Using a FIPS 140-2 Enabled System in Oracle® Solaris 11.3

October 2019

Part No: E54966

This article describes how to configure an Oracle Solaris system to provide FIPS 140-2 Level 1 cryptography to kernel-level and user-level consumers of cryptography, for example, Kerberos, Secure Shell, and the Apache HTTP Server. It describes how to enable the providers and the consumers, and includes an example of enabling Secure Shell and the Apache HTTP Server to run in FIPS 140-2 mode.

Introduction to FIPS 140-2 Level 1 Cryptography in Oracle Solaris

In August 2016, the U.S. National Institute of Standards and Technology (NIST) issued two certificates that validate the Cryptographic Framework feature of Oracle Solaris to the FIPS 140-2 Level 1 standard. The Oracle Solaris certificates are numbered 2698 and 2699, and are based on the Oracle Solaris 11.3 SRU 5.6 release.

The OpenSSL module that runs on Oracle Solaris 11.3 was validated for FIPS 140-2 in November 2013 and issued certificate 1747. Any application that uses a FIPS 140-2 validated OpenSSL for its cryptography can use this module. For links to the certificates, see "FIPS 140-2 Level 1 Guidance Documents for Oracle Solaris Systems" on page 16.

FIPS 140-2, a U.S. Federal Information Processing Standard, is a requirement for many regulated industries and U.S. government agencies that process sensitive but unclassified information. The aim of FIPS 140-2 is to provide a degree of assurance that the system has implemented the cryptography correctly. Providing FIPS 140-2 Level 1 cryptography on a computer system is called "running in FIPS 140-2 mode".

Applications and FIPS 140-2

A system that is running in FIPS 140-2 mode has enabled at least one provider of FIPS 140-2 cryptography. Some applications (consumers) call FIPS 140-2 cryptography automatically, for example, the `passwd` command. Some applications call FIPS 140-2 cryptography providers dynamically, for example, OpenSSH. Other applications run in FIPS 140-2 mode when their provider is enabled and the administrator has configured the application to use FIPS 140-2 cryptography only, for example, Kerberos, IPsec, SunSSH, and the Apache HTTP Server.

Changes in the FIPS 140-2 August 2016 Validation From the December 2013 Validation

Between December 2013 and August 2016, NIST updated FIPS 140-2 cryptography and hardware requirements. These updates changed the validation status of several items in the Cryptographic Framework feature of Oracle Solaris.

The following mechanisms have a status change in the August 2016 FIPS 140-2 validation:

- SHA512/224 is validated.
- SHA512/256 is validated.
- SHA1 and HMAC-SHA1 from `libucrypto` are validated.

- SHA1 and HMAC-SHA1 from the PKCS #11 softtoken store are not validated.
- AES-GMAC is not validated.

Software validation is no longer tied to particular hardware, as it was in the December 2013 certificates. For the list of approved hardware, see "Oracle Solaris System Hardware Validated for FIPS 140-2" on page 17.

Enabling FIPS 140-2 Providers on an Oracle Solaris System

Because FIPS 140-2 provider modules are CPU intensive, they are not enabled by default. As the administrator, you are responsible for enabling the providers in FIPS 140-2 mode and configuring consumers.

The Oracle Solaris OS offers two providers of cryptographic algorithms that are validated for FIPS 140-2 Level 1:

- The Cryptographic Framework feature of Oracle Solaris is the central cryptographic store on an Oracle Solaris system and provides two FIPS 140-2 modules. The *userland* module supplies cryptography for applications that run in user space and the *kernel* module provides cryptography for kernel-level processes. Both modules can leverage the algorithm acceleration from SPARC and x86 processors when available.
  - The Oracle Solaris Userland Cryptographic Framework module provides cryptography for any application that calls into it. The module provides encryption, decryption, hashing, secure random number generation, signature generation and verification, certificate generation and verification, message authentication functions, and key pair generation for RSA and DSA. User-level applications that call into the userland Cryptographic Framework run in FIPS 140-2 mode, for example, the `passwd` command and IKEv2.
  - The Oracle Solaris Kernel Cryptographic Framework module provides cryptography for the kernel module. The module provides encryption, decryption, hashing, secure random number generation, signature generation and verification, and message authentication functions. Kernel-level consumers, for example, Kerberos and IPsec, use proprietary APIs to call into the kernel Cryptographic Framework.
- The OpenSSL object module provides cryptography for all consumers whose code supports FIPS 140-2.

  OpenSSL is the Open Source toolkit for the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols, and provides a cryptography library.

  In Oracle Solaris, the FIPS 140-2 capable OpenSSL module supports two frequently used applications:
  - Secure Shell – Both the OpenSSH and the SunSSH implementations are supported.
  - Apache HTTP Server Version 2.4 – Can also use the Cryptographic Framework to run in FIPS 140-2 mode.

    Apache HTTP Server Version 2.2 must use the Cryptographic Framework as its FIPS 140-2 provider. To run Version 2.2 in FIPS 140-2 mode, use the PKCS #11 engine rather than OpenSSL.

How to Enable the FIPS 140-2 Providers in Oracle Solaris

For an example of enabling the providers in FIPS 140-2 mode and enabling applications to use them, see "Example of Running in FIPS 140-2 Mode on an Oracle Solaris 11.3 SRU 5.6 System" on page 8.

- To run the Cryptographic Framework in FIPS 140-2 mode, see "How to Create a Boot Environment With FIPS 140-2 Enabled" in *Managing Encryption and Certificates in Oracle Solaris 11.3*.

- To run OpenSSL in FIPS 140-2 mode, see "OpenSSL and Oracle Solaris" in *Managing Encryption and Certificates in Oracle Solaris 11.3*.

About the Cryptographic Framework in FIPS 140-2 Mode

The Cryptographic Framework implements many cryptographic algorithms with varying key lengths. Each variant of an algorithm is called a *mechanism*. Not all mechanisms are validated for FIPS 140-2.

When running in FIPS 140-2 mode, the userland Cryptographic Framework does not enforce the use of FIPS 140-2 validated algorithms. This design choice enables you to apply your own security policy.

Tip - To accommodate a legacy system, non-compliant applications, or problem resolution, you can leave all Cryptographic Framework algorithms enabled. For strict enforcement of FIPS 140-2 mode, you should disable non-FIPS 140-2 algorithms in the Cryptographic Framework. For an example, see the final steps in "Example of Running in FIPS 140-2 Mode on an Oracle Solaris 11.3 SRU 5.6 System" on page 8.

After enabling the providers in FIPS 140-2 mode, you must configure applications and programs to use FIPS 140-2 algorithms.

The `cryptoadm` and `pktool` commands list the algorithms that the Cryptographic Framework supports.

- To display a complete list of cryptographic mechanisms, use the `cryptoadm list -vm` command. See the `cryptoadm`(1M) man page.
- To display the list of curves for ECC algorithms, use the `pktool gencert listcurves` command. See the `pktool`(1) man page.

  For information about ECC curves in Oracle Solaris that are FIPS 140-2 validated for Oracle Solaris, see "FIPS 140-2 Algorithms in the Cryptographic Framework" on page 14.
- For information about the FIPS 140-2 algorithms that are validated for the Cryptographic Framework, review the Oracle Solaris security policies that are listed in "FIPS 140-2 Level 1 Guidance Documents for Oracle Solaris Systems" on page 16. The supported algorithms differ slightly between the kernel Cryptographic Framework and the userland Cryptographic Framework.

About OpenSSL in FIPS 140-2 Mode in Oracle Solaris

When running in FIPS 140-2 mode, OpenSSL enforces the use of FIPS 140-2 validated algorithms. Therefore, applications that use OpenSSL in FIPS 140-2 mode cannot access invalid algorithms.

For more information and examples, see the following:

- "OpenSSL and Oracle Solaris" in *Managing Encryption and Certificates in Oracle Solaris 11.3*
- OpenSSL on Oracle Solaris 11.2 (`https://blogs.oracle.com/solaris/openssl-on-oracle-solaris-112-v2`)
- `openssl`(5) man page

Note - For an example of configuring OpenSSL in FIPS 140-2 mode, see "Example of Running in FIPS 140-2 Mode on an Oracle Solaris 11.3 SRU 5.6 System" on page 8.

Hardware Acceleration and FIPS 140-2 Performance

For best performance, consumers of FIPS 140-2 providers should use hardware-accelerated cryptography where possible. The Cryptographic Framework runs with hardware acceleration in FIPS 140-2 mode on the systems listed in "Oracle Solaris System Hardware Validated for FIPS 140-2" on page 17.

To get hardware acceleration on a SPARC T4 or SPARC T5 server when running OpenSSL in FIPS 140-2 mode, use the `pkcs11` engine.

Note - On SPARC systems, the FIPS 140-2 version of OpenSSL that is included in the Oracle Solaris 11.3 release takes advantage of assembly language optimizations but not hardware acceleration. On Intel systems, the FIPS 140-2 version of OpenSSL that is included in Oracle Solaris 11.3 takes advantage of AES-NI hardware acceleration and assembly language optimizations.

For more information, see "SPARC Acceleration of Optimized Cryptographic Functions" in *Managing Encryption and Certificates in Oracle Solaris 11.3*. For an example, see "Example of Running in FIPS 140-2 Mode on an Oracle Solaris 11.3 SRU 5.6 System" on page 8.

Enabling FIPS 140-2 Consumers on an Oracle Solaris System

To run in FIPS 140-2 mode, applications on your FIPS 140-2-enabled system must use algorithms that the U.S. government has validated for FIPS 140-2 mode on Oracle Solaris. When FIPS 140-2 providers are enabled, some consumers use FIPS 140-2 algorithms by default, for example, the `passwd` command. Other consumers require configuration to use only FIPS 140-2 algorithms.

As an administrator, you are responsible for configuring consumers to use FIPS 140-2 algorithms that are validated for Oracle Solaris and avoiding invalid algorithms. Follow these guidelines:

- Avoid an algorithm that is available on Oracle Solaris but is not part of the FIPS 140-2 validation for Oracle Solaris, for example, two-key Triple DES.
- Avoid an algorithm that is part of the FIPS 140-2 certificate for Oracle Solaris but that has a key length shorter than FIPS 140-2 requires, for example, 1024-bit RSA.
- Avoid an algorithm that is part of the FIPS 140-2 certificate for Oracle Solaris but the consumer cannot use it, for example, Elliptic-Curve Cryptography (ECC) over a Koblitz curve for IKEv2. IKEv2 supports ECC over primes only.
- Avoid all algorithms that are not part of the FIPS 140-2 certificate for Oracle Solaris but are in the Cryptographic Framework, for example, the `MD5` symmetric key algorithm and weaker versions of other symmetric algorithms.
- Specify FIPS 140-2 algorithms only, even when other algorithms are available to consumers. Many consumers fall in this category.

Note - Any application that cannot use FIPS 140-2 validated algorithms, such as the Internet Key Exchange Protocol Version 1 (IKEv1), should not be run on a FIPS 140-2 system.

Apache HTTP Server as a FIPS 140-2 Consumer

Oracle Solaris 11.3 provides two versions of the Apache HTTP Server. Version 2.4 installs as the package `pkg:/web/server/apache-24` and Version 2.2 installs as the package `pkg:/web/server/apache-22`. To run in FIPS 140-2 mode, Version 2.4 can use either the FIPS 140-2 OpenSSL provider or the PKCS #11 engine option. Version 2.2 must use the PKCS #11 engine option, which is the Cryptographic Framework.

Note - You can run both versions of the web server in FIPS 140-2 mode if you configure each version to listen on a different port.

You can use either the Cryptographic Framework (`pktool gencert` command) or the FIPS 140-2 version of OpenSSL (`openssl -newkey` command) to generate the web server certificate.

For the configuration steps, see "Example of Running in FIPS 140-2 Mode on an Oracle Solaris 11.3 SRU 5.6 System" on page 8.

See also:

- `openssl`(1openssl) man page
- `openssl`(5) man page
- "How to Configure an Apache 2.2 Web Server to Use the SSL Kernel Proxy" in *Securing the Network in Oracle Solaris 11.3*
- `ksslcfg`(1M) man page

Secure Shell as a FIPS 140-2 Consumer

Oracle Solaris 11.3 provides two implementations of Secure Shell: OpenSSH and SunSSH. Both implementations can run in FIPS 140-2 mode.

For OpenSSH, you must install the `openssh` package. When the FIPS 140-2 version of OpenSSL is enabled, OpenSSH runs in FIPS 140-2 mode.

For SunSSH, you must configure it to run in FIPS 140-2 mode. For more information, see "SunSSH and FIPS 140-2" in *Managing Secure Shell Access in Oracle Solaris 11.3*, which includes the list of validated FIPS 140-2 algorithms.

For a sample configuration, see "Example of Running in FIPS 140-2 Mode on an Oracle Solaris 11.3 SRU 5.6 System" on page 8.

See also:

- `sshd`(1M) and `ssh`(1) man pages
- `sshd_config`(4) and `ssh_config`(4) man pages
- `ssh-keygen`(1) man page

IPsec and IKEv2 as FIPS 140-2 Consumers

IP Security Architecture (IPsec) provides cryptographic protection for IP packets in IPv4 and IPv6 networks. Internet Key Management (IKE) provides automated key management for IPsec. In Oracle Solaris, IPsec is a consumer of the kernel Cryptographic Framework and IKE version 2 (IKEv2) is a consumer of the userland Cryptographic Framework. As the IPsec and IKE administrator, you are responsible for using IKEv2 with IPsec and for choosing FIPS 140-2 algorithms that are validated for Oracle Solaris.

Note - IKEv1 does not use cryptographic algorithms that are validated for FIPS 140-2. Therefore, IKEv1 should not be used on a system that is running in FIPS 140-2 mode.

To ensure that IPsec and IKEv2 run in FIPS 140-2 mode, you must specify FIPS 140-2 algorithms after booting into an Oracle Solaris system where FIPS 140-2 mode is enabled. You are responsible for using FIPS 140-2 algorithms in IPsec and IKEv2 configuration files, and for key types and hash types for certificates and certificate signing requests (CSRs) that you generate with the `ikev2cert` command. For a summary list, see "IPsec and FIPS 140-2" in *Securing the Network in Oracle Solaris 11.3*. For the full list of validated algorithms, review "FIPS 140-2 Algorithms in the Cryptographic Framework" on page 14.

See also:

- "How to Use IPsec to Protect Web Server Communication With Other Servers" in *Securing the Network in Oracle Solaris 11.3*
- "How to Configure IKEv2 With Self-Signed Public Key Certificates" in *Securing the Network in Oracle Solaris 11.3*
- "How to Generate and Store Public Key Certificates for IKEv2 in Hardware" in *Securing the Network in Oracle Solaris 11.3*
- `ipsecconf`(1M), `ikev2cert`(1M), `ikev2.config`(4), and `pktool`(1) man pages

Kerberos as a FIPS 140-2 Consumer

The Kerberos client installs as the package `pkg:/security/kerberos-5`, and the KDC manager installs as the package `pkg:/security/kerberos-5/kdc`. As the Kerberos administrator, you are responsible for enabling Kerberos servers, the Kerberos database, and Kerberos clients to use the FIPS 140-2 algorithm that is validated for Oracle Solaris.

Several Kerberos configuration files specify the encryption types to use for the KDC database and Kerberos clients. In those files, you must configure Kerberos to use FIPS 140-2 encryption types only and to disallow weak keys.

For the procedure, see "How to Configure Kerberos to Run in FIPS 140-2 Mode" in *Managing Kerberos and Other Authentication Services in Oracle Solaris 11.3*.

See also:

- `kdc.conf`(4) and `krb5.conf`(4) man pages
- `kdb5_util`(1M), and `krb5kdc`(1M) man pages

Key Management Framework as a FIPS 140-2 Consumer

The Key Management Framework (KMF) manages cryptographic keys and cryptographic policy in Oracle Solaris. `pktool` is the KMF command for creating symmetric and asymmetric keys. As the KMF administrator, you are responsible for choosing FIPS 140-2 algorithms that are validated for Oracle Solaris. See examples in "How to Create a Certificate by Using the pktool gencert Command" in *Managing Encryption and Certificates in Oracle Solaris 11.3* and the `pktool`(1) man page.

`passwd` Command as a FIPS 140-2 Consumer

The `passwd` command is a consumer of the userland Cryptographic Framework. Two configuration files, `/etc/security/crypt.conf` and `/etc/security/policy.conf`, determine which password hash the system uses.

The `passwd` command calls the `crypt()` function by using the PAM modules `pam_authtok_store.so.1` and `pam_unix_auth.so.1`. The `crypt()` function dynamically loads plugins from the message digest library, `libmd()`, based on entries in the `crypt.conf` file. Available plugins include `SHA256`, `SHA512`, and `MD5`. The `policy.conf` file lists the plugins that are allowed. By default, the `policy.conf` file does not allow the use of `MD5`.

Note - The cryptographic password hash policy in the `/etc/security/policy.conf` file promotes interoperability with systems that use non-FIPS 140-2 hashes. To promote FIPS 140-2 security, remove any non-FIPS 140-2 hashes from the `CRYPT_ALGORITHMS_ALLOW` entry in the `policy.conf` file.

For examples, see "Creating a Login for a Trusted User" in *Securing Users and Processes in Oracle Solaris 11.3* and "Creating a Role" in *Securing Users and Processes in Oracle Solaris 11.3*.

See also:

- `crypt`(3C) and `libmd`(3LIB) man pages
- `crypt.conf`(4) and `policy.conf`(4) man pages
- `passwd`(1) and `passwd`(4) man pages

`encrypt`, `decrypt`, `digest`, and `mac` Commands as FIPS 140-2 Consumers

The user commands `encrypt`, `decrypt`, `digest`, and `mac` are consumers of the Cryptographic Framework. The site security team should guide regular users to choose FIPS 140-2 algorithms of a validated key length.

For examples, see the following:

- "Protecting Files With the Cryptographic Framework" in *Managing Encryption and Certificates in Oracle Solaris 11.3*
- `encrypt`(1), `decrypt`(1), `digest`(1), and `mac`(1) man pages

Example of Running in FIPS 140-2 Mode on an Oracle Solaris 11.3 SRU 5.6 System

The example in this section configures an Oracle Solaris system to run Apache HTTP Server Version 2.4 in FIPS 140-2 mode. The system is a SPARC T5-2 server, which provides cryptographic acceleration in the SPARC5 processor.

The main steps are:

1. Create and boot into a BE that you will configure for FIPS 140-2 Level 1.
2. In the new BE, enable the FIPS 140-2 providers.
3. Enable two consumers, Secure Shell and Apache HTTP Server Version 2.4 in FIPS 140-2 mode.
4. Modify the `policy.conf` file to remove interoperability with systems that do not use FIPS 140-2 password hashes.
5. Reboot the BE.
6. Test.

The following example describes the detailed actions you would take to accomplish this configuration.

1. Create a BE based on your current configuration and boot it.

```
# beadm create S11.3-FIPS-140
# beadm activate S11.3-FIPS-140
# reboot
```

   The preceding command gives a useful name to the BE. The BE is not yet running in FIPS 140-2 mode.
2. In the new BE, enable the two FIPS 140-2 providers.

First, enable the Cryptographic Framework provider.

```
# cryptoadm enable fips-140
```

If the `crypto/fips-140` package is not yet installed, this command installs the package.

3. Install the OpenSSL FIPS 140-2 provider.

```
# pkg install openssl-fips-140
```

Oracle Solaris either installs the package, or indicates that it is already on your system: `No updates necessary for this image.`

4. (Optional) Verify that the two FIPS 140-2 provider packages are installed.

```
# pkg verify -v openssl-fips-140 fips-140
PACKAGE                                                                 STATUS
pkg://solaris/library/security/openssl/openssl-fips-140                 OK
pkg://solaris/crypto/fips-140                                           OK
```

Caution - Do not proceed if these packages are not installed. Install them before continuing.

5. Enable the second FIPS 140-2 provider, OpenSSL.

   a. Verify that the OpenSSL FIPS 140-2 provider is on the system.

```
# pkg mediator -a openssl
MEDIATOR     VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
openssl      vendor            vendor     default
openssl      system            system     fips-140
```

   The value `fips-140` under `IMPLEMENTATION` indicates that the OpenSSL FIPS 140-2 provider is on the system.

   b. Enable the FIPS 140-2 OpenSSL provider.

```
# pkg set-mediator -I fips-140 openssl
```

Caution - If the provider that you type as the argument to the `pkg set-mediator` command is unavailable, this BE will become unusable because critical operating system components require a working `openssl` module. For more information, see "Specifying a Default Application Implementation" in *Adding and Updating Software in Oracle Solaris 11.3*.

Troubleshooting – If the BE is unusable after this command, activate the original BE and boot into it, destroy the unusable BE, and reconfigure.

6. Enable the Secure Shell consumer to run in FIPS 140-2 mode.

   You can run either the OpenSSH or the SunSSH implementation of Secure Shell in FIPS 140-2 mode. SunSSH is running by default.

   - Enable the OpenSSH consumer in FIPS 140-2 mode.

     a. Verify that the OpenSSL provider is FIPS 140-2 capable.

```
# pkg mediator openssl
MEDIATOR     VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
openssl      vendor            vendor     fips-140
```

     b. Complete "How to Use the OpenSSH Implementation of Secure Shell" in *Managing Secure Shell Access in Oracle Solaris 11.3*.

After you enable the OpenSSH implementation and the FIPS 140-2 OpenSSL provider, OpenSSH runs in FIPS 140-2 mode dynamically.

- Configure and enable the SunSSH consumer in FIPS 140-2 mode.

  a. Add the following information to the end of the `sshd_config` and `ssh_config` files to use FIPS 140-2 mode.

  ```
  # pfedit /etc/ssh/sshd_config /etc/ssh/ssh_config
  ## This system operates in FIPS 140-2 mode. SSH in FIPS 140-2 mode
  ## cannot use the OpenSSL engine. UseOpenSSLEngine yes has no effect.
  UseFIPS140 yes
  UseOpenSSLEngine no
  ```

  b. Generate a private key in PKCS #8 format for use with SunSSH in FIPS 140-2 mode.

  Follow the instructions in How to Add a CA Cert to Oracle Solaris (`https://blogs.oracle.com/solaris/how-to-add-a-ca-cert-to-solaris-v2`).

  Then, create your private key with the `ssh-keygen` command.

  When you use the `ssh-keygen` command, the default key length is 1024, which is not a validated length. You must specify a valid key length by using the `-b` option.

7. Configure the Apache HTTP Server in FIPS 140-2 mode.

   a. Generate the web server certificate by using a FIPS 140-2 algorithm at a validated key length.

   For example, use the `pktool` command, specify an RSA SHA-384 hash, and use the default 2048-bit key length.

   ```
   # pktool gencert keystore=pkcs11 \
   > label=fipskey \
   > subject "/C=CTRY/ST=County area/L=City/CN=`hostname`" \
   > keytype=rsa hash=sha384 keylen=2048 \
   > serial 0xnnnnnnnn
   ```

   b. Create the `ssl.conf` configuration file.

   ```
   # cp /etc/apache2/2.4/samples-conf.d/ssl.conf /etc/apache2/2.4/conf.d/
   ```

   c. For clarity, comment on the use of OpenSSL for FIPS 140-2 mode.

   ```
   # pfedit /etc/apache2/2.4/conf.d/ssl.conf
   ## In Oracle Solaris 11.3, the OpenSSL
   ## module is FIPS 140-2 validated.
   SSLCryptoDevice builtin
   ```

   Note - If you were configuring Apache HTTP Server Version 2.2, the value of `SSLCryptoDevice` would be `pkcs11`.

   d. Ensure that other keying information is correctly configured for your site policy.

   ```
   # grep ^SSLCipherSuite /etc/apache2/2.4/conf.d/ssl.conf
   SSLCipherSuite AES256-SHA:AES128-SHA
   # grep ^SSLHonorCipherOrder /etc/apache2/2.4/conf.d/ssl.conf
   SSLHonorCipherOrder on
   ```

   e. Complete your site configuration of the web server.

   For example, specify the SSL protocol versions.

   ```
   # grep ^SSLProtocol /etc/apache2/2.4/conf.d/ssl.conf
   SSLProtocol all -SSLv2 -SSLv3
   ```

8. Prevent the use of a non-FIPS 140-2 password hash by removing `2a` as an allowable hash.

```
# pfedit /etc/security/policy.conf
CRYPT_ALGORITHMS_ALLOW=5,6
```

9. (Optional) Ensure that all logins use the correct hash.

   a. List all users who can log in to the BE.

```
# logins -xo -S files | grep PS
root:0:root:0:Super-User:/root:/usr/bin/bash:PS ...
testuser1:111:test:110:Tester1:/home/tester1:/usr/bin/bash:PS ...
testuser2:112:test:110:Tester2:/home/tester2:/usr/bin/bash:PS ...
admin:141:fipadm:140:FIPS 140-2 Administrator:/home/admin:/usr/bin/bash:PS ...
```

   Tip - Use the `-S ldap` option to find all users in the LDAP repository.

   b. Force each user to create a new password at login.

```
# passwd -f [-r files | ldap ] username
```

   Tip - You can write a script that forces all users to change their password at login.

10. After the consumers are configured, reboot the BE.

```
# reboot
```

11. Test the configuration.

- Verify that the providers are operating in FIPS 140-2 mode.

  The following output indicates that the Cryptographic Framework is operating in FIPS 140-2 mode.

```
# cryptoadm list fips-140
User-level providers:
=====================
/usr/lib/security/$ISA/pkcs11_softtoken: FIPS 140-2 mode is enabled.

Kernel providers:
=================
des: FIPS 140-2 mode is enabled.
aes: FIPS 140-2 mode is enabled.
ecc: FIPS 140-2 mode is enabled.
sha1: FIPS 140-2 mode is enabled.
sha2: FIPS 140-2 mode is enabled.
rsa: FIPS 140-2 mode is enabled.
swrand: FIPS 140-2 mode is enabled.

Kernel hardware providers:
========================:
n2rng: FIPS 140-2 mode is enabled.
```

  The following output indicates that OpenSSL is operating in FIPS 140-2 mode.

```
# pkg mediator openssl
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
openssl      system            system  fips-140
```

- Trace the Apache HTTP Server's cryptographic use.

a. In a terminal window, trace the OpenSSL cryptographic calls on the Apache HTTP Server Version 2.4 process.

```
# truss -w \!all -t \!all -v \!all \
  -u libcrypto::FIPS_evp_\* \
  -f /usr/apache2/2.4/bin/httpd -k start
```

Note - This command traces FIPS 140-2 envelope (`evp`) function calls to the `/usr/lib/libcrypto.so.1` library.

---

b. Send a web server request and review the output for use of the FIPS 140-2 envelope.

```
# openssl s_client -connect localhost:443 -tls1
...
GET / HTTP/1.0
...
8358/1@1: -> libcrypto:FIPS_evp_sha1()
8358/1@1: <- libcrypto:FIPS_evp_sha1() = 0xf94984b8
8358/1@1: -> libcrypto:FIPS_evp_aes_128_cbc()
8358/1@1: <- libcrypto:FIPS_evp_aes_128_cbc() = 0xf94980d8
...
```

- Test the Secure Shell login from a non-FIPS 140-2 system and a FIPS 140-2 system.
- Review the log files for Secure Shell and the Apache HTTP Server.

  Secure Shell returns errors when FIPS 140-2 algorithms are not being used.

12. (Optional) To prevent the use of non-FIPS 140-2 algorithms by all Cryptographic Framework consumers, disable the non-FIPS 140-2 mechanisms.

Tip - To implement a strict policy for Cryptographic Framework consumers, create a script that implements the policy, then create a second BE for the strict policy version of FIPS 140-2 mode.

The following set of commands prevents the use of kernel algorithms that are not validated for FIPS 140-2 mode. The list is truncated to highlight the non-FIPS 140-2 algorithm mechanisms.

```
# cryptoadm -vm
...
Kernel providers:
=================
des: CKM_DES_ECB,CKM_DES_CBC,CKM_DES3_ECB,CKM_DES3_CBC
arcfour: CKM_RC4
blowfish: CKM_BLOWFISH_ECB,CKM_BLOWFISH_CBC
camellia: CKM_CAMELLIA_ECB,CKM_CAMELLIA_CTR,CKM_CAMELLIA_CBC
md4: CKM_MD4
md5: CKM_MD5,CKM_MD5_HMAC,CKM_MD5_HMAC_GENERAL
# cryptoadm disable provider=des mechanism=CKM_DES_ECB,CKM_DES_CBC
# cryptoadm disable provider=arcfour mechanism=all
# cryptoadm disable provider=blowfish mechanism=all
# cryptoadm disable provider=camellia mechanism=all
# cryptoadm disable provider=md4 mechanism=all
# cryptoadm disable provider=md5 mechanism=all
```

The following command shows the policy for kernel Cryptographic Framework providers after you disable non-FIPS 140-2 mechanisms.

```
# cryptoadm list -p
...
des: all mechanisms are enabled, except CKM_DES_CBC,CKM_DES_ECB.
aes: all mechanisms are enabled.
arcfour: no mechanisms presented.
blowfish: all mechanisms are enabled, except CKM_BLOWFISH_ECB,CKM_BLOWFISH_CBC.
camellia: all mechanisms are enabled, except CKM_CAMELLIA_ECB,CKM_CAMELLIA_CTR,CKM_CAMELLIA_CBC.
ecc: all mechanisms are enabled.
sha1: all mechanisms are enabled.
sha2: all mechanisms are enabled.
md4: no mechanisms presented.
md5: all mechanisms are enabled, except CKM_MD5,CKM_MD5_HMAC,CKM_MD5_HMAC_GENERAL.
rsa: all mechanisms are enabled.
swrand: random is enabled.
```

To prevent the use of userland mechanisms, specify `/usr/lib/security/$ISA/pkcs11_softtoken.so` as the provider, then specify the mechanisms. To list the mechanisms in userland, use the following command:

```
# cryptoadm list -vm provider=/usr/lib/security/\$ISA/pkcs11_softtoken.so
Mechanism Name                Minimum    Maximum  ...
----------------------------- -------- ----------  ...
CKM_CAMELLIA_CBC                   16         32  ...
CKM_CAMELLIA_CBC_PAD               16         32  ...
CKM_CAMELLIA_CTR                   16         32  ...
CKM_CAMELLIA_ECB                   16         32  ...
CKM_CAMELLIA_KEY_GEN               16         32  ...
...
CKM_ECDSA                         112        571  ...
CKM_ECDSA_SHA1                    112        571  ...
CKM_ECDH1_DERIVE                  112        571  ...
```

For example, the following command disables the Camellia mechanisms in userland:

```
# cryptoadm disable provider=/usr/lib/security/\$ISA/pkcs11_softtoken.so \
>mechanism=CKM_CAMELLIA_CBC,CKM_CAMELLIA_CBC_PAD,CKM_CAMELLIA_CTR,CKM_CAMELLIA_ECB,
CKM_CAMELLIA_KEY_GEN
# cryptoadm list -p
User-level providers:
=====================
/usr/lib/security/$ISA/pkcs11_kernel.so: all mechanisms are enabled.
/usr/lib/security/$ISA/pkcs11_softtoken.so: all mechanisms are enabled,
except
 CKM_CAMELLIA_KEY_GEN,CKM_CAMELLIA_ECB,CKM_CAMELLIA_CBC,CKM_CAMELLIA_CBC_PAD,CKM_CAMELLIA_CTR.
 random is enabled.
```

Caution - Test the strict policy BE thoroughly before using it in a production environment.

13. To stop using FIPS 140-2 mode, activate the original BE and reboot.

```
# beadm activate original-BE
# reboot
```

FIPS 140-2 Algorithm Lists and Certificate References for Oracle Solaris Systems

This section lists the algorithms that can be used in FIPS 140-2 mode and the algorithms that should be avoided.

Note - These lists are provided for convenience only. The official U.S. FIPS 140-2 certification and guideline documents are the definitive source.

FIPS 140-2 Algorithms in the Cryptographic Framework

To ensure that a consumer of the Cryptographic Framework is using a FIPS 140-2 validated algorithm, choose an algorithm from the following summary of validated algorithms, modes, and key lengths.

For the definitive lists of algorithms, review the security policy references in "FIPS 140-2 Level 1 Guidance Documents for Oracle Solaris Systems" on page 16.

Note - The key length of an algorithm can be significant. Shorter key lengths might not be validated for FIPS 140-2.

- AES – With the following modes and key lengths only:
  - CBC mode – 128-bit, 192-bit, and 256-bit key lengths
  - CCM mode – 128-bit, 192-bit, and 256-bit key lengths
  - CFB mode – 128-bit key length
  - CTR mode – 128-bit, 192-bit, and 256-bit key lengths
  - ECB mode – 128-bit, 192-bit, and 256-bit key lengths
  - GCM mode – 128-bit, 192-bit, and 256-bit key lengths
  - XTS mode – 128-bit and 256-bit key lengths, for data storage only
- 3DES – In CBC and ECB modes for keying option 1.
- Diffie-Hellman – Used in key agreement, in 2048-bit to 5012-bit key lengths, userland Cryptographic Framework only.
- Elliptic-Curve Diffie-Hellman (ECDH) – Allowed for use in key agreement in 2048-bit to 5012-bit key lengths, userland Cryptographic Framework only.
- DSA – 2048-bit key length and longer.
- ECC – With the following curves only. ECC contributes to ECDSA and ECDH. The first name is the NIST name; the second name is its equivalent in Oracle Solaris.
  - P-192 – `secp192r1`
  - P-224 – `secp224r1`
  - P-256 – `secp256r1`
  - P-384 – `secp384r1`
  - P-521 – `secp521r1`
  - B-163 – `sect163r2`
  - B-233 – `sect233r1`
  - B-283 – `sect283r1`
  - B-409 – `sect409r1`
  - B-571 – `sect571r1`

    - K-163 – `sect163k1`
    - K-233 – `sect233k1`
    - K-283 – `sect283k1`
    - K-409 – `sect409k1`
    - K-571 – `sect571k1`
- HMAC SHA1 – Has no variants.
- HMAC SHA2 – 224-bit to 512-bit key lengths.
- ECDSA SHA1 – Signature verification.
- RSA – 2048-bit key length and longer, with SHA1, and SHA2 with 256-bit to 512-bit key lengths.
- SHA1 – Has no variants.
- SHA2 – 224-bit to 512-bit key lengths.
- SHA512/224 – A truncated version of SHA-512, where the initial values are generated by using the method described in Secure Hash Standard: Updated Specifications Approved and Issued as Federal Information Processing Standard (FIPS) 180-4 (`https://csrc.nist.gov/publications/detail/itl-bulletin/2012/05/secure-hash-standard-updated-specifications-approved-and-issued/final`).
- SHA512/256 – A truncated version of SHA-512, where the initial values are generated by using the method described in Secure Hash Standard: Updated Specifications Approved and Issued as Federal Information Processing Standard (FIPS) 180-4.
- `swrand` – Software entropy source the kernel Cryptographic Framework. Both kernel and userland have a NIST-approved DRBG (Deterministic Random Bit Generator). See Recommendation for Random Number Generation Using Deterministic Random Bit Generators (`https://csrc.nist.gov/publications/detail/sp/800-90a/rev-1/final`).
- `intelrd` – Hardware entropy source in the kernel Cryptographic Framework. Both kernel and userland have a NIST-approved DRBG (Deterministic Random Bit Generator). See Recommendation for Random Number Generation Using Deterministic Random Bit Generators.

The following algorithms with specified key lengths are allowed in a FIPS 140-2 configuration:

- RSA key wrapping – Key lengths longer than 112 bits are allowed.
- Diffie-Hellman key agreement – Key lengths longer than 112 bits are allowed, userland Cryptographic Framework only.
- Elliptic Curve Diffie-Hellman (ECDH) key agreement – Key lengths longer than 112 bits are allowed, userland Cryptographic Framework only.

Algorithms That Are Not Approved for FIPS 140-2 in the Cryptographic Framework

In FIPS 140-2 mode, you cannot use an algorithm from the following summarized list of algorithms even if the algorithm is implemented in the Cryptographic Framework or is a FIPS 140-2 validated algorithm for other providers.

For the definitive lists of algorithms, review the security policy references in “FIPS 140-2 Level 1 Guidance Documents for Oracle Solaris Systems” on page 16.

- Two-key Triple-DES – Also written 3DES, is a weak algorithm that provides only 80 bits of security.
- MD4 – Message Digest Algorithm 4, developed by Ronald Rivest in 1990, is a demonstrably vulnerable algorithm.
- MD5 and HMAC MD5 – Message Digest Algorithm 5 can be used in FIPS 140-2 mode with TLS only.

The MD5 algorithm, developed by Ron Rivest in 1991, produces a 128-bit hash value. MD5 is commonly used to verify data integrity. MD5 is not suitable for applications like SSL certificates or digital signatures that rely on collision resistance for digital security.

- RC4 – Also known as ARCFOUR or ARC4, RC4 is a software stream cipher that is used in Transport Layer Security (TLS) to protect Internet traffic, and in WEP to secure wireless networks. RC4 is demonstrably vulnerable when the beginning of the output keystream is not discarded or when keys are not random.
- AES – Modes not explicitly validated, such as XCBC-MAC and CTS.
- Blowfish – A symmetric key block cipher, designed in 1993 by Bruce Schneier, that is not proprietary.
- Camellia – Developed in Japan, is comparable to AES, and is designed to be suitable for both software and hardware implementations, from low-cost smart cards to high-speed network systems.
- DES – Data Encryption Standard, developed by IBM, was published as an U.S. Federal Information Processing Standard (FIPS) in 1977. In today's computing environment, its 56-bit key length is weak.
- DSA key generation – The 512-bit and 1024-bit key lengths are weak. Longer key lengths are validated for FIPS 140-2.
- DSA signature generation – The 512-bit and 1024-bit key lengths are weak. Longer key lengths are validated for FIPS 140-2.
- DSA signature verification – The 512-bit key length is weak. Longer key lengths are validated for FIPS 140-2.
- RSA signature generation – The 256-bit, 512-bit, and 1024-bit key lengths are weak. Longer key lengths are validated for FIPS 140-2.
- RSA signature verification – The 256-bit and 512-bit key lengths are weak. Longer key lengths are validated for FIPS 140-2.
- RSA key wrapping – The key lengths less than 112 bits are weak. Longer key lengths are allowed for FIPS 140-2.
- Diffie-Hellman – Key lengths less than 112 bits are weak. Longer key lengths are allowed for key agreement, userland Cryptographic Framework only.
- ECDH – Key lengths less than 112 bits are weak. Longer key lengths are allowed for key agreement, userland Cryptographic Framework only.

FIPS 140-2 Level 1 Guidance Documents for Oracle Solaris Systems

The security policies in the following table provide a complete list of cryptographic mechanisms that are validated to run in FIPS 140-2 mode on Oracle Solaris.

TABLE 1 FIPS 140-2 Certificates and Security Policies for Provider Modules in Oracle Solaris

Certificate	Provider Module	Security Policy
2698	Oracle Solaris Kernel Cryptographic Framework (https://csrc.nist.gov/projects/cryptographic-module-validation-program/Certificate/2698)	Oracle Solaris 11.3 kCF Security Policy
2699	Oracle Solaris Userland Cryptographic Framework (https://csrc.nist.gov/projects/cryptographic-module-validation-program/Certificate/2699)	Oracle Solaris 11.3 uCF Security Policy
1747	OpenSSL FIPS Object Module Version 2.0.10 (https://csrc.nist.gov/projects/cryptographic-module-validation-program/Certificate/1747)	OpenSSL FIPS Object Module 2.0 Security Policy
1051	OpenSSL FIPS Object Module Version 1.2.4 (https://csrc.nist.gov/projects/cryptographic-module-validation-program/Certificate/1051)	OpenSSL FIPS Object Module 1.2 Security Policy

The following FIPS 140-2 standard document and transitions document provide guidance about the FIPS 140-2 process and deprecated or restricted algorithms and their weaker variants:

- Security Requirements for Cryptographic Modules (`https://csrc.nist.gov/publications/detail/fips/140/2/final`)
- Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths (`https://csrc.nist.gov/publications/detail/sp/800-131a/rev-1/final`)

Oracle Solaris System Hardware Validated for FIPS 140-2

The following Oracle Solaris system hardware and processors are FIPS 140-2-compliant. All systems were validated with and without hardware acceleration.

For the definitive platform list, review the security policy references in “FIPS 140-2 Level 1 Guidance Documents for Oracle Solaris Systems” on page 16.

- Oracle SPARC T4, T5, and T7 Series Servers
- Oracle SPARC M5, M6, and M7 Series Servers
- Oracle SPARC S7 Series Servers
- Oracle Minicluster S7-2 Engineered Systems
- Oracle Netra SPARC T4-1B and T5-1B Servers
- Oracle Sun Blade X3 and X4 Series Servers
- Oracle Sun Server X3, X4, and X5 Series
- Oracle Netra Server X3-2 and X5-2
- Oracle Server X6-2 and X6-2L
- Fujitsu M10 Servers
- Fujitsu SPARC M12 Servers

Using a FIPS 140-2 Enabled System in Oracle Solaris 11.3

Part No: E54966

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Référence: E54966

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` ou le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` si vous êtes malentendant.