SUSE Linux Enterprise Server 12 - StrongSwan Cryptographic Module Version 1.0 and 2.0

FIPS 140-2 Non-Proprietary Security Policy

Version 2.0

Last update: 2018-03-07

Prepared by:

atsec information security corporation

9130 Jollyville Road, Suite 260

Austin, TX 78759

www.atsec.com

©2018 SUSE Linux Products GmbH / atsec information security corporation

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Table of contents

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

1 Introduction

1.1 Purpose

This document is the non-proprietary Security Policy for the SUSE Linux Enterprise Server 12 - StrongSwan Cryptographic Module version 1.0 and 2.0. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS 140-2 (Federal Information Processing S24tandards Publication 140-2) for a security level 1 module.

This document was prepared as part of the requirements for conformance to FIPS 140-2 Level 1. It is intended for security officers, developers, system administrators and end-users.

FIPS 140-2 details the requirements of the Governments of the U.S. and Canada for cryptographic modules, aimed at the objective of protecting sensitive but unclassified information.

For more information on the FIPS 140-2 standard and validation program please refer to the NIST website at http://csrc.nist.gov/.

Throughout the document “SUSE Linux Enterprise Server 12 - StrongSwan Cryptographic Module”, “the StrongSwan Module”, or “the Module” are used interchangeably to refer to the SUSE Linux Enterprise Server 12 - StrongSwan Cryptographic Module.

1.2 Document Organization / Copyright

This non-proprietary Security Policy document may be reproduced and distributed only in its original entirety without any revision, ©2015 SUSE.

1.3 External Resources / References

The SUSE website (www.suse.com) contains information about SUSE Linux Enterprise Server.

The Cryptographic Module Validation Program website (http://csrc.nist.gov/groups/STM/cmvp/) contains links to the FIPS 140-2 certificate and SUSE contact information.

Appendix A contains the abbreviations and Appendix B contains the additional references.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

2 Cryptographic Module Specification

2.1 Module Overview

For FIPS 140-2 purposes, the SUSE Linux Enterprise Server 12 - StrongSwan Cryptographic Module is a software-only, security level 1 cryptographic module, running on a multi-chip standalone platform. The current version of the module is 1.0 and 2.0. The Module supplies cryptographic support of the IKEv2 protocols for the SUSE Linux Enterprise Server (SLES) user space.

Table 1 shows the security level claimed for each of the eleven sections that comprise the FIPS 140-2 standard:

FIPS 140-2 Section		Security Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services and Authentication	1
4	Finite State Model	1
5	Physical Security	N/A
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC	1
9	Self Tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A

Table 1: Security Levels

The StrongSwan Module does not have a direct dependency on AES-NI implementation, but it is bound to the SUSE Linux Enterprise Server 12 - OpenSSL Module with FIPS140-2 validation certificate #2435, which could be operated either with or without AES-NI enabled. Therefore, the StrongSwan Module has been tested in the following software configurations:

- 64-bit x86_64 with AES-NI disabled
- 64-bit x86_64 with AES-NI enabled

Table 2 shows the multi-chip standalone platforms on which the Module has been tested:

Module Version	Platform	Processor	Test Configuration
1.0	ProLiant DL320e Gen8	X86_64	SUSE Linux Enterprise Server 12 (with and without AES-NI)
2.0	FUJITSU Server PRIMERGY CX2570 M2 inside a CX400 M1 enclosure	Intel Xeon E5 family	SUSE Linux Enterprise Server 12 SP2 (with and without AES-NI)

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

2.0	IBM	Z13	SUSE Linux Enterprise Server 12 SP2

Table 2: Tested Platforms

This cryptographic module consists of the following components:

- strongswan-ipsec RPM file providing the IPsec protocol with the version of the RPM file of:
  - 5.1.3-15.1.x86_64 for v1.0
  - 5.1.3-26.5.1.x86_64 for v2.0
  - 5.1.3-26.5.1.s390x for v2.0
- Each RPM package contains the following files:
  - /usr/lib/ipsec/charon
  - /usr/lib/ipsec/duplichk
  - /usr/lib/ipsec/imv_policy_manager
  - /usr/lib/ipsec/pt-tls-client
  - /usr/lib/ipsec/scepclient
  - /usr/lib/ipsec/starter
  - /usr/lib/ipsec/stroke
- strongswan-libs0 RPM file providing the shared libraries with the version of the RPM file of:
  - 5.1.3-15.1.x86_64 for v1.0
  - 5.1.3-26.5.1.x86_64 for v2.0
  - 5.1.3-26.5.1.s390x for v2.0
- The following files are provided by the RPM. Note that the RPM also delivers other shared libraries which are not part of the module and are not accessible in FIPS mode.
  - /usr/lib64/ipsec/libcharon.so.0.0.0
  - /usr/lib64/ipsec/libstrongswan.so.0.0.0
  - /usr/lib64/ipsec/libtls.so.0.0.0
  - /usr/lib64/ipsec/plugins/libstrongswan-openssl.so
- The strongswan-hmac RPM file containing the HMAC integrity verification files for the 64-bit shared libraries with the version of the RPM file of:
  - 5.1.3-15.1.x86_64 for v1.0
  - 5.1.3-26.5.1.x86_64 for v2.0
  - 5.1.3-26.5.1.s390x for v2.0
- The following files are provided:
  - /usr/lib/ipsec/._fipscheck.hmac
  - /usr/lib/ipsec/.charon.hmac
  - /usr/lib/ipsec/.starter.hmac
  - /usr/lib/ipsec/.stroke.hmac

This document can be reproduced and distributed only whole and intact, including this copyright notice.

    - /usr/lib/ipsec/.libcharon.so.0.0.0.hmac
    - /usr/lib/ipsec/.libstrongswan.0.0.0.hmac
    - /usr/lib/ipsec/.libtls.0.0.0.hmac
    - /usr/lib/ipsec/plugins/.libstrongswan-openssl.hmac

- The dracut-fips package provides for the configuration of FIPS mode with the version of the RPM file of:
    - 037-37.2.x86_64 for v1.0
    - 044.1-109.26.1.x86_64 for v2.0
    - 044.1-109.26.1.s390x for v2.0
- It provides the following files:
    - /etc/dracut.conf.d/40-fips.conf
    - /usr/lib/dracut/modules.d/01fips/fips-boot.sh
    - /usr/lib/dracut/modules.d/01fips/fips-noboot.sh
    - /usr/lib/dracut/modules.d/01fips/fips.sh
    - /usr/lib/dracut/modules.d/01fips/module-setup.sh

- The fipscheck package which ensures the system boots into FIPS mode with the version of the RPM file of:
    - 1.2.0-9.3.x86_64 for v1.0
    - 1.2.0-9.3.x86_64 for v2.0
    - 1.2.0-9.3.s390x for v2.0
- provides the following files:
    - /usr/bin/.fipscheck.hmac
    - /usr/bin/fipscheck

The integrity check and all cryptographic operations other than the KDF in IKEV2 for the StrongSwan Module are performed by the SUSE Linux Enterprise Server 12 - OpenSSL Module with FIPS140-2 v2.0 and v3.0 (hereafter referred to as “the OpenSSL module”) with FIPS 140-2 validation certificate #2435 (v2.0) and #3038 (v3.0).

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

2.2 Modes of Operation

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

In FIPS approved mode, the Module will support the approved cryptographic algorithms as shown in Table 3. Column four lists the CAVP validation numbers obtained from NIST for successful validation testing of the implementation of the cryptographic algorithms on the platform as shown in Table 2.

In the FIPS approved mode of operation, the Module will require the following approved cryptographic algorithms from the OpenSSL module v2.0 and v3.0.

Algorithm	Usage	Keys/CSPs	CAVS Certs. (v2.0)	CAVS Certs. (v3.0)
Triple-DES (CBC)	Encryption and Decryption	Triple-DES keys 168 bits	Cert. #1823	Certs. #2439, and #2455
AES (CBC, CTR, CCM and GCM)	Encryption and Decryption	AES keys 128, 192 and 256 bits	Certs. #3197, #3198 and #3199	Certs. #4588, #4594, #4595, #4622, #4623, #4645, #4646, and #4647
SHA-1 SHA-224 SHA-256 SHA-384 SHA-512	Hashing	N/A	Certs. #2645, #2646 and #2648	Certs. #3768, #3769, #3770, #3771, #3788 and #3789
HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512	Message Integrity	At least 112 bits HMAC key	Certs. #2014, #2015 and #2016	Certs. #3042, #3043, #3044, #3045, #3059 and #3060
AES CMAC	Message Integrity Generation	AES keys 128, 192 and 256 bits	Certs. #3197, #3198. #3199	Certs. #4588, #4594, #4595, #4622, #4623, #4645, #4646, and #4647
Hash-based DRBG	SP 800-90A Random Number Generation	Seed and nonce	Certs. #674, #675 and #676	Certs. #1536, #1537, #1538, and #1539
Diffie-Hellman DLC primitive	SP 800-56A Key Agreement	Diffie-Hellman public and private components with key size 2048 and 3072 bits	CVL Cert. #431	CVL Cert. #1263

This document can be reproduced and distributed only whole and intact, including this copyright notice.

EC Diffie-Hellman DLC primitive	SP 800-56A Key Agreement	EC Diffie-Hellman public and private components with P-256, P-384 and P-521	CVL Cert. #431	CVL Cert. #1263
RSA	FIPS 186-4 Signature Generation and Verification	RSA keys 2048 and 3072 bits	Cert. #1628	Cert. #2505
ECDSA	FIPS 186-4 Signature Generation and Verification	ECDSA keys P-256, P-384 and P-521	Cert. #586	Cert. #1127

Table 3: Approved Algorithms - provided by the bound OpenSSL Module

The SUSE Linux Enterprise Server 12 - StrongSwan Cryptographic Module implements the IKEv2 protocol and the following cryptographic algorithm:

Algorithm	Function	CAVS Cert. (v1.0)	CAVS Cert. (v2.0)
SP 800-135 Key Derivation in IKEv2	Key derivation in IKEv2	CVL Cert. #486	CVL Certs. #1539 and #1541

Table 4: Approved Algorithm - provided by the StrongSwan Module

Note that in FIPS mode the Module only supports the IKEv2 protocol and that the IKEv1 protocol is not supported and should not be used. The IKEv2 protocol has been reviewed and tested by the testing lab but it has not been reviewed or tested by the CAVP or CMVP.

The following table shows the non-approved algorithms that can only be used in non-approved mode. Any use of these non-approved algorithm functions will cause the Module to operate in the non-FIPS mode implicitly.

Non-approved Algorithm	Usage / Description
Diffie-Hellman	Key agreement, key size smaller than 2048 bits or greater than 3072 bits
Camellia	Encryption and decryption

Table 5: Non-approved Algorithms – provided by the bound OpenSSL Module

Notes:

1. All cryptographic algorithms, approved and non-approved, other than the SP 800-135 Key Derivation in IKEv2 are provided by the SUSE Linux Enterprise Server 12 – OpenSSL Module v2.0 and v3.0 (FIPS 140-2 Validation #2435 and #3038)
2. The SUSE Linux Enterprise Server 12 - StrongSwan Cryptographic Module uses the SUSE Linux Enterprise Server 12 – OpenSSL Module v2.0 and v3.0 (FIPS 140-2 Validation #2435 and #3038) for standard cryptographic algorithms and integrity checking. It will require that a copy of FIPS 140-2 level 1 validated version of SUSE Linux Enterprise Server 12 – OpenSSL Module is installed on the system.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

2.3 Cryptographic Boundary

2.3.1 Hardware Block Diagram

The physical boundary of the Module is the surface of the case of the target platform. Figure 1 shows the hardware block diagram of the Module including the processors, memory, internal power supply, power interface, data status and control paths, etc. The bold line surrounding the hardware components represents the Module's physical cryptographic boundary. The hardware devices consist of standard integrated circuits and does not include any security-relevant, semi- or custom integrated circuits or other active electronic circuit elements.

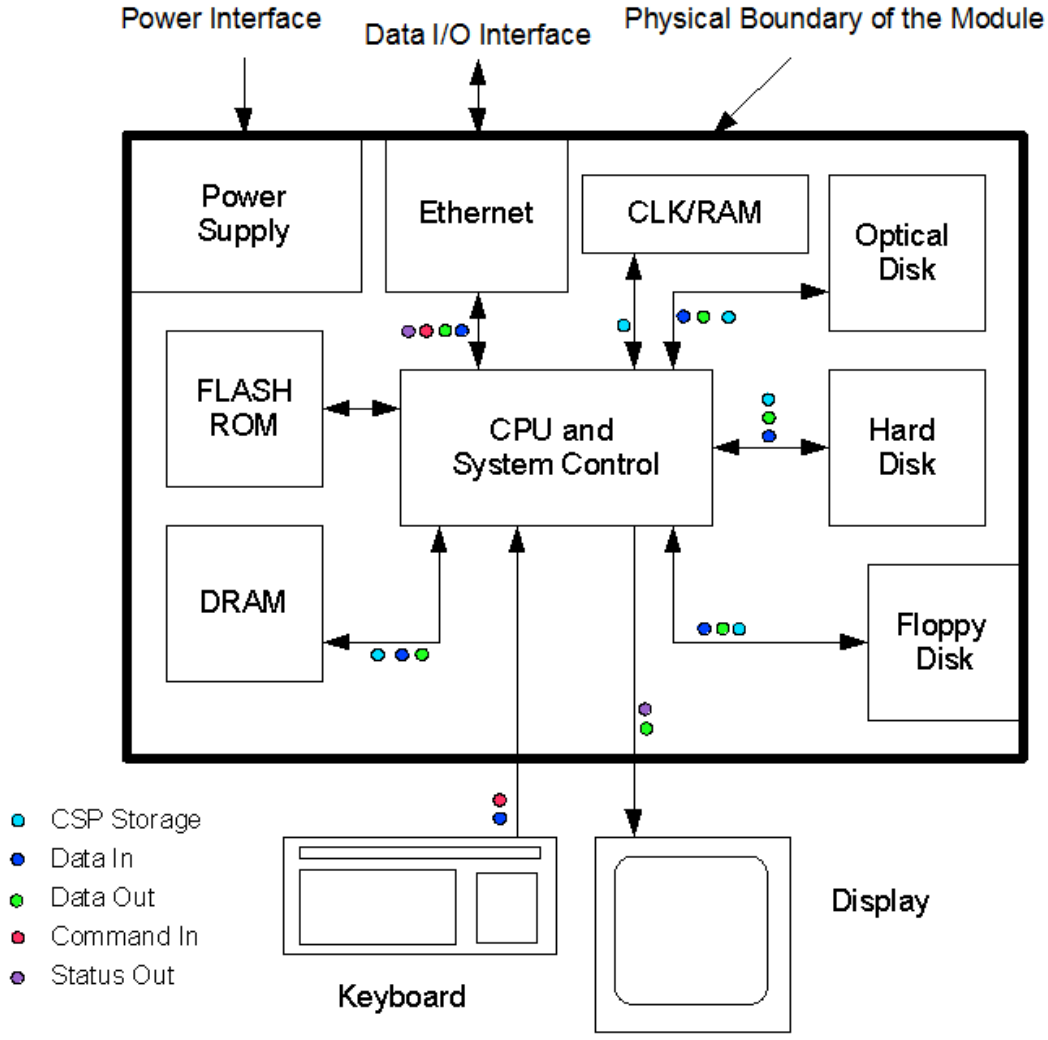

Figure 1: Hardware Block Diagram

2.3.2 Software Block Diagram

The logical boundary of the Module is shown in the Software Block Diagram (Figure 2) and contains:

- The SUSE Linux Enterprise Server 12 - StrongSwan Cryptographic Module
- The HMAC integrity verification file
- the dracut-fips package

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

The dracut-fips package is only used during boot time of the underlying operating system for the configuration of the Module. The OpenSSL module is a shared library to which the StrongSwan Module is bound. The OpenSSL module provides the standard cryptographic services to the StrongSwan Module.

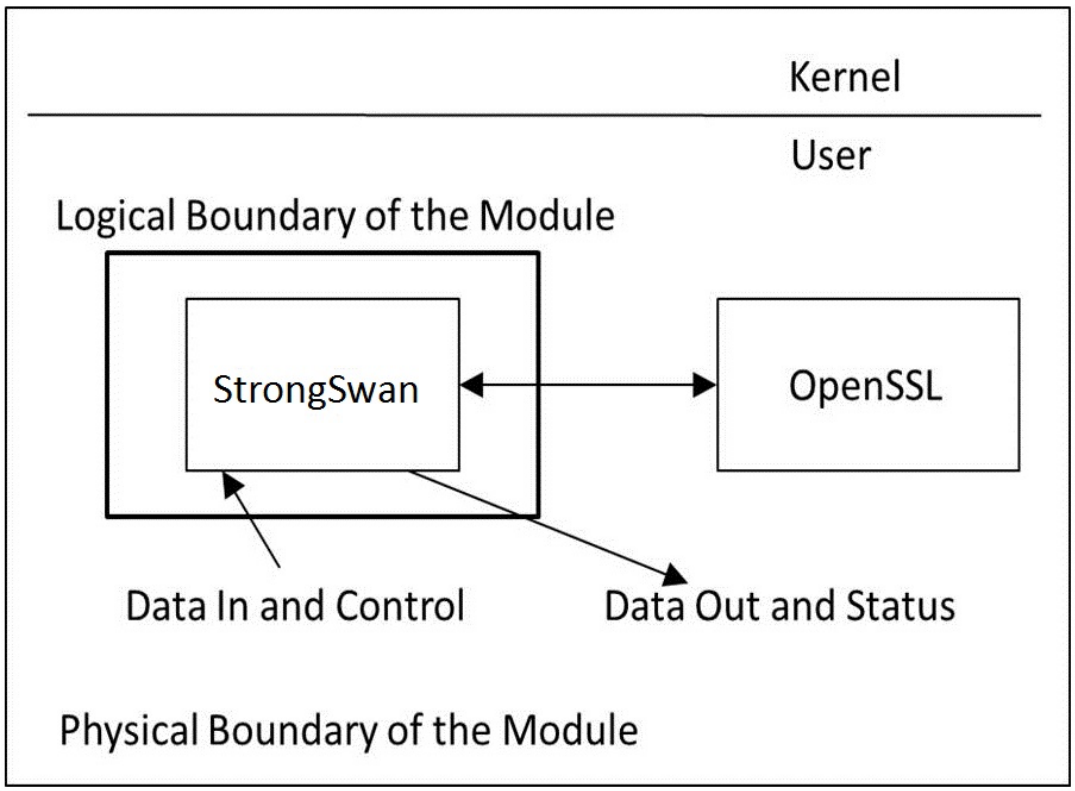

Figure 2: Software Block Diagram

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

3 Cryptographic 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 it runs.

The logical interfaces are the application program interface (API) through which applications request services. The following table summarizes the logical interfaces.

Module Logical interface	Description
Data Input	Host key file, ipsec.secrets and certificate file /etc/ipsec.d.cert, network
Data Output	Network
Control Input	Configuration files, keyboard (commands), command line options
Status Output	Display, network, system log
Power Input	Physical power connector of the General Purpose Computer

Table 6: Ports and Interfaces

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

4 Roles, Services and Authentication

4.1 Roles

The Module meets all FIPS 140-2 level 1 requirements for Roles and Services, implementing both User and Crypto Officer (CO) role. The Module does not allow concurrent operators.

- User role: performs all services, except module installation and configuration.
- CO role: performs installation and configuration.

The User and CO roles are implicitly assumed by the entity accessing the services implemented by the module. No further authentication is performed or required.

4.2 Services

The Module provides services to authorized operators of either the User or CO role according to the applicable FIPS 140-2 security requirements.

Table 7 contains the approved services provided by the StrongSwan Module in the approved mode. For each available service, it lists the associated role(s), the critical security parameters (CSPs) and cryptographic keys involved, and the type(s) of access to the CSPs and cryptographic keys. Please refer to Table 3 and 4 for approved key sizes of the approved algorithms and the algorithm certificates.

The access types are denoted as follows:

- 'R': The item is read or referenced by the service
- 'W': The item is written or updated by the service
- 'Z': The persistent item is zeroized by the service

Service	Algorithm / Function	Role	Cryptographic Keys and CSPs Accessed (source module)	Access Type
Install and Configure the module	None	CO	RSA or ECDSA private/public key (OpenSSL)	RWZ
Manage Charon IKE daemon (start, stop, etc.)	Commands	CO	DRBG seed and nonce (N/A, provided by /dev/urandom)	RW
Negotiate IKE to establish security associations (SAs)	RSA/ECDSA Signature Generation and Verification, Hash-based DRBG, Diffie-Hellman and EC Diffie-Hellman, AES/Triple-DES Encryption and Decryption,	User	Servers RSA or ECDSA private/public key (OpenSSL) Peers RSA or ECDSA public key (N/A) DRBG seed and nonce (N/A) Diffie-Hellman and EC Diffie-Hellman private and public key components (OpenSSL)	RWZ RWZ RW RW

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Service	Algorithm / Function	Role	Cryptographic Keys and CSPs Accessed (source module)	Access Type
	SHS Keyed-Hash, HMAC (provided by the bound OpenSSL module. Please see Table 3) SP 800-135 KDF in IKEv2 (provided by the StrongSwan module. Please see Table 4) Zeroize of servers RSA/ECDSA key		AES or Triple-DES encryption keys (ISAKMP SA tunnel encryption key, IKEv2 SA tunnel encryption key and IPSec SA tunnel encryption key) (StrongSwan) HMAC data authentication key (OpenSSL)	RW RW
Close Association	Zeroize	User	Diffie-Hellman and EC Diffie-Hellman public and private components (OpenSSL) AES or Triple-DES encryption keys (ISAKMP SA tunnel encryption key, IKEv2 SA tunnel encryption key and IPSec SA tunnel encryption key) (StrongSwan) HMAC data authentication key (OpenSSL)	Z Z Z
Self-test	Start/restart of module	CO	Software Integrity key for StrongSwan (HMAC SHA-256) (StrongSwan) Software Integrity key for OpenSSL (HMAC SHA-256) (OpenSSL) Software Integrity key for fipscheck (HMAC SHA-256) (OpenSSL)	R R R
Show status	Exit codes	CO	None	N/A

Table 7: Approved Services

Notes:

- The Show Status indicator is via the messages sent to syslog to the location specified on the syslog configuration on the system.

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

- Self tests are run when the Module is started. They can be run on demand by restarting the Module.

The following table lists the non-approved service provided by the StrongSwan Module in non-approved mode.

Service	Algorithm / Function	Role	Access Type
Negotiate IKE to establish security associations (SAs)	Camellia Encryption and Decryption, Diffie-Hellman with non-approved key size as listed in Table 5 (The non-approved cipher suites can be configured in ipsec.conf file)	User	RWZ

Table 7A: non-Approved Service

4.3 Authentication

The Module does not implement authentication. The role is implicitly assumed on entry.

4.4 Mechanism and Strength of Authentication

The Module does not implement authentication.

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

5 Physical Security

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

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

6 Operational Environment

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

6.1 Policy

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

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

In FIPS 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.

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

7 Cryptographic Key Management

7.1 Random Number Generation

The Module employs a SP 800-90A deterministic random bit generator (DRBG) which is called from the OpenSSL module.

The Linux kernel provides /dev/urandom as a source of random numbers for DRBG seeds. The Linux kernel initializes this pseudo device at system startup.

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.

7.2 Key Life Cycle Table

The following table identifies the cryptographic keys and 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 via the OpenSSL module.

Key	Type	Generation	Establish-ment	Entry and Output method	Storage	Zeroization
RSA Private and Public Keys	RSA key	N/A	N/A	N/A	Plaintext	Immediately after use
ECDSA Private and Public Keys	ECDSA key	N/A	N/A	N/A	Plaintext	Immediately after use
ISAKMP SA Tunnel Encryption Keys	AES or Triple-DES	N/A	Established during the ISAKMP SA handshake using Diffie-Hellman or EC Diffie-Hellman	N/A	Ephemeral	Close of ISAKMP SA or termination of the Charon daemon
IKEv2 SA Tunnel Encryption Keys	AES or Triple-DES	N/A	Established during the IKEv2 SA handshake using Diffie-Hellman or EC Diffie-Hellman and SP 800-135 IKEv2 KDF	N/A	Ephemeral	Close of IKEv2 SA or termination of the Charon daemon
IPSec SA Tunnel Encryption Keys	AES or Triple-DES	N/A	Established during the IPSec handshake using	N/A	Ephemeral	Close of ISAKMP SA and IKEv2 SA or overwritten

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Key	Type	Generation	Establish-ment	Entry and Output method	Storage	Zeroization
			Diffie-Hellman or EC Diffie-Hellman			by renegotiated IPSec SA or termination of the Charon daemon
Diffie-Hellman Private and Public Parameters	Diffie-Hellman	SP 800-90A Hash-based DRBG	N/A	N/A	Ephemeral	Close of ISAKMP SA and IKEv2 SA or termination of the Charon IKE daemon
EC-Diffie-Hellman Private and Public Parameters	EC-Diffie-Hellman	SP 800-90A Hash-based DRBG	N/A	N/A	Ephemeral	Close of ISAKMP SA and IKEv2 SA or termination of the Charon IKE daemon
DRBG seed	SP 800-90A Hash-based DRBG	N/A	N/A	N/A (see section 7.1), provided by /dev/urandom	Ephemeral	N/A
DRBG nonce	SP 800-90A Hash-based DRBG	N/A	N/A	N/A (see section 7.1), provided by /dev/urandom	Ephemeral	N/A
Software Integrity Key for OpenSSL application	HMAC SHA-256	N/A	N/A	N/A	Plaintext within the OpenSSL library	Termination of the fipscheck application
Software Integrity Key for fipscheck application	HMAC SHA-256	N/A	N/A	N/A	Plaintext within the fipscheck library	Termination of the fipscheck application
Software Integrity Key for StrongSwan application	HMAC SHA-256	N/A	N/A	N/A	Plaintext within the StrongSwan library	Termination of the fipscheck application

Table 8: Key Life Cycle

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Notes:

The Module ships without containing any keys and CSPs. When the Module is configured, the CO specifies secrets (RSA private keys, x.509 certificates or eXtended Authentication passwords (XAUTH)) in the /etc/ipsec.secrets file (see the ipsec.secrets(5) man page).

A secret is associated with the correct entity using optional selectors. A selector is an IP address, a fully qualified domain name, user@FQDN, %any or %any6. To authenticate a connection between two hosts, the entry that most specifically matches the host and peer IDs is used. An entry with no selectors will match any host and peer. An entry with one selector will match a host and peer if the selector matches the host's ID (the peer is not considered). An entry with multiple selectors will match a host and peer if the host ID and peer ID each match one of the selectors. If the key is for an asymmetric authentication technique (e.g., RSA) an entry with multiple selectors will match a host and peer even if only the host Id matches a selector (all selectors are assumed to be identities of the host).

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.

7.3 Key Zeroization

For volatile memory, overwriting with hex 0 is included in deallocation operations. There are no restrictions when zeroizing any cryptographic keys and CSPs.

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

8 Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

The test platform as shown in Table 2 is compliant to 47 CFR FCC Part 15, Subpart B, Class A (Business use).

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

9 Self Tests

9.1 Power-Up Tests

Software Integrity Test: all cryptographic function tests are performed by the SUSE Linux Enterprise Server 12 - OpenSSL Module v2.0 and v3.0 before it will perform cryptographic operations for the StrongSwan Module.

9.1.1 Software Integrity Test Details

StrongSwan userspace modules have their integrity verified at startup by the software integrity test.

The integrity check is performed by the SUSE Linux Enterprise Server 12 - OpenSSL Module utility fipscheck using HMAC-SHA256.

When the Module starts, it exercises the power-on self test, including the software integrity test. The software integrity test (HMAC-SHA256) constitutes a known answer test for the HMAC-SHA256 algorithm.

The user space integrity verification is performed as follows:

The StrongSwan application links with the library libfipscheck.so which is intended to execute fipscheck to verify the integrity of the calling application file using HMAC-SHA256. Upon calling the FIPSCHECK_verify() function provided with libfipscheck.so, the fipscheck application is loaded and executed, and the following steps are performed:

- OpenSSL, as loaded by fipscheck, performs the integrity check of the OpenSSL library files using HMAC-SHA256.
- The application fipscheck performs the integrity check of its application file using HMAC-SHA256 provided by OpenSSL.
- The fipscheck application performs the integrity check of the calling application. The fipscheck computes the HMAC-SHA256 checksum of the file from the command line and compares the computed value to the value stored inside the /path/to/application/.<applicationfilename>.hmac checksum file. The fipscheck application returns the appropriate exit value based on the comparison result (zero if the checksum is OK – which is enforced by the libfipscheck.so library). The fipscheck application also automatically verifies the integrity of libfipscheck when loaded.

No operator intervention is required during the running of the self tests.

Successful or unsuccessful completion of the power-up and integrity tests is displayed via the messages sent to syslog to the location specified on the syslog configuration on the system.

See section 10.3 for descriptions of possible self test errors and recovery procedures.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

10 Guidance

Password-based encryption and password-based key generation do not provide sufficient strength to satisfy FIPS 140-2 requirements. As a result, data processed with password-based encryption methods are considered to be unprotected.

NOTE: The SUSE Linux Enterprise Server 12 - StrongSwan Cryptographic Module requires that a copy of a FIPS 140-2 validated version of the SUSE Linux Enterprise Server 12 - OpenSSL Module (validation certificate #2435 and #3038) be installed on the same operational environment.

10.1 Crypto Officer Guidance

The version of the RPM containing the validated module is stated in section 2 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.

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

For proper operation of the in-module integrity verification, the prelink must be disabled. This can be done by setting PRELINKING=no in the /etc/sysconfig/prelink configuration file. If the libraries were already prelinked, the prelink should be undone on all the system files using the 'prelink -u -a' command.

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 parameter in the /etc/default/grub configuration file in the GRUB_CMDLINE_LINUX_DEFAULT line:

```
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.

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

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

10.1.1 Configuration Changes and FIPS Approved Mode

Use caution 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, it does not enable the FIPS approved mode.

All user space modules depend on this file for transitioning into FIPS approved mode.

10.2 User Guidance

See the StrongSwan(8) man page for general usage documentation of StrongSwan.

10.3 Handling Self Test Errors

OpenSSL self test failures may prevent the StrongSwan Module from operating. See the Guidance section in the Security Policy of SUSE Linux Enterprise Server 12 – OpenSSL Module for instructions on handling OpenSSL self test failures.

The StrongSwan Cryptographic Module self test consists of the software integrity test. If the integrity test fails, StrongSwan enters an error state. When an error occurs, a message is written to the syslog. The only recovery from this type of failure is to reload the SUSE Linux Enterprise Server 12 - StrongSwan Cryptographic Module. If the user downloaded the software, the package hash needs to be verified to confirm a proper download.

Conditional tests are performed within the bound SUSE Linux Enterprise Server 12 – OpenSSL Module. See the OpenSSL Security Policy for details on conditional test failures. OpenSSL self test failures may prevent the StrongSwan Module from operating.

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

11 Mitigation of Other Attacks

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

Appendix A Glossary and Abbreviations

AES	Advanced Encryption Specification
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining Message Authentication Code
CMAC	Cipher-based Message Authentication Code
CFB	Cipher Feedback
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter Mode
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
FIPS	Federal Information Processing Standards Publication
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
O/S	Operating System
PKCS	Public Key Cryptography Standards
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SSH	Secure Shell
TDES	Triple-DES

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

XTS	XEX Tweakable Block Cipher with Ciphertext Stealing

Appendix B References

FIPS 140-2 PUB **Security Requirements for Cryptographic Modules**
January 2011
http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

FIPS180-4 **Secure Hash Standard (SHS)**
March 2012
http://csrc.nist.gov/publications/fips/fips180-4/fips 180-4.pdf

FIPS186-4 **Digital Signature Standard (DSS)**
July 2013
http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

FIPS197 **Advanced Encryption Standard**
November 2001
http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

FIPS198-1 **The Keyed Hash Message Authentication Code (HMAC)**
July 2008
http://csrc.nist.gov/publications/fips/fips198 1/FIPS-198 1_final.pdf

PKCS#1 **Public Key Cryptography Standards (PKCS) #1: RSA Cryptography**
Specifications Version 2.1
February 2003
http://www.ietf.org/rfc/rfc3447.txt

SP800-38A **NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques**
December 2001
http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

SP800-38B **NIST Special Publication 800-38B - Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication**
May 2005
http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf

SP800-38C **NIST Special Publication 800-38C - Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality**
May 2004
http://csrc.nist.gov/publications/nistpubs/800-38C/SP800-38C_updated July20_2007.pdf

SP800-38D **NIST Special Publication 800-38D - Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC**
November 2007
http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.

SP800-38E **NIST Special Publication 800-38E - Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices**
January 2010
http://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf

SP800-56A **NIST Special Publication 800-56A Revision 2 - Recommendation for Pair Wise Key Establishment Schemes Using Discrete Logarithm Cryptograph**y
May 2013
http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800 56Ar2.pdf

SP800-67 **NIST Special Publication 800-67 Revision 1 - Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher**
January 2012
http://csrc.nist.gov/publications/nistpubs/800-67-Rev1/SP-800-67-Rev1.pdf

SP800-90A **NIST Special Publication 800-90A - Recommendation for Random Number Generation Using Deterministic Random Bit Generators**
January 2012
http://csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf

©2018 SUSE Linux Products GmbH / atsec information security corporation
This document can be reproduced and distributed only whole and intact, including this copyright notice.