SUSE Linux Enterprise Server 12 - OpenSSH Client Module v1.0 and v2.0

FIPS 140-2 Non-Proprietary Security Policy

Version 2.0

Last Update: 2017-12-21

Prepared by:
atsec information security corporation
9130 Jollyville Road, Suite 260
Austin, TX 78759
www.atsec.com

Contents

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

1. Cryptographic Module Specification

This document is the non-proprietary security policy for the SUSE Linux Enterprise Server 12 - OpenSSH Client Module, and was prepared as part of the requirements for conformance to the Federal Information Processing Standard (FIPS) 140-2, Security Level 1.

1.1. Description of Module

The SUSE Linux Enterprise Server 12 - OpenSSH Client Module (hereafter referred to as “the Module” or “the OpenSSH Client Module”) is a software only which supplies cryptographic support for the client-side of the SSH protocol in the SUSE Linux Enterprise Server (SLES) user space. The current version of the Module is 1.0.

Configuration and installation of the Module for v1.0 requires the following RPM packages:

- the openssh-6.6p1-22.1.x86_64.rpm which includes the binary code of the OpenSSH client application;
- the openssh-fips-6.6p1-22.1.rpm which includes the HMAC integrity verification file;
- the dracut-fips-037-37.2.x86_64.rpm which provides the configuration of the FIPS mode.

Configuration and installation of the Module for v2.0 requires the following RPM packages:

- the openssh-7.2p2-74.11.1.x86_64.rpm and openssh-7.2p2-74.11.1.s390x.rpm which includes the binary code of the OpenSSH client application;
- the openssh-fips-7.2p2-74.11.1.x86_64.rpm and openssh-fips-7.2p2-74.11.1.s390x.rpm which includes the HMAC integrity verification file;

For FIPS 140-2 purposes, the Module is classified as a multi-chip standalone module. The HMAC checksum file is used for integrity check of the Module. The dracut-fips RPM package is only used for the configuration of the Module in every boot, which is not active when the Module is operational and does not provide any services to

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

users interacting with the Module.

The SUSE Linux Enterprise Server 12 - OpenSSH Client Module uses the SUSE Linux Enterprise Server 12 - OpenSSL Module v2.0 and v3.0 (hereafter referred to as “the OpenSSL module”) with FIPS 140-2 Validation #2435 (v2.0) and #3038 (v3.0) for standard cryptographic services, and requires that a copy of the OpenSSL module be installed on the system for the OpenSSH Client Module to operate in a validated mode.

The following table shows the security level claimed for each of the eleven sections of the FIPS 140-2 validation.

Security Component	FIPS 140-2 Security Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A

Table 1. Security Level of the Module

The Module has been tested in the following configurations:

- 64-bit x86_64 with AES-NI
- 64-bit x86_64 without AES-NI

The Module has been tested on the following multi-chip standalone platforms:

Module Version	Platform	Processor	Test Configuration
1.0	HP Proliant DL320e Gen8	X86_64	SUSE Linux Enterprise Server 12
2.0	FUJITSU Server PRIMERGY CX2570 M2 inside a CX400 M1 enclosure	Intel Xeon E5 family	SUSE Linux Enterprise Server 12 SP2
2.0	IBM z13	z13	SUSE Linux Enterprise Server 12 SP2

Table 2. Tested Platforms

To operate the Module, the operating system must be restricted to a single operator mode of operation.

1.2. Description of Approved Mode

The Module supports two modes of operation: FIPS Approved mode and non-Approved mode.

When the Module is powered on, it will execute power-up self-tests automatically. After it passes the self-tests, the Module will check the content of the file '/proc/sys/crypto/fips_enabled'. If the file exists and the content is '1' then the Module will be in FIPS Approved mode; otherwise, it will be in non-Approved mode.

In FIPS Approved mode, the OpenSSH Client Module will require the following Approved cryptographic algorithms from the OpenSSL module.

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	Usage	Keys/CSPs	CAVS Certs. for OpenSSL #2435	CAVS Cert. OpenSSL #3038
AES (CBC, CTR and GCM)	Encryption and Decryption	AES keys 128 bits, 192 bits and 256 bits	Certs. #3197, #3198 and #3199	Certs. #4588, #4594, #4595, #4622, #4623, #4645, #4646, and #4647
Triple-DES (CBC)	Encryption and Decryption	Triple-DES keys 168 bits	Cert. #1823	Certs. #2439 and #2455
RSA	Signature Generation and Verification	RSA keys 2048 bits and 3072 bits	Cert. #1628	Certs. #2505 and #2519
ECDSA	Signature Generation and Verification	ECDSA keys P-256, P-384 and P-521	Cert. #586	Certs. #1127 and #1131
SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	Message Digest	N/A	Certs. #2645, #2646, and #2648	Certs. #3768, #3769, #3770, #3771, #3788 and #3789
HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512	Message Authentication Code	At least 112 bits HMAC key	Certs. #2014, #2015 and #2016	Certs. #3042, #3043, #3044, #3045, #3059 and #3060
SP 800-90A DRBG	Random Number Generation	Seed and nonce	Certs. #674, #675 and #676	Certs. #1531, #1535, #1536, #1537, #1538, #1539, #1540, #1552, #1553
SP800-56A DLC primitive Diffie-Hellman	Key Agreement	Diffie-Hellman public and private components with key size 2048 and 3072 bits	CVL Cert. #431	Certs. #1263 and #1276
SP800-56A DLC primitive EC Diffie-Hellman	Key Agreement	EC Diffie-Hellman public and private components with P-256, P-384 and P-521	CVL Cert. #431	Certs. #1263 and #1276

Table 3. Approved Algorithms provided by the bound OpenSSL Module

The SUSE Linux Enterprise Server 12 - OpenSSH Client Module itself implements the client-side SSH protocol and the following cryptographic algorithm:

Algorithm	Usage	Keys/CSPs	CAVS Cert. (V1.0)	CAVS Cert. (v2.0) X86	CAVS Cert. (v2.0) z13
SP800-135 Key Derivation in SSH	Key Derivation in SSH	Session encryption and data authentication keys	CVL Cert. #483	CVL Certs. #1492	CVL Certs. #1493

Table 4. Approved Algorithms provided by the OpenSSH Client Module

The Module should be used with SSHv2 protocol only, as SSHv1 protocol is not supported in FIPS Approved mode.

In non-Approved mode, the OpenSSH Client Module will also require the following non-Approved services from the OpenSSL module.

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm	Usage
DES	Encryption and Decryption
Blowfish	Encryption and Decryption
CAST	Encryption and Decryption
RC4	Encryption and Decryption
MD5	Message Digest
RIPEMD160	Message Digest
DSA	Signature Generation with key size of 1024 bits
RSA	Signature Generation with key size smaller than 2048 bits or greater than 3072 bits
RSA	Signature Verification with key size smaller than 1024 bits and greater than 3072 bits

Table 5. Non-Approved Algorithms provided by the bound OpenSSL Module

The OpenSSH Client Module implements the following non-Approved algorithm.

Algorithm	Usage
ChaCha20	Encryption and Decryption
Poly1305	Message Authentication Code
UMAC	Message Authentication Code
Curve25519-based ECDH	EC Diffie-Hellman Key Agreement using Curve25519
Ed25519	Signature Generation and Verification based on Curve25519

Table 6. Non-Approved Algorithms provided by the OpenSSH Client Module

1.3. Cryptographic Module Boundary

The physical module boundary is the surface of the case of the test platform. The logical module boundary is depicted in the software block diagram and is embodied by:

- the OpenSSH client application found at /usr/bin/ssh;
- the HMAC integrity verification file found at /usr/bin/ssh.hmac.

The OpenSSL module is a shared library to which the OpenSSH Client Module is bound. The OpenSSL module provides the standard cryptographic services to the OpenSSH Client Module.

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

1.3.1. Hardware Block Diagram

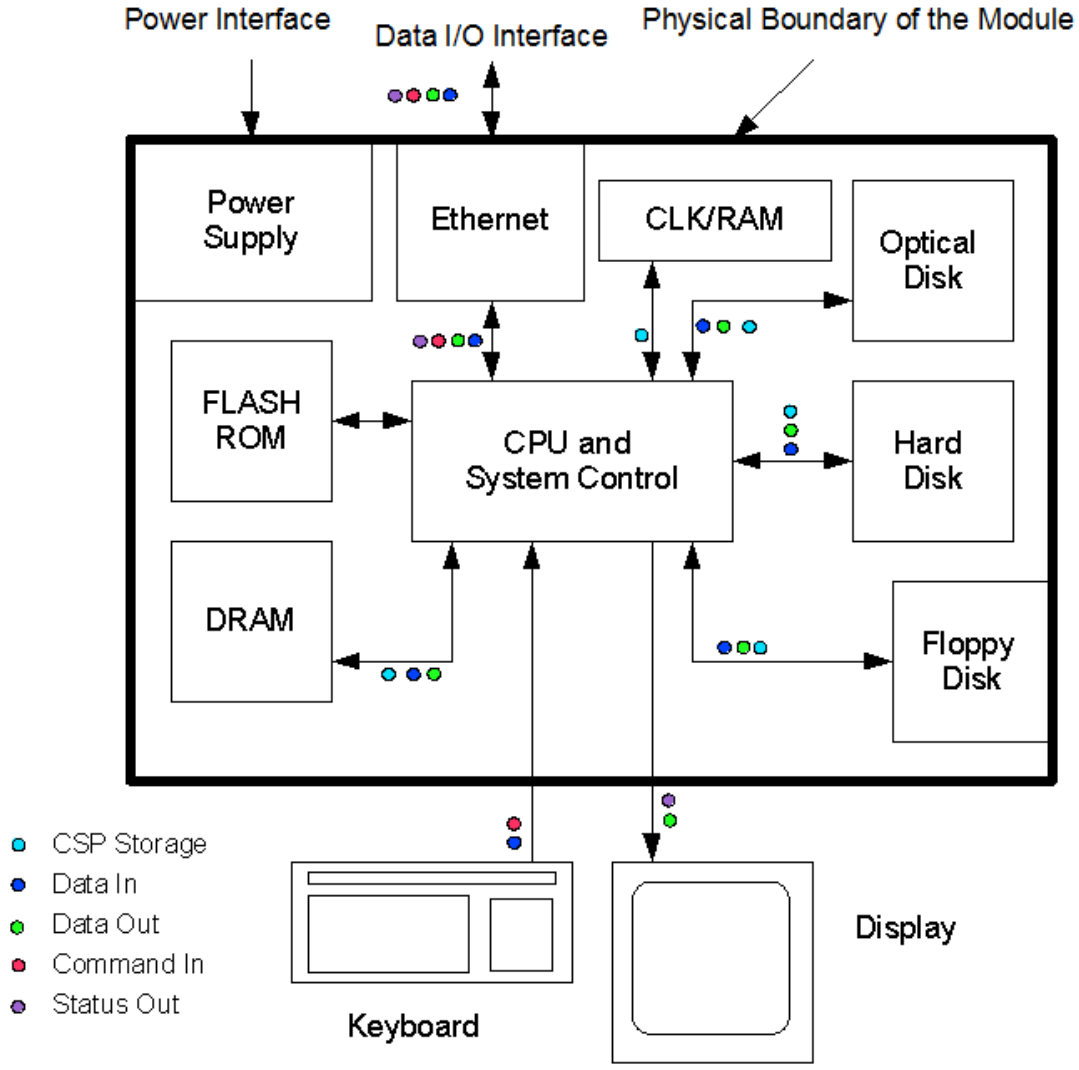

Figure 1. Hardware Block Diagram

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

1.3.2. Software Block Diagram

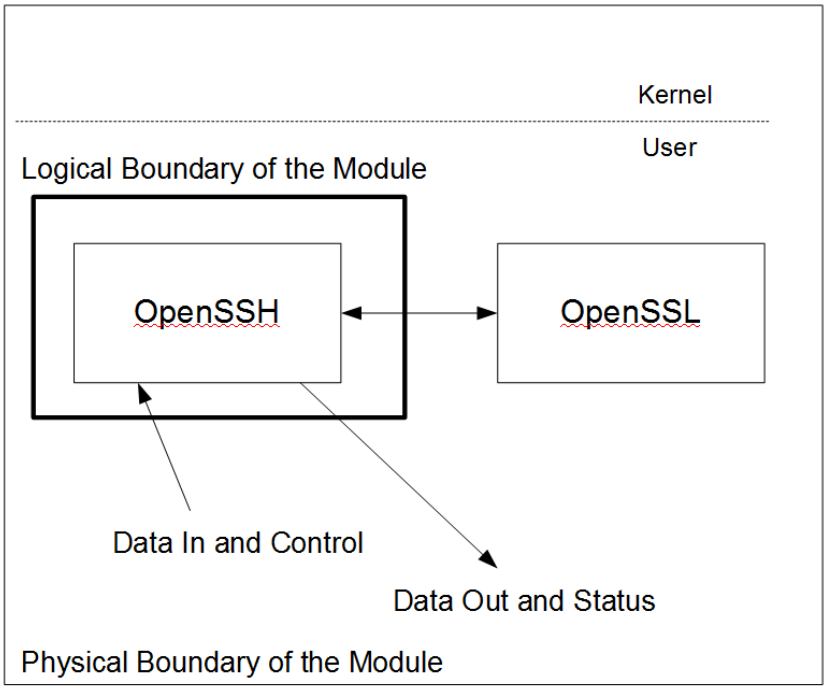

Figure 2. Software Block Diagram

1.4. SUSE Linux Cryptographic Modules and FIPS 140-2 Validation

1.4.1. FIPS Approved Mode

The FIPS Approved mode ensures that FIPS required self tests are executed and that ciphers are restricted to those that have been FIPS validated by the CMVP.

The FIPS Approved mode for a Module becomes effective as soon as the Module power on self tests complete successfully and the Module loads into memory.

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

2. Cryptographic Module Ports and Interfaces

As a software-only module, the logical interfaces are of the most importance. For the purpose of the FIPS 140-2 validation, the logical interfaces of the software-only module can be mapped to the external physical ports of the hardware platform on which it runs as shown in the table below.

Module Logical Interface	Description	Physical ports of the platform on which the module runs
Data Input	Input parameters of the ssh command on the command line with configuration file ~/.ssh/known_hosts, /etc/ssh/ssh_known_hosts, key files ~/.ssh/id_ecdsa*, ~/.ssh/id_rsa*, data via SSHv2 channel, data via local or remote portforwarding port, data via TUN device	Keyboard, Ethernet port
Data Output	Output data returned by the ssh command	Display, Ethernet port
Control Input	Invocation of the ssh command on the command line or via the configuration file ~/.ssh/config, /proc/sys/crypto/fips_enabled, environment variable SSH_USE_STRONG_RNG	Keyboard, Ethernet port
Status Output	Status messages returned after the command execution	Display, Ethernet port
Power Input	Physical power connector	PC power supply port

Table 7. Ports and Interfaces

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

3. Roles, Services, and Authentication

This section defines the roles, services, and authentication mechanisms and methods with respect to the applicable FIPS 140-2 requirements.

3.1. Roles

Role	Services (see list below)
User	Establish & Maintain SSH Session Close SSH Session (Zeroize) Terminate SSH Application (Zeroize) Self Tests Show Status
Crypto Officer	Installation Configure SSH Client

Table 8. Roles

The User and Crypto Officer roles are implicitly assumed by the entity accessing services provided by the Module.

3.2. Services

The Module supports services that are available to users in the various roles. All of the services are described in detail in the module's user documentation.

The following table lists the services available in FIPS Approved mode to the various roles, the cryptographic algorithms involved, and the access to the Approved cryptographic keys and CSPs resulting from services (please refer to Table 3 and 4 for Approved key sizes of the Approved algorithms and the algorithm certificates).

R – The item is read or referenced by the service.
W – The item is written or updated by the service.
Z – The item is zeroized by the service.

Service	Function	Algorithm	Cryptographic Keys and CSPs Accessed	Access Type (RWZ)	Role
Establish & Maintain SSH Session	Encryption and Decryption	AES, Triple-DES (provided by the bound OpenSSL module)	AES or Triple-DES session encryption key	RW	User
	Signature Generation and Verification	RSA, ECDSA (provided by the bound OpenSSL module)	Client's RSA or ECDSA private key Client's RSA or ECDSA public key Server's RSA or ECDSA public key	RWZ, RW	
	Message Digest	SHS (provided by the bound OpenSSL module)	N/A	N/A	
	Message Authentication Code	HMAC (provided by the bound OpenSSL module)	HMAC data authentication key	RW	
	Random Number Generation	SP800-90A DRBG (provided by the bound OpenSSL module)	DRBG seed and nonce	RW	

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

Service	Function	Algorithm	Cryptographic Keys and CSPs Accessed	Access Type (RWZ)	Role
	Key Agreement	Diffie-Hellman, EC Diffie-Hellman (provided by the bound OpenSSL module)	Diffie-Hellman private and public components, EC Diffie-Hellman private and public components	RW	
	Key Derivation	SP800-135 Key Derivation in SSH (provided by the OpenSSH Client Module)	Session encryption and data authentication keys	RW	
Close SSH Session	Zeroize	None	AES or Triple-DES session encryption key, HMAC data authentication key, DRBG seed and nonce, Diffie-Hellman private and public components, EC Diffie-Hellman private and public components	Z	User
Terminate SSH Application	Zeroize	None	AES or Triple-DES session encryption key, HMAC data authentication key, DRBG seed and nonce, Diffie-Hellman private and public components, EC Diffie-Hellman private and public components	Z	User
Self-Tests	Integrity test of the OpenSSH Client Module Invoked by restarting it	HMAC (provided by the bound OpenSSL module)	HMAC-SHA-256 key	R	User
Show Status	Show status via verbose mode and exit codes	None	None	N/A	User
Installation	None	None	None	N/A	Crypto Officer
Configure SSH Client	None	None	None	N/A	Crypto Officer

Table 9. Services available in FIPS Approved mode

The following table lists the services available in non-Approved mode (please refer to Table 5 and 6 for non-Approved key sizes and usage of the non-Approved algorithms). The module enforces the separation of keys or CSPs between the Approved mode and non-Approved mode.

Service	Function	Algorithm	Role
Establish & Maintain SSH	Encryption and Decryption	AES, Triple-DES, DES, Blowfish, CAST, RC4 (provided by the bound OpenSSL module)	User

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

Service	Function	Algorithm	Role
Session		ChaCha20 (provided by the OpenSSH Client Module)	
	Signature Generation and Verification	RSA, ECDSA, DSA (provided by the bound OpenSSL module; please refer to Table 5 for non-Approved key sizes) Ed25519 (provided by the OpenSSH Client Module)	
	Message Digest	SHS, MD5, RIPEMD160 (provided by the bound OpenSSL module)	
	Message Authentication Code	HMAC (provided by the bound OpenSSL module) Poly1305, UMAC (provided by the OpenSSH Client Module)	
	Random Number Generation	SP800-90A DRBG (provided by the bound OpenSSL module)	
	Key Agreement	Diffie-Hellman, EC Diffie-Hellman (provided by the bound OpenSSL module) Curve25519-based ECDH (provided by the OpenSSH Client Module)	
	Key Derivation	SP800-135 Key Derivation in SSH (provided by the OpenSSH Client Module)	
Close SSH Session	Zeroize	None	User
Terminate SSH Application	Zeroize	None	User
Self-Tests	Integrity test of the OpenSSH Client Module invoked by restarting it	HMAC (provided by the bound OpenSSL module)	User
Show Status	Show status via verbose mode and exit codes	None	User
Installation	None	None	Crypto Officer
Configure SSH Client	None	None	Crypto Officer

Table 9A. Services available in non-Approved mode

3.2.1. Operator Authentication

At security level 1, authentication is neither required nor employed. The role is implicitly assumed on entry.

3.3. Mechanism and Strength of Authentication

At security level 1, authentication is not required.

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

4. Physical Security

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

5. Operational Environment

This Module will operate in a modifiable operational environment per the FIPS 140-2 definition.

5.1. Policies

The operating system shall be restricted to a single operator mode of operation (i.e., concurrent operators are explicitly excluded).

The operator that makes use of the cryptographic module is the single user, even when the application is serving multiple clients.

In the FIPS approved mode, the ptrace(2) system call, the debugger (gdb(1)) and strace(1) shall not be used. In addition, other tracing mechanisms offered by the Linux environment, such as ftrace or systemtap shall not be used.

6. Cryptographic Key Management

The following table identifies the cryptographic keys and Critical Security Parameters (CSPs) used within the Module. Cryptographic keys and CSPs are never output from the Module in plaintext. An Approved key generation method is used to generate keys and CSPs.

6.1. Key Life Cycle Table

Key	Type	Generation	Establishment	Access by Service	Entry and Output Method	Storage	Zeroization
Client Private Keys	RSA or ECDSA keys	N/A	N/A	Establish & Maintain SSH Session	N/A	Persistently stored in plaintext; temporarily stored in volatile RAM	Immediately after each use
Client Public Keys (not a CSP)	RSA or ECDSA keys	N/A	N/A	Establish & Maintain SSH Session	Exported	Persistently stored in plaintext; temporarily stored in volatile RAM	N/A
Server Public Keys (not a CSP)	RSA or ECDSA keys	N/A	N/A	Establish & Maintain SSH Session	Imported	Persistently stored in plaintext; temporarily stored in volatile RAM	N/A
Session Data Authentication Keys	HMAC-SHA-1, HMAC-SHA-256, or HMAC-SHA-512	N/A	Established during the SSH handshake through Diffie-Hellman/EC Diffie-Hellman and SP 800-135 SSH KDF	Establish & Maintain SSH Session	N/A	Temporarily stored in volatile RAM	Close SSH Session or Terminate SSH Application
Session Encryption Keys	AES or Triple-DES	N/A	Established during the SSH handshake through Diffie-Hellman/EC Diffie-Hellman and SP 800-135 SSH KDF	Establish & Maintain SSH Session	N/A	Temporarily stored in volatile RAM	Close SSH Session or Terminate SSH Application

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

Key	Type	Generation	Establishment	Access by Service	Entry and Output Method	Storage	Zeroization
Software Integrity Key	HMAC-SHA-256	N/A	N/A	Self-Test	N/A	Stored in plaintext as part of the module	Close SSH Session or Terminate SSH Application
Diffie-Hellman private and public components	Diffie-Hellman	SP 800-90A DRBG provided by the bound OpenSSL module	N/A	Establish & Maintain SSH Session	N/A	Temporarily stored in volatile RAM	Close SSH Session or Terminate SSH Application
EC Diffie-Hellman private and public components	EC Diffie-Hellman	SP 800-90A DRBG provided by the bound OpenSSL module	N/A	Establish & Maintain SSH Session	N/A	Temporarily stored in volatile RAM	Close SSH Session or Terminate SSH Application
DRBG seed	SP 800-90A DRBG	N/A	N/A	Establish & Maintain SSH Session	N/A (provided by /dev/urandom)	Temporarily stored in volatile RAM	N/A (Termination of the SSH application where OpenSSL zeroizes seed)
DRBG nonce	SP 800-90A DRBG	N/A	N/A	Establish & Maintain SSH Session	N/A (provided by /dev/urandom)	Temporarily stored in volatile RAM	N/A (Termination of the SSH application where OpenSSL zeroizes nonce)

Table 10. Key Life Cycle

Notes:

The Module ships without containing any keys and CSPs. When the Module is configured, the Crypto Officer generates a ECDSA or RSA private/public key pair that is stored in plaintext form in keystore files in the filesystem.

A Crypto Officer or User adds server's public keys to the ~/.ssh/known_hosts file using the ssh-keyscan utility or manually adding the public keys. If the ssh_keyscan utility is used, the Crypto Officer or User must verify the keys are correct to prevent a man-in-the-middle attack.

The server's public key is associated with the correct entity as each key is associated with its relevant hostname, or IPv4 address as a lookup index. Moreover, using an incorrect public key results in failure to establish a valid session as the corresponding private key cannot correctly authenticate against an incorrect public key.

The only key management operations during initial configuration include generating the client's public-private key pair and storing client's public keys in the ~/.ssh/, which is out of scope for this validation.

The client's public key is only sent to the server and the Module does not use it for cryptographic purposes.

Diffie-Hellman or EC Diffie-Hellman key agreement is invoked at the beginning of a session as well as after each 1 GB of data transfer or 1 hour of operation, whichever occurs first.

Persistently stored secret and private keys are out of scope, but may be zeroized using a FIPS140-2 approved mechanism to clear data on hard disks.

6.2. Key Zeroization

For volatile memory, memset is included in deallocation operations. There are no restriction when zeroizing any cryptographic keys and CSPs. The OpenSSH Client Module calls the appropriate destruction functions from the OpenSSL module API.

6.3. Random Number Generation

The Module uses a FIPS-Approved, SP 800-90A compliant Deterministic Random Bit Generator (known as SP

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

800-90A DRBG). It is provided by the SUSE Linux Enterprise Server 12 - OpenSSL Module v2.0 and v3.0, to which this OpenSSH Client Module is bound. It is seeded by the Linux kernel.

The Linux kernel offers /dev/urandom as a source of random numbers for seeding the DRBG. The Linux kernel initializes these pseudo device at system startup.

When the environment variable of SSH_USE_STRONG_RNG is used, the seed source of /dev/random is selected. This environment variable must have a positive integer greater than or equal to 6 to be honored. That integer value specifies the number of bytes obtained from /dev/random and mixed into the DRBG state via the OpenSSL RNG RAND_add API call.

For the details of SP 800-90A DRBG implementation in the OpenSSL module and its seed sources, please refer to the SUSE Linux Enterprise Server 12 - OpenSSL Module v2.0 FIPS 140-2 Security Policy, Section 6.1, “Random Number Generation.”

The OpenSSL module performs Continuous Random Number Generation Test (CRNGT) on the output of the SP 800-90A DRBG to ensure that consecutive random numbers do not repeat. The CRNGT on the random numbers for seeding the DRBG is performed by the kernel.

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

7. Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

The test platform that runs the Module meets the requirements of 47 CFR FCC PART 15, Subpart B, Class A (Business use)

8. Self Tests

FIPS 140-2 requires that the Module perform self tests to ensure the integrity of the module and the correctness of the cryptographic functionality at startup. In addition, some functions require continuous verification of function, such as the random number generator. All of these tests are listed and described in this section.

8.1. Power-Up Tests

No operator intervention is required during the running of the self tests. Input, output and cryptographic functions cannot be performed while the Module is in a self-test or error state. If the power-up self-tests fail, subsequent calls to the Module will also fail – thus no further cryptographic operation is possible.

See section 9.3 for descriptions of possible self test errors and recovery procedures.

The power-up and conditional self tests of the cryptographic algorithms are entirely performed by the bound OpenSSL module.

8.1.1. Software Integrity Test Details

The OpenSSH Client Module checks its integrity automatically at startup. It uses the HMAC-SHA-256 algorithm (provided by the OpenSSL module) to calculate the HMAC value of the OpenSSH client application binary file and then compares the value with the value stored in the HMAC integrity verification file. If the two values match, the module passes the integrity test. Otherwise, it fails the integrity test.

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

9. Guidance

NOTE: The Module requires that a copy of a FIPS 140-2 validated version of the SUSE Linux Enterprise Server 12 - OpenSSL Module v2.0 or v3.0 be installed on the same operational environment.

9.1. Crypto Officer Guidance

The version of the RPMs containing the validated module is stated in section 1, above. The integrity of the RPM is automatically verified during the installation and the crypto officer shall not install the RPM file if the RPM tool indicates an integrity error. In addition, the OpenSSL module referenced in section 1 must be installed according to its Security Policy.

The RPM package of the Module can be installed by standard tools recommended for the installation of RPM packages on a SUSE Linux system.

To bring the module into FIPS Approved mode, perform the following:

1. Install the dracut-fips package:

```
# zypper install dracut-fips
```

2. Recreate the INITRAMFS image:

```
# dracut -f
```

After regenerating the initrd, the crypto officer has to append the following string to the kernel command line by changing the setting in the boot loader:

```
fips=1
```

After editing the configuration file, please run the following command to change the setting in the boot loader:

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

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" respectively. For example:

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

The partition of /boot is located on /dev/sda1 in this example. Therefore, the following string needs to be appended to the kernel command line:

```
“boot=/dev/sda1”
```

Reboot to apply these settings.

In addition to the configuration of the kernel, the OpenSSH client configuration should contain the following rules:

- Either no "Ciphers" option or the option with a subset out of "aes128-ctr, aes192-ctr, aes256-ctr, aes128-gcm@openssh.com, aes256-gcm@openssh.com, aes128-cbc, 3des-cbc, aes192-cbc, aes256-cbc, rijndael-cbc@lysator.liu.se” - please note that any other cipher option is not allowed by the OpenSSH Client Module in FIPS Approved mode.
- Either no "MACs" option or the option with a subset out of "hmac-sha1-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com, hmac-sha1, hmac-sha2-256, hmac-sha2-512” – please note that any other MAC option is not allowed by the OpenSSH Client Module in FIPS Approved mode.
- Either no “KexAlgorithms” options or the option with a subset out of “diffie-hellman-group14-sha1, diffie-hellman-group-exchange-sha1, diffie-hellman-group-exchange-sha256, ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521” - please note that any other KexAlgorithms option is not allowed by the OpenSSH Client Module in FIPS Approved mode.

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

- "Protocol 2" must be specified – please note that protocol version 1 is disabled in FIPS Approved mode.
- Remove or do not configure DSA host keys as OpenSSH only supports 1024 bit DSA keys which is disallowed according to SP 800-131A. The crypto office should inform the user not to use DSA keys for user authentication.
- Remove or do not configure Ed25519 host keys as Ed25519 keys are not allowed in FIPS Approved mode. The crypto office should inform the user to not use Ed25519 keys for user authentication.
- When re-generating RSA host keys, the crypto officer should generate RSA keys with a size of 2048 bit or 3072 bit according to SP 800-131A. The crypto officer should inform the user not to use RSA keys with key sizes other than 2048 bits or 3072 bits.

Restart the OpenSSH client application (ssh) for changes to take effect.

A client public-private key pair that can be used for non-interactive authentication of a user to a server can be generated using 'ssh-keygen'. See the ssh-keygen(1) man page for documentation and setup instructions.

The OpenSSH client maintains a list of known servers and their public keys in /etc/ssh/ssh_known_hosts and per-user in ~/.ssh/known_hosts, which the crypto officer can install keys of known servers into. If any of these files is pre-populated using ssh-keyscan, the list of known public keys must be verified against a trusted source.

9.1.1. Configuration Changes and FIPS Approved Mode

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

9.2. User Guidance

See the ssh(1) man page for general usage documentation of the ssh client.

When connecting to an unknown server, the user will be prompted to verify a fingerprint of the server's public key. This must be done by consulting a trusted source.

9.3. Handling Self Test Errors

OpenSSL self test failures will prevent the OpenSSH Client Module from operating. See the Guidance section in the SUSE Linux Enterprise Server 12 - OpenSSL Module v2.0 or v3.0 FIPS 140-2 Security Policy for instructions on handling OpenSSL self test failures.

The OpenSSH Client Module self test only consists of the software integrity test. If the integrity test fails, the Module enters an error state. The only recovery from this type of failure is to reload the SUSE Linux Enterprise Server 12 - OpenSSH Client Module. If you downloaded the software, verify the package hash to confirm a proper download.

10. Mitigation of Other Attacks

The cryptographic Module is not designed to mitigate any specific attacks.

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

11. Glossary and Abbreviations

AES	Advanced Encryption Specification
CAVP	Cryptographic Algorithm Validation Program
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CVL	Component Verification List
DES	Data Encryption Standard
DRBG	Deterministic Random Bit Generator (this document refers to the DRBG types defined in SP 800-90A)
DSA	Digital Signature Algorithm
HMAC	Hash Message Authentication Code
KDF	Key Derivation Function
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
O/S	Operating System
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SSH	Secure Shell

Table 11, Abbreviations

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.

12. References

[1] FIPS 140-2 Standard, http://csrc.nist.gov/groups/STM/cmvp/standards.html

[2] FIPS 140-2 Implementation Guidance, http://csrc.nist.gov/groups/STM/cmvp/standards.html

[3] FIPS 140-2 Derived Test Requirements,http://csrc.nist.gov/groups/STM/cmvp/standards.html

[4] FIPS 197 Advanced Encryption Standard, http://csrc.nist.gov/publications/PubsFIPS.html

[5] FIPS 180-4 Secure Hash Standard, http://csrc.nist.gov/publications/PubsFIPS.html

[6] FIPS 198-1 The Keyed-Hash Message Authentication Code (HMAC), http://csrc.nist.gov/publications/PubsFIPS.html

[7] FIPS 186-4 Digital Signature Standard (DSS), http://csrc.nist.gov/publications/PubsFIPS.html

[8] NIST SP 800-67 Revision 1, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, http://csrc.nist.gov/publications/PubsFIPS.html

[9] NIST SP 800-56A, Recommendation for Pair-Wise Key Establishment Schemes using Discrete Logarithm Cryptography (Revised), http://csrc.nist.gov/publications/PubsFIPS.html

[10] NIST SP 800-90A, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, http://csrc.nist.gov/publications/PubsFIPS.html

[11] NIST SP 800-131A, Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths, http://csrc.nist.gov/publications/PubsFIPS.html

[12] NIST SP 800-135 Revision 1, Recommendation for Existing Application-Specific Key Derivation Functions, http://csrc.nist.gov/publications/PubsFIPS.html

[13] OpenSSH client man pages regarding ssh(1)

© 2017 SUSE Linux Products GmbH / atsec information security. This document can be reproduced and distributed only whole and intact, including this copyright notice.