Cert #
Key Derivation Function (NIST SP 800-135)		
SSH v2	AES-ECB (Key Sizes 128, 192, 256); TDES-ECB; (SHA-1, SHA 256, SHA-384, SHA-512);	A 1233

Table 2: FIPS Approved or Allowed Security Function from OpenSSH Module

Note: The SSH protocol except the SP 800-135 Key Derivation Function has not been reviewed tested by the CAVP and CMVP.

The OpenSSH and the bound OpenSSL module together provide the Diffie Hellman and EC Diffie Hellman key agreement compliant with scenario X1 from IG D.8. The OpenSSH module only implements the KDF portion of the key agreement as stated in the above table and the bound OpenSSL module provides the shared secret computation as stated in the below table.

- EC Diffie-Hellman key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength.
- Diffie-Hellman key agreement; key establishment methodology provides between 112 and 200 bits of encryption strength.

3.5 Approved or Allowed Security Functions from the Bound OpenSSL Module

The following table shows Approved or allowed security functions provided by the bound OpenSSL module.

Approved or Allowed Security Functions		Cert #
Symmetric Algorithms		
AES	**AESNI:** AES in CBC, CTR; (E/D; Key Sizes 128, 192, 256)	A 1207
	AESNI_AVX: AES in GCM Mode; External/Internal IV (E; Key Sizes 128, 256)	A 1210
	AESNI_CLMULNI: AES in GCM Mode; External/Internal IV (E; Key Sizes 128, 256)	A 1211
	AESNI_ASM: AES in GCM Mode; External/Internal IV (E; Key Sizes 128, 256)	A 1212
	AESASM: AES in CBC, CTR; (E/D; Key Sizes 128, 192, 256)	A 1208
	AESASM_AVX:	A 1213

	AES in GCM Mode; External/Internal IV (E; Key Sizes 128, 256)	
	AESASM_CLMULNI: AES in GCM Mode; External/Internal IV (E; Key Sizes 128, 256)	A 1214
	AESASM_ASM: AES in GCM Mode; External/Internal IV (E; Key Sizes 128, 256)	A 1215
	BAES_CTASM: AES in CBC, CTR; (E/D; Key Sizes 128, 192, 256)	A 1209
	BAES_CTASM_AVX: AES in GCM Mode; External/Internal IV (E; Key Sizes 128, 256)	A 1216
	BAES_CTASM_CLMULNI: AES in GCM Mode; External/Internal IV (E; Key Sizes 128, 256)	A 1217
	BAES_CTASM_ASM: AES in GCM Mode; External/Internal IV (E; Key Sizes 128, 192, 256)	A 1218
	AES_C: AES in CBC, CTR (E/D; Key Sizes 128, 192, 256)	A 2394
	CE: AES in CBC, CTR (E/D; Key Sizes 128, 192, 256)	A 2396
	VPAES: AES in CBC, CTR (E/D; Key Sizes 128, 192, 256)	A 2397
	AES_C_GCM: AES in GCM Mode; Internal IV (E; Key Sizes 128, 256)	A 2398
	CE_GCM: AES in GCM Mode; Internal IV (E; Key Sizes 128, 256)	A 2399
	VPAES_GCM: AES in GCM Mode; Internal IV (E; Key Sizes 128, 256)	A 2400
Triple-DES	**TDES_C:** Triple-DES in CBC Mode; (KO1, D/E)[1]	A 1206
Secure Hash Standard (SHS)		
SHS	**SHA_AVX2:** SHA-1, SHA-256, SHA-384, SHA-512	A 1224
	SHA_AVX: SHA-1, SHA-256, SHA-384, SHA-512	A 1225
	SHA_SSSE3: SHA-1, SHA-256, SHA-384, SHA-512	A 1226
	SHA_ASM: SHA-1, SHA-256, SHA-384, SHA-512	A 1227
	NEON : SHA-256	A 2395
Data Authentication Code		
HMAC	**SHA_AVX2:** HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512	A 1224
	SHA_AVX:	A 1225

[1] Though the key size is 192 bits, the strength of that key is 112 bits only according to IG 7.5.

	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512	
	SHA_SSSE3: HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512	A 1226
	SHA_ASM: HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512	A 1227
	NEON: HMAC-SHA-256	A 2395
Asymmetric Algorithms		
RSA	**SHA_AVX2:** **FIPS 186-4:** PKCS 1.5 (Sig Gen) Modulus Sizes 2048, 3072, 4096 with hash sizes SHA-256, SHA-512; PKCS 1.5 (Sig Ver) Modulus Sizes 1024, 2048, 3072, 4096 with hash sizes SHA-1, SHA-256, SHA-512;	A 1224
	SHA_AVX: **FIPS 186-4:** PKCS 1.5 (Sig Gen) Modulus Sizes 2048, 3072, 4096 with hash sizes SHA-256, SHA-512; PKCS 1.5 (Sig Ver) Modulus Sizes 1024, 2048, 3072, 4096 with hash sizes SHA-256, SHA-512;	A 1225
	SHA_SSSE3: **FIPS 186-4:** PKCS 1.5 (Sig Gen) Modulus Sizes 2048, 3072, 4096 with hash sizes SHA-256, SHA-512; PKCS 1.5 (Sig Ver) Modulus Sizes 1024, 2048, 3072, 4096 with hash sizes SHA-256, SHA SHA-512;	A 1226
	SHA_ASM: **FIPS 186-4:** PKCS 1.5 (Sig Gen) Modulus Sizes 2048, 3072, 4096 with hash sizes SHA-256, SHA-512; PKCS 1.5 (Sig Ver) Modulus Sizes 1024, 2048, 3072, 4096 with hash sizes SHA-256, SHA-512;	A 1227
ECDSA	**SHA_AVX2:** **FIPS 186-4:** Key Gen, Sig Gen, Sig Ver; Curves P-256, P-384, P-521 with hash sizes SHA-256, SHA-384, SHA-512	A 1224
	SHA_AVX: **FIPS 186-4:** Key Gen, Sig Gen, Sig Ver; Curves P-256, P-384, P-521 with hash sizes SHA-256, SHA-384, SHA-512	A 1225
	SHA_SSSE3: **FIPS 186-4:** Key Gen, Sig Gen, Sig Ver; Curves P-256, P-384, P-521 with hash sizes SHA-256, SHA-384, SHA-512	A 1226
	SHA_ASM: **FIPS 186-4:** Key Gen, Sig Gen, Sig Ver; Curves P-256, P-384, P-521 with hash sizes SHA-256, SHA-384, SHA-512	A 1227
Random Number Generation (NIST SP 800-90A)		
DRBG	**AESNI:** **CTR_DRBG**: [Prediction Resistance Tested: Enabled and Not Enabled; Supports Reseed: (AES-256)]	A 1207
	AESASM: CTR_DRBG: [Prediction Resistance Tested: Enabled and Not Enabled; Supports Reseed: (AES-256)]	A 1208

	BAES_CTASM: **CTR_DRBG:** [Prediction Resistance Tested: Enabled and Not Enabled; Supports Reseed: (AES-256)]	A 1209
	DRBG_10X: **CTR_DRBG**: [Prediction Resistance Tested: Enabled and Not Enabled; Supports Reseed: (AES-256)]	A 1228
	AES_C: **CTR_DRBG**: [Prediction Resistance Tested: Enabled and Not Enabled; Supports Reseed: (AES-256)]	A 2394
	CE: **CTR_DRBG**: [Prediction Resistance Tested: Enabled and Not Enabled; Supports Reseed: (AES-256)]	A 2396
	VPAES: **CTR_DRBG**: [Prediction Resistance Tested: Enabled and Not Enabled; Supports Reseed: (AES-256)]	A 2397
Key Agreement (NIST SP 800-56Ar3)		
KAS-FFC-SSC-SP800-56Ar3	**FFC_DH:** **KAS-FFC-SSC SP 800-56Ar3:** Domain Parameter Generation and Mod P Methods (MODP-2048, MODP-4096, MODP-8192) KAS role: initiator, responder	A 1322
KAS-ECC-SSC-SP800-56Ar3	**SHA_AVX2:** **KAS-ECC-SSC SP 800-56Ar3:** ephemeralUnified scheme for EC Diffie-Hellman shared secret computation (Curves P-256, P-384, P-521). KAS role: initiator, responder	A 1224
	SHA_AVX: **KAS-ECC-SSC SP 800-56Ar3:** ephemeralUnified scheme for EC Diffie-Hellman shared secret computation (Curves P-256, P-384, P-521). KAS role: initiator, responder	A 1225
	SHA_SSSE3: **KAS-ECC-SSC SP 800-56Ar3:** ephemeralUnified scheme for EC Diffie-Hellman shared secret computation (Curves P-256, P-384, P-521). KAS role: initiator, responder	A 1226
	SHA_ASM: **KAS-ECC-SSC SP 800-56Ar3:** ephemeralUnified scheme for EC Diffie-Hellman shared secret computation (Curves P-256, P-384, P-521). KAS role: initiator, responder	A 1227
Safe Primes Key Generation and Verification	**FFC_DH:** Safe Primes Key Generation and Verification: Safe Prime Groups: MODP-2048, MODP-4096, MODP-8192	A 1322
Entropy		
ENT (NP)	NIST SP 800-90B	N/A

Table 3: Approved Security Functions from Bound OpenSSL Module

3.6 Non-Approved Security Functions from OpenSSH Module

The use of following non-Approved services will put the module in non-approved mode of operation implicitly.

Algorithm	Usage
Ed25519	Signature scheme based on Curve 25519

Table 4: Non-Approved Functions From OpenSSH Module

3.7 Non-Approved Security Functions from OpenSSL Module

The use of following non-Approved services will put the module in non-approved mode of operation.

Algorithm	Usage
DSA	Signature generation/verification
RSA with non-compliant key size	Signature generation using keys less than 2048 bits or greater than 4096 bits
	Signature verification using keys less than 1024 bits or greater than 4096 bits

Table 5: Non-Approved Functions From OpenSSL Module

4. Module Ports and Interfaces

As a software-only module, the module does not have physical ports. For the purpose of the FIPS 140-2 validation, the physical ports are interpreted to be the physical ports of the hardware platform on which the module runs.

The module interfaces can be categorized as follows:

- Data Input Interface
- Data Output Interface
- Control Input interface
- Status Output Interface

The table below shows the mapping of ports and interfaces as per FIPS 140-2 Standard.

FIPS 140 Interface	Physical Port	Module Interfaces
Data Input	Keyboard, Ethernet port	Input parameters of the sshd command on the command line with host key files in /etc/ssh, ~/.ssh/authorized_keys, locally stored data, data via SSHv2 channel, input data via local or remote port-forwarding port, input data sent to the bound OpenSSL module via its API parameters.
Data Output	Display, Ethernet Port	Output data returned by the sshd command, output data sent via the SSHv2 channel, output data sent via local or remote port-forwarding port, output data sent to the bound OpenSSL module via its API parameters.
Control Input	Keyboard, Ethernet port	Invocation of the sshd command on the command line or via the configuration file /etc/ssh/sshd_config, SSHv2 protocol message requests received from SSH client
Status Output	Display, Ethernet Port	Status messages returned after the command execution, status of processing SSHv2 protocol message requests
Power Input	PC power supply	N/A

Table 6: Mapping of FIPS 140 Logical Interfaces to Logical Ports

5. Physical Security

The Module is comprised of software only and thus does not claim any physical security.

6. Operational Environment

The module operates in a modifiable operational environment per FIPS 140-2 Security Level 1 specifications. The module runs on a commercially available general-purpose operating system executing on the hardware specified in sections 6.1 and 6.2.

6.1 Tested Environments

The Module was tested on the following environments with and without PAA i.e. AES-NI:

Operating Environment	Processor	Hardware
Oracle Linux 7.8 64 bit	Intel® Xeon® Platinum 8167M	Oracle Server X7-2C
Oracle Linux 7.8 64 bit	AMD EPYC™ 7551	Oracle Server E1-2C
Oracle Linux 7.8 64 bit	Ampere® Altra® Neoverse-N1	Oracle Server A1-2C

Table 7: Tested Operating Environment

6.2 Vendor Affirmed Environments

The following platforms have not been tested as part of the FIPS 140-2 level 1 certification however Oracle "vendor affirms" that these platforms are equivalent to the tested and validated platforms. Additionally, Oracle affirms that the module will function the same way and provide the same security services on any of the systems listed below.

Operating Environment	Hardware
Oracle Linux 7 64-bit	Oracle X Series Servers
Oracle Linux 7 64-bit	Oracle E Series Servers
Oracle Linux 7 64-bit	Oracle A Series Servers
Oracle Linux 7 64-bit	Marvell CN23XX OCTEON (MIPS) SmartNIC
Oracle Linux 7 64-bit	Marvell CN93XX LiquidIO III (ARM) SmartNIC
Oracle Linux 7 64-bit	Pensando DSC-200 (ARM) SmartNIC

Table 8: Vendor Affirmed Operating Environment

Note: Per FIPS 140-2 IG G.5, the Cryptographic Module Validation Program (CMVP) makes no statement as to the correct operation of the module or the security strengths of the generated keys when this module is ported and executed in an operational environment not listed on the validation certificate.

6.3 Operational Environment Policy

The operating system is restricted to a single operator (concurrent operators are explicitly excluded). The entity using the application is the single user of the module. In FIPS Approved mode, the ptrace(2) system call, the debugger (gdb(1)), and strace(1) shall be not used.

7. Roles, Services and Authentication

7.1 Roles

The roles are implicitly assumed by the entity accessing the module services. The Module supports the following roles:

- **User Role**: Performs services to establish, maintain and close SSH session, show status and self-tests.
- **Crypto Officer Role**: Performs module installation and configuration and terminate sshd application.

7.2 FIPS Approved Services and Descriptions

The following table shows the available services, the roles allowed, the Critical Security Parameters (CSPs) involved and how they are accessed in the FIPS mode. In the table below, the "U" represents a User Role, and "CO" denotes a Crypto Officer role.

U	CO	Service Name	Service Description	Keys and CSP(s)	Access
X		Establish SSH Session	SSH authentication	RSA or ECDSA key pair	R, W, X
			Negotiate a SSH V2 key agreement	Diffie-Hellman or EC Diffie-Hellman key pair, shared secret	
			Key derivation using SP800-135 SSH KDF	shared secret, derived session encryption keys (Triple-DES or AES), and derived data authentication (HMAC) keys	
X		Maintain SSH Session	Provide data encryption and data authentication over SSH V2 network protocol	Derived session encryption keys (Triple-DES or AES), and derived data authentication (HMAC) keys	R
X		Close SSH session	Zeroize SSH derived session encryption and data authentication keys by closing the SSH session	Derived session encryption key (Triple-DES or AES) and data authentication keys, shared secret	Z
	X	Terminate sshd application	Zeroize SSH derived session encryption and data authentication keys by terminating the sshd application		
X		Self-Test	Perform on-demand self-tests	None	N/A
X		Show Status	Show status of the module	None	N/A
	X	Installation	Install the SSH Server	None	N/A
	X	Configure SSH Server	Configure the SSH Server	None	N/A

R – Read, W – Write, X – Execute, Z – Zeroize

Table 9: FIPS Approved Services and Descriptions

7.3 Non FIPS Approved Services and Descriptions

The following table shows Non FIPS approved services. Any use of these services will put the module in non-FIPS mode implicitly.

U	CO	Service Name	Service Description	Keys and CSP(s)	Access
X		Establish SSH Session	SSH authentication	DSA, RSA (with key size restrictions listed in Table 5), Ed25519 curve	R, W, X

R – Read, W – Write, X – Execute, Z – Zeroize

Table 10: Non FIPS Approved Services and Descriptions

7.4 Operator Authentication

The module does not support operator authentication mechanisms.

8. Key and CSP Management

The following keys, cryptographic key components and other critical security parameters are contained in the module.

CSP Name	Generation/Input	Use	Zeroization
Shared Secret	N/A (entered via API parameter from the bound OpenSSL module)	Shared secret used to derive session keys.	Zeroized by closing SSH session or terminating the the sshd application
Derived Session Key (AES, Triple-DES[2],HMAC)	Derived from the shared secret via SP800-135 SSH KDF	SSH session keys used for encrypt/decrypt and data authentication operations.	
Server RSA Private Key	N/A (keys are read from the host key file)	RSA server private key used to authenticate SSH server	
Server ECDSA Private Key		ECDSA private key used to authenticate SSH server	
Server EC Diffie-Hellman Private Key	N/A (keys are entered from the bound OpenSSL Module via API parameters)	EC Diffie-Hellman private key used as part of the key agreement protocol.	
Server Diffie-Hellman Private Key		Diffie-Hellman private key used as part of the key agreement protocol.	
Client RSA Public Key	N/A (keys are read from the authorized_keys file)	RSA client public key used as part of the SSH key establishment protocol.	
Client ECDSA Public Key		ECDSA client public key used as part of the SSH key agreement protocol.	
Client EC Diffie-Hellman Public Key	N/A (keys are exchanged during handshake)	EC Diffie-Hellman client public key used as part of the SSH key agreement protocol.	
Client Diffie-Hellman Public Key		Diffie-Hellman client public key used as part of the SSH key agreement protocol.	

Table 11: CSP Table

8.1 Random Number and Key Generation

The module does not implement any random number generator, nor does it provide key generation. The module only provides key derivation through the implementation of the SP 800-135 KDF.

When establishing the SSH Session, the module calls the bound OpenSSL module which generates the shared secret. The module derives keys from this shared secret by applying SP 800-135 KDF. When the module requests encryption/decryption services provided by the OpenSSL bound module, the resulting derived symmetric key (i.e. the output of the SP 800-135 KDF) will be passed to the OpenSSL bound module via API parameters. The module does not support manual key entry.

[2] According to IG A.13, the same Triple-DES key shall not be used to encrypt more than 2^20 64-bit blocks of data.

8.2 Key/CSP Storage

The module does not perform persistent storage of keys. The keys and CSPs are temporarily stored as plaintext in the RAM. The server's public and private keys are stored in the host key files in /etc/ssh directory, which are outside its logical boundary.

8.3 Key/CSP Zeroization

The destruction functions overwrite the memory occupied by keys with zeros and deallocates the memory with the free () call. In case of abnormal termination, or swap in/out of a physical memory page of a process, the keys in physical memory are overwritten before the physical memory is allocated to another process.

9. Self-Tests

9.1 Power-Up Self-Tests

The module performs power-up self-tests at module initialization to ensure that the module is not corrupted. The self-tests are automatically triggered without any user intervention.

While the module is performing the power-up tests, services are not available, and input or output data is not possible: the module is not available for use until the self-tests are completed successfully.

9.1.1 Integrity Tests

The integrity check is performed by the fipscheck application using the HMAC-SHA-256 algorithm implemented by the bound Oracle Linux OpenSSL Cryptographic Module. When the OpenSSH module starts, it triggers the power-on self-tests which includes the software integrity test.

The user space integrity verification is performed as follows: the OpenSSH Server application links with the library libfipscheck.so which is intended to execute fipscheck to verify the integrity of the OpenSSH server application file using the HMAC-SHA-256 algorithm. Upon calling the FIPSCHECK_verify() function provided with libfipscheck.so, fipscheck is loaded and executed, and the following steps are performed:

1. OpenSSL, loaded by fipscheck, performs the integrity check of the OpenSSL library files using the HMAC-SHA-256 algorithm
2. fipscheck performs the integrity check of its application file using the HMAC-SHA-256 algorithm provided by the OpenSSL Module
3. fipscheck automatically verifies the integrity of libfipscheck.so before processing requests of calling applications
4. The fipscheck application performs the integrity check of the OpenSSH server application file. The fipscheck computes the HMAC-SHA-256 checksum of that and compares the computed value with the value stored inside the /usr/lib64/fipscheck/<applicationfilename>.hmac checksum file. The fipscheck application returns the appropriate exit value based on the comparison result: zero if the checksum is OK, an error code otherwise (which brings the OpenSSH Module into the error state). The libfipscheck.so library reports the result to the OpenSSH server application.

If any of those steps fail, an error code is returned and the OpenSSH Module enters the error state with the message 'FIPS integrity verification test failed'. In Error state, all data output is inhibited and no cryptographic operation is allowed. The module needs to be reloaded in order to recover from the Error state.

9.1.2 Cryptographic Algorithm Tests

The OpenSSH module performs an SSH KDF KAT as defined in SP 800-135. The OpenSSH module will use the Oracle Linux 7 OpenSSL Cryptographic Module as a bound module which provides the underlying cryptographic algorithms. All the known answer tests besides the SSH KDF KAT are implemented by the bound OpenSSL Module. If the SSH KDF KAT fails, an error code is returned, and the OpenSSH Module enters the error state with the message 'FIPS POST failed.' Again, in Error state, all data output is inhibited, and no cryptographic operation is allowed. The module needs to be reloaded in order to recover from the Error state.

9.2 On-Demand self-tests

The module provides the Self-Test service to perform self-tests on demand. On demand self-tests can be invoked by powering-off and reloading the module. This service performs the same tests executed during power-up.

During the execution of the on-demand self-tests, crypto services are not available and no data output or input is possible.

10. Crypto-Officer and User Guidance

The following guidance items are to be used for assistance in maintaining the module's validated status while in use.

10.1 Crypto-Officer Guidance

The version of the RPM file containing the FIPS validated Module is stated in section 3.1 above. The Oracle Linux OpenSSL Cryptographic Module referenced in section 3.1 must be installed according to its Security Policy.

The RPM package of the Module can be downloaded from the Oracle Linux 7 "Security Validation (Update 8)" yum repository and installed by standard tools recommended for the installation of Oracle packages on an Oracle Linux system (for example, yum, RPM, and the RHN remote management tool).

To configure the operating environment to support FIPS validated module, perform the following steps:

1. Install OpenSSH-server RPM file e.g for x86_64 use yum command:
   # yum install openssh-server-7.4p1-21.0.3.el7.x86_64.rpm or openssh-server-7.4p1-21.0.3.el7.aarch64.rpm
2. The /etc/ssh/moduli needs to be removed or disabled
3. Install fipscheck RPM file
   # yum install fipscheck-1.4.1-6.el7.x86_64.rpm or fipscheck-1.4.1-6.el7.aarch64.rpm
4. Install the fipscheck-lib RPM file
   # yum install fipscheck-lib-1.4.1-6.el7.x86_64.rpm or fipscheck-lib-1.4.1-6.el7.aarch64.rpm
5. Perform the following steps to configure the boot loader part of the operating environment to support FIPS validated module:

   a) Identify the boot partition and the UUID of the partition. If /boot or /boot/efi resides on a separate partition, the kernel parameter boot=<partition of /boot or /boot/efi> must be supplied. The partition can be identified with the command:

      # df /boot or df /boot/efi

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/sda1	233191	30454	190296	14%	/boot

      # blkid /dev/sda1

   /dev/sda1: UUID="6046308a-75fc-418e-b284-72d8bfad34ba" TYPE="xfs"

   b) As the root user, edit the /etc/default/grub file as follows:

      i. Add the fips=1 option to the boot loader configuration.
         GRUB_CMDLINE_LINUX="vconsole.font=latarcyrheb-sun16
         rd.lvm.lv=ol/swap rd.lvm.lv=ol/root crashkernel=auto
         vconsole.keymap=uk rhgb quiet fips=1"

ii. If the contents of /boot reside on a different partition to the root partition, you must use the boot=UUID=boot_UUID line to the boot loader configuration to specify the device that should be mounted onto /boot when the kernel loads.

```
GRUB_CMDLINE_LINUX="vconsole.font=latarcyrheb-sun16
    rd.lvm.lv=ol/swap rd.lvm.lv=ol/root crashkernel=auto
    vconsole.keymap=uk rhgb quiet
    boot=UUID=6046308a-75fc-418e-b284-72d8bfad34ba fips=1"
```

iii. Save the changes.

This is required for FIPS to perform kernel validation checks, where it verifies the kernel against the provided HMAC file in the /boot directory.

Note:
On systems that are configured to boot with UEFI, /boot/efi is located on a dedicated partition as this is formatted specifically to meet UEFI requirements. This does not automatically mean that /boot is located on a dedicated partition.

Only use the boot= parameter if /boot is located on a dedicated partition. If the parameter is specified incorrectly or points to a non-existent device, the system may not boot.

If the system is no longer able to boot, you can try to modify the kernel boot options in grub to specify an alternate device for the boot=UUID=boot_UUID parameter, or remove the parameter entirely.

6. Rebuild the GRUB configuration as follows:

On BIOS-based systems, run the following command:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

On UEFI-based systems, run the following command:

```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

To ensure proper operation of the in-module integrity verification, prelinking must be disabled on all system files. By default, the prelink package is not installed on the system. However, if it is installed, disable prelinking on all libraries and binaries as follows:

Set PRELINKING=no in the /etc/sysconfig/prelink configuration file.

If the libraries were already prelinked, undo the prelink on all of the system files as follows:

```
# prelink –u –a
```

7. Reboot the system

8. Verify that FIPS module is enabled by running the command:

cat /proc/sys/crypto/fips_enabled

The response should be “1”

The version of the RPM containing the validated Modules is the version listed in Section 3. The integrity of the RPM is automatically verified during the installation of the Modules and the Crypto Officer shall not install the RPM file if the RPM tool indicates an integrity error.

Use care whenever making configuration changes that could potentially prevent access to the /proc/sys/crypto/fips_enabled flag (fips=1) in the file/proc. If the module does not detect this flag during initialization, the module is not setup to operate FIPS compatible mode.

All user space modules depend on this file for running in FIPS compatible mode.

10.1.1 OpenSSH Server Configuration

The user must not use DSA keys for performing SSH authentication as OpenSSH only allows DSA keys with 1024 bit size which are disallowed as per SP800-131A.

The user must not accept DSA host keys potentially offered during the first contact of an SSH server as OpenSSH only allows DSA keys with 1024 bit size which are disallowed as per SP800-131A.

When re-generating RSA host keys, the crypto officer should generate RSA keys with a size of 2048 bit or higher according to [SP800-131A]. The crypto officer should inform the user base to not use RSA keys with key sizes smaller than 2048 bits.

With operating environment setup as stated in the above section, the following restrictions are applicable. For the module, the mode of operation is implicitly assumed depending on the services/security functions invoked as stated in section 3.3 and the successive sections lists the available ciphers from the module. Any use of non-approved cipher or non-Approved key size will result in the module entering the non-FIPS mode of operation. No more cipher addition is possible by configuration or command line options.

- SSH protocol version 1 is not allowed
- GSSAPI is not allowed
- Only the following ciphers are allowed:
  - aes128-ctr
  - aes192-ctr
  - aes256-ctr
  - aes128-cbc
  - aes192-cbc
  - aes256-cbc
  - aes128-gcm@openssh.com
  - aes256-gcm@openssh.com
  - 3des-cbc
  - rijndael-cbc@lysator.liu.se

Only the following message authentication codes are allowed:

- hmac-sha1
- hmac-sha2-256
- hmac-sha2-512
- hmac-sha1- etm@openssh.com
- hmac-sha2-256- etm@openssh.com
- hmac-sha2-512- etm@openssh.com

10.2 User Guidance

Use the 'systemctl start sshd' command to start the OpenSSH server, or configure the server to start using 'systemctl enable/disable'. This module is used by connecting to it with an SSH client. See the documentation of the client, e.g. the Oracle Linux 7 OpenSSH Client Cryptographic Module's Security Policy and the sshd(1) man page, for more information.

10.2.1 Handling Self-Test Errors

The OpenSSH self-test consists of the software integrity test and SSH KDF KAT. If the integrity test fails, OpenSSH enters an error state with the message 'FIPS integrity verification test failed'. If the SSH KDF KAT fails, the error code is returned, and the OpenSSH Module enters the error state with the message 'FIPS POST failed'. To recover from the error state, the module must be restarted. If the failure persists, the module must be reinstalled. The bound OpenSSL module's self tests failures will prevent OpenSSH from operating. See the Guidance section in the OpenSSL Security Policy for instructions on handling OpenSSL self test failures.

10.2.2 AES-GCM

IV generation for AES-GCM only occurs in the context of the SSHv2 protocol. The module is compliant with RFC 4252, 4253 and 5647.

When an SSH session gets terminated for any reason, all keying material will be re-negotiated by the module.

The module enforces a maximum of 2^{31} packets that can be either sent or received by the module for a given SSH session, which satisfied the 2^{64}-1 AES-GCM encryption limit imposed by IG A.5.

11. Mitigation of Other Attacks

The Oracle Linux 7 OpenSSH Server cryptographic module does not mitigate against attacks.

Acronyms, Terms and Abbreviations

Term	Definition
AES	Advanced Encryption Standard
CCCS	Canadian Centre for Cyber Security
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
DRBG	Deterministic Random Bit Generator
ECDSA	Elliptic Curve Digital Signature Algorithm
HMAC	(Keyed) Hash Message Authentication Code
KAS	Key Agreement Scheme
KAT	Known Answer Test
KDF	Key Derivation Function
NIST	National Institute of Standards and Technology
PAA	Processor Algorithm Acceleration
PUB	Publication
RFC	Request for Comment documents are technical specification and organizational notes for the internet
SHA	Secure Hash Algorithm
SSH	Secure Shell

Table 12: Acronyms

References

The FIPS 140-2 standard, and information on the CMVP, can be found at http://csrc.nist.gov/groups/STM/cmvp/index.html. More information describing the module can be found on the Oracle web site at https://www.oracle.com/linux/ .

This Security Policy contains non-proprietary information. All other documentation submitted for FIPS 140-2 conformance testing and validation is "Oracle - Proprietary" and is releasable only under appropriate non-disclosure agreements.

Document	Author	Title
FIPS PUB 140-2	NIST	FIPS PUB 140-2: Security Requirements for Cryptographic Modules
FIPS IG	NIST	Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program
NIST SP 800-135	NIST	Recommendation for Existing Application-Specific Key Derivation Functions
NIST SP 800-131A	NIST	Recommendation for the Transitioning of Cryptographic Algorithms and Key Sizes
NIST SP 800-56Arev3	NIST	Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography
NIST SP 800-90A	NIST	Recommendation for Random Number Generation Using Deterministic Random Bit Generators
NIST SP 800-90B	NIST	Recommendation for the Entropy Sources Used for Random Bit Generation

Table 13: References