ORACLE
Linux

FIPS 140-2 Non-Proprietary Security Policy

Oracle Linux 7 Libreswan Cryptographic Module

FIPS 140-2 Level 1 Validation

Software Version: R7-7.8.0

Date: July 11th, 2022

Title: Oracle Linux 7 Libreswan Cryptographic Module Security Policy
Date: July 11th, 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

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 Libreswan Cryptographic Module by Oracle Corporation. Oracle Linux 7 Libreswan 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 Libreswan Cryptographic Module functions in order to meet the FIPS 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 Libreswan 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 Libreswan 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 Libreswan Cryptographic Module

2.1 Functional Overview

The Oracle Linux 7 Libreswan Cryptographic Module is a framework for providing cryptographic services to other network entities implementing the IKEv1 and IKEv2 protocols.

The cryptographic module combines a vertical stack of Oracle Linux components, and the module intends to limit implementations, which are proved by each separate component, to the external interface. Note: This security policy only covers the IKE protocol, which is a part from the IPsec protocol family.

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 Libreswan Cryptographic Module is a software-only multi-chip standalone module as defined by the requirements within FIPS PUB 140-2. The components within the cryptographic boundary comprising the module are listed as follows:

- Pluto IKE Daemon and it's HMAC file: This comes as part of the Libreswan RPM package libreswan-3.25-9.1.0.3.el7_8.x86_64 or libreswan-3.25-9.1.0.3.el7_8.aarch64.
- Fipscheck library package version fipscheck-lib-1.4.1-6.el7.x86_64.rpm or fipscheck-lib-1.4.1-6.el7.aarch64.rpm with its HMAC file and fipscheck application with it's HMAC file: This is provided as part of the fipscheck RPM package version fipscheck-lib-1.4.1-6.el7.x86_64.rpm or fipscheck-lib-1.4.1-6.el7.aarch64.rpm. Fipscheck performs the integrity validation of itself along with the IKE Daemon binary.

The following components which act as bound modules need to be installed for the Oracle Linux 7 Libreswan Cryptographic Module to operate:

- The bound module Oracle Linux 7 NSS Cryptographic Library with FIPS 140-2 Certificate #4171 (hereafter referred to as the "NSS module") provides cryptographic algorithms used by the IKE Daemon. The IKE Daemon uses the NSS module in accordance with the Security Rules stated in the NSS Cryptographic Library Security Policy.

- The bound module Oracle Linux 7 OpenSSL Library with FIPS 140-2 Certificate #4170 (hereafter referred to as the "OpenSSL module") provides HMAC SHA-256 algorithm required by fipscheck application and library for integrity check.

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

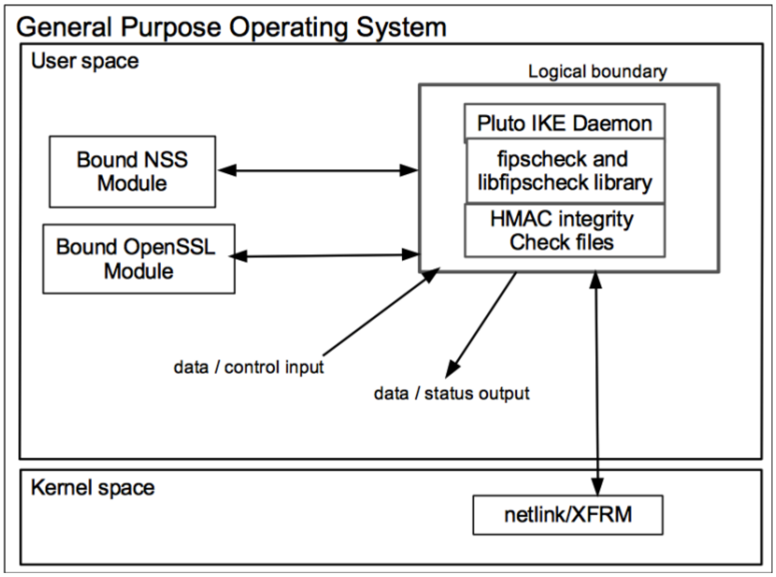

Figure 1 – Oracle Linux 7 Libreswan Logical Cryptographic Boundary

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.

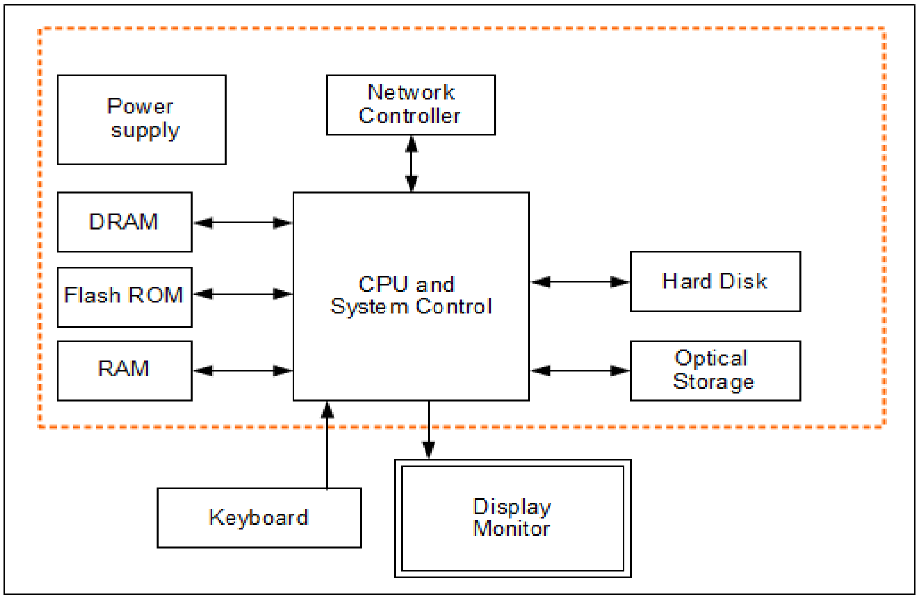

Figure 2 – Oracle Linux 7 Libreswan Hardware Block Diagram

3.3 Modes of Operation

The Module supports two modes of operation: FIPS Approved mode and non-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 8.

3.4 Approved or Allowed Security Functions

The Oracle Linux 7 Libreswan Cryptographic Module contains the following FIPS Approved algorithms:

Approved Security Functions		Certificate
Key Derivation (NIST SP 800-135)		
KDF IKE (CVL)	**Generic_C:** IKEv1, IKEv2 (SHA 1, 256, 384, 512)	A 1903

Table 2: FIPS Approved or Allowed Security Functions

The Libreswan and the bound NSS module together provide the Diffie Hellman and EC Diffie Hellman key agreement in accordance with IG D.8 path X1 (2). The Libreswan module only implements the KDF portion of the key agreement as stated in the above table. The bound NSS module provides the shared secret computation as stated in Table 3 along with scheme and the key sizes used.

- KAS-FFC-SSC (Cert. #A 1179, CVL Cert. #A 1903, key agreement; key establishment methodology provides between 112 and 200 bits of encryption strength);

- KAS-ECC-SSC (Cert. #A 1179, CVL Cert. #A 1903, key agreement; key establishment methodology provides between 128 and 256 bits of encryption strength);

3.5 Approved or Allowed Security Functions Provided by the NSS Bound Module

The following table shows the Approved or allowed security functions from the bound module:

Algorithm	Certificate[1]
Provided by Bound NSS Module	
AES CBC, CTR (128, 192, 256)	A 1097, A 1179
AES CBC (128, 192, 256)	A 2580
Triple-DES CBC	A 1179
ECDSA Key generation: P-256, P-384, P-521	A 1179
Safe Primes Key Generation: MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192	A 1179
RSA Signature Generation: 2048, 3072, 4096; Signature Verification: 2048, 3072, 4096	A 1179
SHA-1, SHA-256, SHA-384, SHA-512	A 1179
HMAC (SHA-1, SHA-256, SHA-384, SHA-512)	A 1179
DRBG Hash_DRBG (SHA-256)	A 1179
KAS-FFC-SSC (dhEphem KAS Role: initiator, responder) Mod P Methods: MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192	A 1179
KAS-ECC-SSC P-256, P-384, P-521 (ephemeralUnified KAS Role: initiator, responder)	A 1179
ENT (NP) (NIST SP800-90B)	N/A
Provided by Bound OpenSSL Module	
HMAC-SHA-256 (for integrity check only), SHA-256 (Prerequisite algorithm for HMAC-SHA-256)	A 1224, A 1225, A 1226, A 1227, A 2395

Table 3: Approved Security Functions from the Bound Modules

[1] The CAVP certificates from the bound modules include additional modes and key sizes for the respective algorithms but only a subset of it that is listed in Table 3 is used by the Libreswan module.

3.6 Non-Approved Security Functions from the Bound NSS Module

The use of following non-Approved algorithms will put the module in non-approved mode implicitly:

Algorithm	Usage
AES GCM	Encryption/decryption
RSA	Signature generation and verification with key size less than 2048 bits or greater than 4096 bits
Diffie-Hellman	Shared Secret Computation with key size less than 2048 bits
MD5	Hashing used as part of HMAC PRF

Table 4: Non-Approved Functions from the Bound NSS Module

4. Module Ports and Interfaces

The module FIPS 140 interfaces can be categorized as follows:

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

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.

Table below shows a mapping of FIPS 140 interfaces to logical ports:

FIPS 140 Interface	Physical Port	Module Interfaces
Data Input	Ethernet Ports	IKE Network Port/Protocol, NSS Key Database file stored in /etc/ipsec.d/
Data Output	Ethernet Ports	IKE Network Port/Protocol, Linux Kernel (netlink/XFRM Interface)
Control Input	Management Ethernet Port, USB for Keyboard/Mouse, Serial Port	IKE Network Port/Protocol, Configuration Files (/etc/ipsec.conf, /etc/ipsec.d/, /etc/ipsec.secrets), Linux Kernel (netlink/XFRM Interface), command line
Status Output	Management Ethernet Port, Serial Port	Log File, IKE Network Port/Protocol

Table 5: 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

6.1 Tested Environments

The module operates in a modifiable operational environment per FIPS 140-2 level 1 specifications. The module was tested on the following environments without PAA:

Operating Environment	Processor	Hardware
Oracle Linux 7.8 64-bit	Intel® Xeon® 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 6: Tested Operating Environments

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

Table 7: Vendor Affirmed Operational Environments

Note: CMVP makes no statement as to the correct operation of the module when so ported if the specific operational environment is 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 that request cryptographic services is the single user of the module.

In operational mode, the ptrace(2) system call, the debugger (gdb(1)), and strace(1) shall be not used.

7. Roles, Services and Authentication

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

7.1 Roles

The module supports the following roles:

- **User Role**: performs key derivation and negotiates IKE to establish security association.
- **Crypto Officer Role**: performs module installation and configuration, manages Pluto IKE Daemon, self-tests, show status and Zeroize.

The module is a Security Level 1 software-only cryptographic module and does not implement authentication. The User and Crypto Officer roles are implicitly assumed by the entity accessing the module services.

7.2 FIPS Approved Operator Services and Descriptions

The module supports following services in approved mode:

U	CO	Service Name	Service Description	Keys and CSP(s)	Access Type
	X	Install and Configure the Module	Installation and secure configuration of the cryptographic module	RSA Private Key, pre-shared key	R, W, X, Z
	X	Manage Pluto IKE Daemon	Manage Pluto IKE daemon like start and stop activities and destroy keys.	all Keys and CSP's	R, X, Z
X		Negotiate IKE to Establish Security Associations (SA's)	Negotiate key agreement using IKE to establish security associations	RSA, KAS-ECC-SSC/KAS-FFC-SSC private keys, KAS-ECC-SSC/KAS-FFC-SSC shared secret, IKE SA Tunnel Encryption Keys, IKE SA Tunnel Integrity Keys, IPsec SA encryption keys, IPsec SA Tunnel Integrity Keys	R, W, X
	X	Self-Tests	Execute integrity test	HMAC-SHA-256 key	X
	X	Show Status	Show status of the module	None	N/A
	X	Zeroize	Zeroize keys and CSP's	Keys and CSP's	X, Z

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

Table 8: FIPS Approved Services

The module supports following services in non-approved mode.

U	CO	Service Name	Service Description	Keys	Access Type
	X	Install and Configure the Module	Installation and secure configuration of the cryptographic module	RSA private key listed in Table 4	R, W, X, Z
X		Negotiate IKE to Establish Security Associations (SA's)	Negotiate key agreement using IKE to establish security associations with algorithms in Table 4	RSA, DH private key listed in Table 4	R, W, X

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

Table 9: Non-FIPS Approved Services

7.3 Authentication

The module is a Security Level 1 software-only cryptographic module and does not implement authentication. The role is implicitly assumed based on the service requested.

8. Key and CSP Management

The following keys, cryptographic key components and other critical security parameters are used by the Module. The module leverages the Oracle Linux NSS Cryptographic Library for cryptographic services with the exception of HMAC-SHA-256 used for integrity test that is performed by the Oracle Linux 7 OpenSSL Cryptographic Library and the key derivation components that are contained within the Pluto IKE Daemon.

CSP	Use	Storage	Zeroization
RSA Private Key	Keys used for peer authentication.	Ephemeral	Close of IKE SA or termination of Pluto IKE Daemon
Pre-shared key	Pre-computed value provided to the module used to derive session keys for the IKE with PSK lengths of 8, 384, 768, 8192 bits for IKEv1.	Ephemeral	Close of IKE SA or termination of Pluto IKE Daemon
KAS-ECC-SSC/KAS-FFC-SSC private key	keys used for key agreement. Generated by the bound NSS module.	Ephemeral	Close of IKE SA or termination of Pluto IKE Daemon
KAS-ECC-SSC/KAS-FFC-SSC Shared Secret	Shared secret established by KAS-ECC-SSC/KAS-FFC-SSC key agreement	Ephemeral	Close of IKE SA or termination of Pluto IKE Daemon
IKE SA Tunnel Encryption Keys (AES, 3-key Triple-DES)	Session keys derived from shared secret by KDF	Ephemeral	Close of IKE SA or termination of Pluto IKE Daemon
IKE SA Tunnel Integrity Keys (HMAC)	Derived from shared secret by KDF. Provide data integrity of data traffic travelling over the IKE secure tunnel.	Ephemeral	Close of IKE SA or termination of Pluto IKE Daemon
IPsec SA Tunnel Encryption Keys (AES, 3-key Triple-DES)	Session keys derived from shared secret by KDF	Ephemeral	Close of IKE SA or overwritten by re-negotiated IPsec SA or termination of Pluto IKE Daemon
IPsec SA Tunnel Integrity Keys (HMAC)	Derived from shared secret by KDF. Provide data integrity of data traffic travelling over the IPsec secure tunnel.	Ephemeral	Close of IKE SA or overwritten by re-negotiated IPsec SA or termination of Pluto IKE daemon

Table 10: CSP Table

Note: The RSA private keys are encrypted by the NSS module. When an operation requires a private key, the first pointer or handle to the private key is obtained using the public key and CKA_ID (key ID). Only during the operation, private keys are decrypted, and the operation is performed. After the operation, the memory pointing to the private key is zeroized by the NSS module.

8.1 Random Number 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.

8.2 Key/CSP Storage

Public and private keys are provided to the module by the calling process and are destroyed when released by the appropriate IKE Network Port/Protocol. The module does not perform persistent storage of keys.

8.3 Key/CSP Zeroization

The destruction functions overwrites the memory that is occupied by the key information with predefined value before it is deallocated.

9. Self-Tests

9.1 Power-Up Self-Tests

The module performs power-up tests at module initialization which includes the software integrity test to ensure that the module is not corrupted. The self-tests are triggered automatically without any user intervention. While the module is performing the power-up tests, services are not available and data input or output is inhibited.

9.2 Integrity Tests

The integrity check is performed by the fipscheck application using the HMAC-SHA-256 algorithm implemented by the bound OpenSSL module. The OpenSSL module computes an HMAC-SHA-256 value for the fipscheck utility, as well as all applications forming the Libreswan module. The integrity verification is performed as follows:

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

1. Fipscheck loads the OpenSSL module, which performs its own integrity check using the HMAC-SHA-256 algorithm;
2. Fipscheck performs the integrity check of its own application file using the HMAC-SHA-256 algorithm provided by the OpenSSL module;
3. Fipscheck automatically verifies the integrity of libfipscheck.so library before processing requests of calling applications;
4. The fipscheck application performs the integrity check of the Libreswan application file as follows:

The fipscheck computes the HMAC-SHA-256 checksum of the file and compares the computed value to 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, which is enforced by the libfipscheck.so library. Otherwise, an error code will be shown, which puts the module into the error state.

If any of the above steps fails, an error code (a non-zero value) will be returned and the module enters the error state. In Error state, all output is inhibited and no cryptographic operation is allowed. The Module needs to be reinitialized in order to recover from the Error state.

The power-up self tests can be performed on demand by reinitializing the Module.

9.3 Cryptographic Algorithm Tests

The Libreswan module will use the Oracle Linux 7 NSS Cryptographic Module and the Oracle Linux 7 OpenSSL Cryptographic Module as bound modules which provides the underlying cryptographic algorithms and their respective POST’s.

The following table provides the list of Known Answer Test (KAT) for the Libreswan module:

Algorithm	Test
IKE KDF	IKEv1 and IKEv2 KAT with SHA-256.

Table 11: Power-On Self-Tests for Libreswan Module

9.4 Conditional Self-Tests

Conditional tests are performed during operational state of the module when the respective crypto functions are used. If any of the conditional tests fails, module transitions to error state.

The Oracle Linux 7 Libreswan cryptographic module does not implement any conditional self-tests. All conditional self-tests are implemented by the bound NSS module.

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

All cryptographic functions for the Oracle Linux 7 Libreswan Cryptographic Module will be provided by a copy of a FIPS 140-2 validated version of the NSS module. The HMAC-SHA-256 algorithm provided by the bound OpenSSL module is used to perform integrity verification. Below is a list of actions needed to configure the Pluto IKE daemon:

As stated in Guidance section of Oracle Linux 7 OpenSSL Cryptographic Module and Oracle Linux 7 NSS Cryptographic Module security policy, after configuring the operating environment to support FIPS, the file /proc/sys/crypto/fips_enabled will contain 1. If the file does not exist or does not contain “1”, the operating environment is not configured to support FIPS and the module will not operate as a FIPS validated module.

The version of the RPM containing the validated module is stated in section 3.1 above. The RPM package of the Module shall 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). The integrity of the RPM is automatically verified during the installation of the Module and the Crypto Officer shall not install the RPM file if the Oracle Linux Yum Server indicates an integrity error. The RPM files listed in section 3 are signed by Oracle and during installation; Yum performs signature verification which ensures as secure delivery of the cryptographic module. If the RPM packages are downloaded manually, then the CO should run ‘rpm –K <rpm-file-name>’ command after importing the builder’s GPG key to verify the package signature. In addition, the CO can also verify the hash of the RPM package to confirm a proper download.

To set up the module in FIPS capable environment, perform the following steps:

1. Install Libreswan RPM file e.g for x86_64 or aarch use yum command
   ```
   # yum install  libreswan-3.25-9.1.0.3.el7_8.x86_64.rpm or libreswan-3.25-9.1.0.3.el7_8.aarch64.rpm
   ```
2. Install 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
   ```
3. Install the dracut-fips package:
   ```
   # yum install dracut-fips-033-572.0.5.el7.x86_64.rpm or dracut-fips-033-572.0.5.el7.aarch64.rpm
   ```
4. Install the dracut-fips-aesni package (if AES-NI is supported):
   To check if AES-NI is supported run:
   ```
   # grep aes /proc/cpuinfo
   ```
   If it is supported, run:
   ```
   # yum install dracut-fips-aesni-033-572.0.5.el7.x86_64.rpm or dracut-fips-aesni-033-572.0.5.el7.aarch64.rpm
   ```
5. Recreate the INITRAMFS image:
   ```
   # dracut -f
   ```
6. Perform the following steps to configure the boot loader so that the system boots into FIPS capable environment:

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.

7. 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
```

8. Reboot the system

9. Verify that FIPS capable environment is enabled in the Operating System by running the command:

```
# cat /proc/sys/crypto/fips_enabled
```

The response should be “1”

10. Configure Pluto as specified in ipsec.conf(5), and ipsec.secrets(5) man pages, as well as the file README.nss provided by the RPM package listed in section 3.1.
11. To start and stop the module, use the (service ipsec) command.
12. ikelifetime should not be larger than 1 hour.
    a. ikelifetime = 1h
13. salifetime should not be larger than 1 hour.
    a. Salifetime = 1h
14. Aggressive mode should not be used.
15. Stopping the module will zeroize the ephemeral CSPs and keys.
16. To check FIPS 140-2 module status, use:
    ```
    # ipsec status | grep fips
    000 fips mode=enabled;
    ```
17. The database for the cryptographic keys used by the Pluto Daemon must be initialized after it has been created as documented in the README.nss documentation with the following command, assuming that the database is stored in the directory /etc/ipsec.d/. modutil -fips true -dbdir /etc/ipsec.d
    a. NOTE: Encryption and decryption of data is done implicitly when the Pluto is asked to set up a new Security Association.
18. For proper operation of the in-module integrity verification, the prelink has to be disabled. For this purpose, Libreswan installs a libreswan-prelink.conf file in % {_sysconfdir}/prelink.conf.d/ that instructs prelink to skip /usr/libexec/ipsec.

10.1.1 Configuration Changes and FIPS Capable Environment

Use caution whenever making configuration changes that could potentially prevent access to the /proc/sys/crypto/fips_enabled flag (fips=1). If the file does not contain “1”, the operating environment is not configured to support FIPS and the module will not operate as a FIPS validated module.

10.2 Handling Self-Test Errors

OpenSSL and NSS self-test failures and Libreswan power-up self-tests may prevent Libreswan from operating. See the Guidance section in the OpenSSL and NSS Security Policies for instructions on handling OpenSSL or NSS self-test failures.

Power-up self-test errors are non-fatal errors that transition the module into an error state. The application must be restarted or reinstalled to recover from these errors. Libreswan outputs NSS error codes that can be used to determine the cause of the errors. Also, the user of the module will know that the IKE KDF KAT failed by the module restricting usage of IKE protocol from any calling application.

In the case of integrity test failure, Libreswan enters an error state and outputs the following error: FIPS HMAC integrity verification self-test FAILED.

The only recovery from this type of failure is to reinstall the Libreswan module. If you downloaded the software, verify the package hash to confirm a proper download.

11. Mitigation of Other Attacks

The Module does not mitigate against any attacks.

Acronyms, Terms and Abbreviations

Term	Definition
AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CCCS	Canadian Centre for Cyber Security
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
DH	Diffie-Hellman
DHE	Diffie-Hellman Ephemeral
DRBG	Deterministic Random Bit Generator
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
GCM	Galois Counter Mode
HMAC	(Keyed) Hash Message Authentication Code
IKE	Internet Key Exchange
KAT	Known Answer Test
KDF	Key Derivation Function
NIST	National Institute of Standards and Technology
NSS	Network Security Services
PAA	Processor Algorithm Acceleration
PKCS	Public Key Cryptographic Standard
POST	Power On Self-Test
PRF	Pseudo Random Function
PUB	Publication
SA	Security Associations
SHA	Secure Hash Algorithm
TLS	Transport Layer Security

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.

[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 800-38A, Recommendation for Block Cipher Modes of Operation: Methods and Techniques, http://csrc.nist.gov/publications/PubsFIPS.html
[9] NIST SP 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, http://csrc.nist.gov/publications/PubsFIPS.html
[10] NIST SP 800-38F, Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping, http://csrc.nist.gov/publications/PubsFIPS.html
[11] NIST SP 800-56Arev3, Recommendation for Pair-Wise Key Establishment Schemes using Discrete Logarithm Cryptography (Revised), https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf
[12] NIST SP 800-67 Revision 1, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, http://csrc.nist.gov/publications/PubsFIPS.html
[13] NIST SP 800-90A, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, http://csrc.nist.gov/publications/PubsFIPS.html
[14] Oasis, "PKCS #11 v2.40: Cryptographic Token Interface Base Specification", 2015.
[15] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems", CRYPTO '96, Lecture Notes In Computer Science, Vol. 1109, pp. 104-113, Springer-Verlag, 1996. http://www.cryptography.com/timingattack/
[16] D. Boneh and D. Brumley, "Remote Timing Attacks are Practical", http://crypto.stanford.edu/~dabo/abstracts/ssl-timing.html
[17] C. Percival, "Cache Missing for Fun and Profit", http://www.daemonology.net/papers/htt.pdf
[18] N. Ferguson and B. Schneier, Practical Cryptography, Sec. 16.1.4 "Checking RSA Signatures", p. 286, Wiley Publishing, Inc., 2003.