Using a FIPS 140 Enabled System in Oracle® Solaris 11.2

August 2014

ORACLE®

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 (SSH), and the Apache Web Server. It describes how to enable the providers and the consumers, and includes an example of enabling SSH and the Apache Web Server to run in FIPS 140 mode.

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

In December 2013, the U.S. National Institute of Standards and Technology (NIST) issued four certificates that validate the Cryptographic Framework feature of Oracle Solaris to the FIPS 140-2 Level 1 standard. The Oracle Solaris certificates are numbered 2060, 2061, 2076, and 2077, and are based on the Oracle Solaris 11.1 SRU 3 and SRU 5.5 releases. The Oracle Solaris 11.2 release in FIPS 140 mode uses the same algorithms.

The OpenSSL module that runs on Oracle Solaris 11.2 was validated for FIPS 140-2 in November 2013 and issued certificate 1747. Any application that uses OpenSSL for its cryptography can use this validated module. For links to the certificates, see "FIPS 140-2 Level 1 Certificate References for Oracle Solaris Systems" on page 15. For the Oracle Solaris 11.1 releases, the OpenSSL FIPS 140 module is private. The only application that can take advantage of it is the Solaris version of Secure Shell (SSH).

FIPS 140, 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 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 mode".

A system that is running in FIPS 140 mode has enabled at least one provider of FIPS 140 cryptography. Some applications use FIPS 140 cryptography automatically, for example the `passwd` command. Other applications must be enabled in FIPS 140 mode, for example, SSH, while other applications run in FIPS 140 mode when their provider is enabled and the application uses FIPS 140 cryptography only, for example, Kerberos, IPsec, and the Apache Web Server.

Enabling FIPS 140 Providers on an Oracle Solaris System

Oracle Solaris systems offer 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 modules. The *userland* module supplies cryptography for applications that run in user space and the *kernel* module provides cryptography for kernel-level processes.

  These library modules provide encryption, decryption, hashing, signature generation and verification, certificate generation and verification, and message authentication functions for applications. User-level applications that call into these modules run in FIPS 140 mode, for example, the `passwd`

command and IKEv2. 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 SSH and web applications.

  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, SSH and the Apache Web Server are consumers of the OpenSSL FIPS 140 module.

  Oracle Solaris ships a FIPS 140 version of OpenSSL with Oracle Solaris 11.2 that is available to all consumers but the version shipped with Oracle Solaris 11.1 is available to Solaris SSH only.

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 mode and configuring consumers.

Note - In this article, FIPS 140-validated means that you are running in FIPS 140 mode on an Oracle Solaris release that is validated by NIST. FIPS 140-approved means that the algorithms that you are using are the same as the algorithms in the FIPS 140 version, but might not be validated in an Oracle Solaris release.

How to Enable the FIPS 140 Providers in Oracle Solaris

For an example of enabling the providers in FIPS 140 mode and enabling applications to use them, see "Example of Enabling Two Applications in FIPS 140 Mode on an Oracle Solaris System" on page 8.

- To run the Cryptographic Framework in FIPS 140 mode, see "How to Create a Boot Environment with FIPS 140 Enabled" in "Managing Encryption and Certificates in Oracle Solaris 11.2 ".
- To run OpenSSL in FIPS 140 mode, see "OpenSSL Support in Oracle Solaris" in "Managing Encryption and Certificates in Oracle Solaris 11.2 ".

About the Cryptographic Framework in FIPS 140 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.

When running in FIPS 140 mode, the userland Cryptographic Framework does not enforce the use of FIPS 140-approved 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 mode, you can disable non-FIPS 140 algorithms in the Cryptographic Framework. For an example, see the final steps in "Example of Enabling Two Applications in FIPS 140 Mode on an Oracle Solaris System" on page 8.

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

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

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

  For the list of ECC curves in Oracle Solaris that are FIPS 140-validated for Oracle Solaris, see "FIPS 140 Algorithms in the Cryptographic Framework" on page 13.

- For a complete list of FIPS 140 algorithms that are validated for the Cryptographic Framework, review the Oracle Solaris security policies that are listed in “FIPS 140-2 Level 1 Certificate References for Oracle Solaris Systems” on page 15. The supported algorithms differ slightly between the kernel Cryptographic Framework and the userland Cryptographic Framework.

About OpenSSL in FIPS 140 Mode in Oracle Solaris

When running in FIPS 140 mode, OpenSSL as a FIPS 140-2 provider enforces the use of FIPS 140-validated algorithms. Therefore, the SSH consumer is prevented from using algorithms that are not validated. The Apache Web Server uses the PKCS #11 engine, so the OpenSSL module does not enforce the server's use of FIPS 140 algorithms.

For background and examples, see the following:

- “OpenSSL Support in Oracle Solaris” in “Managing Encryption and Certificates in Oracle Solaris 11.2 ”.
- OpenSSL on Oracle Solaris 11.2 (http://blogs.oracle.com/observatory/entry/openssl_on_solaris_11_2).
- `openssl`(5) man page

Note - For an example of configuring OpenSSL in FIPS 140 mode, see “Example of Enabling Two Applications in FIPS 140 Mode on an Oracle Solaris System” on page 8.

Hardware Acceleration and FIPS 140 Performance

The SPARC T4 and SPARC T5 processors on the Oracle SPARC T-Series servers provide cryptographic acceleration in the hardware, as do Intel AES-NI processors. The Cryptographic Framework was awarded FIPS 140 certificates for its use on SPARC T4 and SPARC T5 processors. OpenSSL was tested on SPARC T3 and Intel AES-NI processors for its FIPS 140 validation, but its validated version does not include inline instructions to the hardware.

Note - The OpenSSL FIPS 140 module that ships with Oracle Solaris does not include hardware-accelerated cryptography. On an Intel system, OpenSSL makes use of assembly language optimizations for FIPS 140 cryptography.

For best performance, consumers of FIPS 140 providers should use hardware-accelerated cryptography where possible. However, because the OpenSSL module in FIPS 140 mode does not include inline instructions to the hardware, you should configure the Apache Web Server to use the PKCS #11 library. SSH uses the OpenSSL builtin engine and cannot be so configured. For more information, see “Cryptographic Optimizations in SPARC T-4 Systems” in “Managing Encryption and Certificates in Oracle Solaris 11.2 ”. For an example, see “Example of Enabling Two Applications in FIPS 140 Mode on an Oracle Solaris System” on page 8.

Enabling FIPS 140 Consumers on an Oracle Solaris System

To run in FIPS 140 mode, you must configure applications on your FIPS 140-enabled system to use algorithms that the U.S. government has validated for FIPS 140 mode on Oracle Solaris. When FIPS 140

providers are enabled, some consumers use FIPS 140 algorithms by default, for example, the `passwd` command. Other consumers require configuration to use only FIPS 140 algorithms.

As an administrator, you are responsible for choosing FIPS 140 algorithms that are validated for Oracle Solaris and avoiding invalid algorithms by keeping in mind the following FIPS 140 configuration issues:

- The algorithm is part of FIPS 140 but not part of the FIPS 140 validation for Oracle Solaris, for example, two-key Triple DES.
- The algorithm is part of FIPS 140 but the key length is shorter than FIPS 140 requires, for example, 1024-bit RSA. A key length that is too short for FIPS 140 mode is the default for some commands in Oracle Solaris, for example, `pktool gencert` and `ikev2cert gencert`.
- The algorithm is part of the FIPS 140 certificate for Oracle Solaris but is not available to the consumer, for example, Elliptic-Curve Cryptography (ECC) over a Koblitz curve for IKEv2. IKEv2 supports ECC over primes.
- The algorithm is not part of FIPS 140 but is available to consumers, for example, the `MD4` symmetric key algorithm and weaker versions of other symmetric algorithms.
- The algorithm is validated for FIPS 140 use on Oracle Solaris but other algorithms are available to consumers, so as the administrator you must specify FIPS 140 algorithms only. Many consumers fall in this category.

Note - If the consumer uses a module that cannot use FIPS 140-validated algorithms (for example, Internet Key Exchange Protocol Version 1 (IKEv1)), you cannot run these consumers on a FIPS 140 system.

Apache Web Server as a FIPS 140 Consumer

The Apache Web Server V2.2 installs as the package `pkg:/web/server/apache-22`. The web server uses the OpenSSL library and can use its PKCS #11 engine option, which is the Cryptographic Framework.

You can use either the Cryptographic Framework (`pktool gencert` command) or the OpenSSL (`openssl -newkey` command) to generate the web server certificate. Certificate authentication is handled through OpenSSL by the PKCS #11 engine. The engine is validated for FIPS 140.

For the configuration steps, see “Example of Enabling Two Applications in FIPS 140 Mode on an Oracle Solaris 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 11.2 ”
- `ksslcfg`(1M) man page

Secure Shell as a FIPS 140 Consumer

Secure Shell (SSH) provides remote system access to Oracle Solaris systems at the initial installation. SSH is a consumer of the OpenSSL FIPS 140 module. As the administrator, you must explicitly enable SSH to run in FIPS 140 mode. For instructions, see “Secure Shell and FIPS 140” in “Managing Secure Shell Access in Oracle Solaris 11.2 ”. The instructions include the list of validated FIPS 140 algorithms.

Because SSH runs on the OpenSSL FIPS 140 module, if you try to use an algorithm that is not validated for the OpenSSL FIPS 140 provider, SSH will fail with an error. To configure SSH in FIPS 140 mode, see "Example of Enabling Two Applications in FIPS 140 Mode on an Oracle Solaris 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 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 algorithms that are validated for Oracle Solaris.

Note - IKE version 1, IKEv1, uses algorithms that are not validated for FIPS 140 mode and therefore IKEv1 cannot run in FIPS 140 mode.

Examples of Enabling IPsec and IKEv2 in FIPS 140 Mode

You use the `ipsecconf`, `ipseckey`, and `ikev2cert` commands with FIPs-validated algorithms to configure IPsec and IKEv2 in FIPS 140 mode.

- In the following excerpt from an `ipsecconf` file, `aes-ccm(256)` is a FIPS 140-validated algorithm:

```
{laddr machine1 raddr machine2} ipsec {encr_algs aes-ccm(256) sa shared}
```

- The following excerpt from an `ikev2cert` command generates a certificate request with the FIPS 140-validated ECC algorithm, using curve `secp521r1` and hash `sha512`:

```
# ikev2cert gencsr label=FIPSokcsr \
subject="C=Country, O=Company\, Inc., OU=CompanyServer, CN=Server" \
keytype=ec curve=secp521r1 hash=sha512 \
outcsr=/tmp/FIPSokcsr
```

- In the following excerpt from an `ikev2.config` file, the AES algorithms in CBC mode with key lengths from 192 to 256 and `sha384` are FIPS 140-validated algorithms:

```
ikesa_xform { encr_alg aes(192..256) auth_alg sha384 dh_group 20 }
```

See also:

- "How to Use IPsec to Protect Web Server Communication With Other Servers" in "Securing the Network in Oracle Solaris 11.2 "
- "How to Configure IKEv2 With Self-Signed Public Key Certificates" in "Securing the Network in Oracle Solaris 11.2 "
- "How to Generate and Store Public Key Certificates for IKEv2 in Hardware" in "Securing the Network in Oracle Solaris 11.2 "

- `ipsecconf`(1M), `ikev2cert`(1M), `ikev2.config`(4), and `pktool`(1) man pages

Kerberos as a FIPS 140 Consumer

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

Examples of Enabling Kerberos in FIPS 140 Mode

Several Kerberos configuration files specify the encryption types to use for the KDC database and Kerberos clients. To satisfy FIPS 140 requirements, you must specify the `des3-cbc-sha1` encryption type. This type is not the default.

To limit all transactions to `des3-cbc-sha1`, you specify that the KDC and Kerberos clients accept that mechanism only.

- In the `[realms]` section of the `/etc/krb5/kdc.conf` file, set the master key type for the KDC database:

  ```
  master_key_type = des3-cbc-sha1-kd
  ```

  Because you can also set encryption by running a command, the configuration files should prevent the use of a non-FIPS 140 algorithm argument to a command.

  ```
  supported_enctypes = des3-cbc-sha1-kd:normal
  ```

- In the `[libdefaults]` section of the `/etc/krb5/krb5.conf` file, limit the encryption types:

  ```
  default_tgs_enctypes = des3-cbc-sha1-kd
  default_tkt_enctypes = des3-cbc-sha1-kd
  permitted_enctypes = des3-cbc-sha1-kd
  ```

  For clarity, explicitly forbid weak encryption types:

  ```
  allow_weak_enctypes = false
  ```

For configuration examples, see Chapter 4, “Configuring the Kerberos Service,” in “Managing Kerberos and Other Authentication Services in Oracle Solaris 11.2 ”.

See also:

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

Key Management Framework as a FIPS 140 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 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 11.2 ” and the `pktool`(1) man page.

`passwd` Command as a FIPS 140 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. Among the plugins are the `SHA256`, `SHA512`, and `MD5` password hash algorithms. The `policy.conf` file lists the password hashes from the `crypt.conf` file that are in effect on the system. By default, the `policy.conf` file does not allow the use of the `MD5` password hash.

Note - The cryptographic password hash policy in the `/etc/security/policy.conf` file promotes interoperability with systems that use Blowfish as a password hash. To promote FIPS 140 security, remove the Blowfish algorithm (`2a`) from the `CRYPT_ALGORITHMS_ALLOW=2a,5,6` entry in the `policy.conf` file.

For examples, see "Creating a Login for a Trusted User" in "Securing Users and Processes in Oracle Solaris 11.2 " and "Creating a Role" in "Securing Users and Processes in Oracle Solaris 11.2 ".

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 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 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 11.2 "
- `encrypt`(1), `decrypt`(1), `digest`(1), and `mac`(1) man pages

Example of Enabling Two Applications in FIPS 140 Mode on an Oracle Solaris System

The example in this section configures an Oracle Solaris system to run SSH and the Apache Web Server in FIPS 140 mode. The system is an Oracle SPARC T5-2, so the Cryptographic Framework takes advantage of cryptographic acceleration on the SPARC T5 processor.

Note - If you have a strict requirement to use only FIPS 140-2 validated cryptography, you must be running the Oracle Solaris 11.1 SRU 5.5 release or the Oracle Solaris 11.1 SRU 3 release. Oracle completed a FIPS 140-2 validation against the Cryptographic Framework in these two specific releases. Oracle Solaris 11.2 builds on this validated foundation and includes software improvements that address performance, functionality, and reliability. Whenever possible, you should configure Oracle Solaris 11.2 in FIPS 140-2 mode to take advantage of these improvements.

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 providers.
3. Enable two consumers, SSH and the Apache Web Server.
4. Modify the `policy.conf` file to remove interoperability with systems that do not use FIPS 140 password hashes.
5. Boot into the FIPS 140 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 S11.2-FIPS-140
# beadm activate S11.2-FIPS-140
# reboot
```

2. In the new BE, enable FIPS 140 mode in the Cryptographic Framework.

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

3. Enable FIPS 140 mode in the OpenSSL module.

Note - If you were configuring an Oracle Solaris 11.1 SRU 3 or SRU 5.5 system, you would skip this step. You cannot enable the OpenSSL provider in Oracle Solaris 11.1.

   a. Ensure that the OpenSSL FIPS 140 module is on the system.

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

Caution - If you switched to the OpenSSL module and it was not on the system, the system might become unusable.

   b. Enable the FIPS 140 OpenSSL provider.

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

4. Configure and enable the SSH consumer in FIPS 140 mode.

   These steps work for Oracle Solaris 11.2, Oracle Solaris 11.1 SRU 5.5, and Oracle Solaris 11.1 SRU 3.

   a. Configure the `sshd_config` and `ssh_config` files to use FIPS 140 mode.

      Add the following information to the end of the files:

```
# pfedit /etc/ssh/sshd_config /etc/ssh/ssh_config
```

```
## This machine operates in FIPS 140 mode. SSH in FIPS 140 mode cannot
## use the OpenSSL engine. UseOpenSSLEngine yes has no effect.
UseFIPSmode yes
UseOpenSSLEngine no
```

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

Follow the instructions in How to Set Up X.509 for Secure Shell on Oracle Solaris 11 (http://www.oracle.com/technetwork/articles/servers-storage-admin/howto-setup-x509-sunssh-1929594.html). 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.

5. Configure the Apache Web Server to use FIPS 140 cryptography.

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

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

```
# 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/apache2/2.2/samples-conf.d/ssl.conf /etc/apache2/2.2/conf.d/
```

c. For clarity, comment on the use of the PKCS #11 engine.

```
# pfedit /etc/apache2/2.2/conf.d/ssl.conf
## Enable Solaris crypto framework
## This machine operates in FIPS 140 mode.
## In Oracle Solaris, use the pkcs11 engine
## because the engine is FIPS 140-validated.
SSLCryptoDevice pkcs11
```

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

```
# grep ^SSLCipherSuite /etc/apache2/2.2/conf.d/ssl.conf
SSLCipherSuite AES256-SHA:AES128-SHA
# grep ^SSLHonorCipherOrder /etc/apache2/2.2/conf.d/ssl.conf
SSLHonorCipherOrder on
```

e. Complete your site configuration of the web server.

For example, on an Oracle Solaris 11.2 system, specify the SSL protocol versions.

```
# grep ^SSLProtocol /etc/apache2/2.2/conf.d/ssl.conf
SSLProtocol all -SSLv2 -SSLv3
```

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

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

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

```
# reboot
```

8. Test the configuration.

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

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

```
# cryptoadm list fips-140
```

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

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

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

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

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

- Create and change several passwords, then verify that the correct hash was used.

```
# passwd admin
New Password: xxxxxxxx
Re-enter Password: xxxxxxxx
# grep admin /etc/shadow
admin:$5$.....::::::
```

The `5` at the beginning of the `admin` entry indicates that the `SHA256` password hash was used.

- Trace the Apache Web Server's cryptographic use.
  a. In a terminal window, trace the Apache Web Server cryptographic calls.

Note - To truss all PKCS #11 library calls, use `-u libpkcs11:`.

```
# truss -w \!all -t \!all -v \!all \
-u libpkcs11::C_GenerateRandom \
-u libpkcs11::C_EncryptUpdate \
-u libpkcs11::C_Decrypt \
-u libpkcs11::C_DigestUpdate \
-f /usr/apache2/2.2/bin/httpd -k start
```

b. Send a web server request and review the output for use of the PKCS #11 engine.

```
# openssl s_client -connect localhost:443 -tls1
...
GET / HTTP/1.0
...
/** PKCS #11 engine sample output **/
27435/1@1: -> libpkcs11:C_EncryptUpdate(0x1087f58, 0x1802198, 0x140, 0x1802198)
27435/1@1: <- libpkcs11:C_EncryptUpdate() = 0
27435/1@1: -> libpkcs11:C_DigestUpdate(0x1087f18, 0xffbff25c, 0xd, 0xfe178000)
27435/1@1: <- libpkcs11:C_DigestUpdate() = 0
...
```

- Test SSH login from a non-FIPS 140 system and a FIPS 140 system.
- Review the log files for SSH and the Apache Web Server.

  SSH returns errors when FIPS 140 algorithms are not being used.

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

The following set of commands prevents the use of kernel algorithms that are not validated for FIPS 140 mode.

```
# cryptoadm -vm      /** truncated list shows only non-FIPS 140 algorithm mechanisms **/
...
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_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 Cryptographic Framework kernel providers after you disable non-FIPS 140 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_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. For example, the following command disables the Camellia mechanisms in userland:

```
# cryptoadm disable provider=/usr/lib/security/\$ISA/pkcs11_softtoken.so \
      >    mechanism=CKM_CAMELLIA_ECB,CKM_CAMELLIA_CBC,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_CBC,CKM_CAMELLIA_ECB. random is enabled.
```

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

10. To stop using FIPS 140 mode, activate the original BE and reboot.

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

FIPS 140 Algorithm Lists and Certificate References for Oracle Solaris Systems

This section lists the algorithms that can be used in FIPS 140 mode and the algorithms that should be avoided. The lists are provided for convenience only. The official U.S. FIPS 140 certification and guideline documents are the definitive source.

FIPS 140 Algorithms in the Cryptographic Framework

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

For the definitive lists of algorithms, study the security policy references in “FIPS 140-2 Level 1 Certificate References for Oracle Solaris Systems” on page 15.

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

- 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.
  - GMAC mode – 128-bit, 192-bit, and 256-bit key lengths.
  - XTS mode – 256-bit and 512-bit key lengths, kernel Cryptographic Framework 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.
- DSA – 2048-bit key length and longer.
- ECC – With the following curves only. 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
- Elliptic-Curve Diffie-Hellman – Used in key agreement, in 2048-bit to 5012-bit key lengths, userland Cryptographic Framework only.
- HMAC SHA1 – Has no variants.
- HMAC SHA2 – 224-bit to 512-bit key lengths.
- 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.
- `swrand` – Random number generator in kernel Cryptographic Framework. Userland has a FIPS 186-2 random number generator.

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

In FIPS 140 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-validated algorithm for other products.

For the definitive lists of algorithms, study the security policy references in "FIPS 140-2 Level 1 Certificate References for Oracle Solaris Systems" on page 15.

- Two-key Triple-DES – A weak algorithm that provides only 80 bits of security.
- SHA512/224 – A truncated version of SHA-512, where the initial values are generated by using the method described in ITL BULLETIN FOR MAY 2012 (http://csrc.nist.gov/publications/nistbul/may-2012_itl-bulletin.pdf).
- SHA512/256 – A truncated version of SHA-512, where the initial values are generated by using the method described in ITL BULLETIN FOR MAY 2012 (http://csrc.nist.gov/publications/nistbul/may-2012_itl-bulletin.pdf).
- MD4 – Message Digest Algorithm 4, developed by Ronald Rivest in 1990, is a demonstrably vulnerable algorithm.
- MD5 – Message Digest Algorithm 5 can be used in FIPS 140 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 is a software stream cipher that is used in Transport Layer Security (TLS) to protect Internet traffic, and 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.

- 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.
- Blowfish – A symmetric key block cipher, designed in 1993 by Bruce Schneier, that is not proprietary.
- AES XCBC-MAC – A Message Authentication Code (MAC) implementation for IPsec that is designed to provide security for packets that vary in length, such as typical IP packets.
- DSA key generation – The 512-bit and 1024-bit key lengths are weak. Longer key lengths are validated for FIPS 140.
- DSA signature generation – The 512-bit and 1024-bit key lengths are weak. Longer key lengths are validated for FIPS 140.
- DSA signature verification – The 512-bit key length is weak. Longer key lengths are validated for FIPS 140.
- RSA signature generation – The 256-bit, 512-bit, and 1024-bit key lengths are weak. Longer key lengths are validated for FIPS 140.
- RSA signature verification – The 256-bit and 512-bit key lengths are weak. Longer key lengths are validated for FIPS 140.
- Diffie-Hellman – The 64-bit, 128-bit, 256-bit, 512-bit, and 1024-bit key lengths are weak. Longer key lengths are validated for FIPS 140.
- Elliptic Curve Diffie-Hellman – The 112-bit to 223-bit key lengths are weak. Longer key lengths are validated for FIPS 140, but deprecated.

FIPS 140-2 Level 1 Certificate References for Oracle Solaris Systems

The security policies for the FIPS 140 providers on an Oracle Solaris system document the module specifications and interfaces and provide a complete list of cryptographic mechanisms that are validated to run in FIPS 140 mode.

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

Certificate	Provider Module	Security Policy
2077	Oracle Solaris Userland Cryptographic Framework	http://www.oracle.com/technetwork/topics/security/140sp2077-2133432.pdf
2076	Oracle Solaris Userland Cryptographic Framework with SPARC T4 and SPARC T5	http://www.oracle.com/technetwork/topics/security/140sp2076-2133433.pdf
2061	Oracle Solaris Kernel Cryptographic Framework	http://www.oracle.com/technetwork/topics/security/140sp2061-2082028.pdf
2060	Oracle Solaris Kernel Cryptographic Framework with SPARC T4 and SPARC T5	http://www.oracle.com/technetwork/topics/security/140sp2060-2082025.pdf
1747	OpenSSL FIPS Object Module Version 2	http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1747.pdf
1051	OpenSSL FIPS Object Module Version 1.2	http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1051.pdf

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

- NIST Standards: FIPS PUB 140-2 (http://csrc.nist.gov/groups/STM/cmvp/standards.html)
- Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths (http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf)

ORACLE