FIPS Administration Tools Crypto Officer Role Guide v1.1

© Copyright 2012 Apple Inc.

Contents

© Copyright 2012 Apple Inc.

Where to obtain the FIPS Administration Tools installer

The Crypto Officer can obtain the FIPS Administration Tools installer at Apple's software download website (http://support.apple.com/kb/DL1518).

How to install the FIPS Administration Tools

Once the Crypto Officer has obtained the FIPS Administration Tools installer, login to the target computer system where the tools will be installed with an administrator account.

Note: After any OS X Lion System and/or Security update, the Crypto Officer must either 1) run the FIPS Administration installer again or 2) run the `FIPSPerformSelfTest create` command. This step is necessary to update the Error Detection Code (EDC) for the integrity validation of the PRNG during the Power On Self Test.

1. Double click on the FIPS Administration Installer Package
2. Click Continue after reading the information on the Introduction page.
3. Click Continue after reading the information on the Read Me page. You can also Print or Save the information on this page as needed.
4. Click Continue after reading the Software License Agreement on the License page. You can also Print or Save the information on this page as needed.
5. Click Agree if you agree with the terms of the software license. Otherwise click Disagree and the installer will exit.
6. Select the drive to install the FIPS Administration Tools and click Continue on the Destination Select page.
7. Click the Install button.
8. Enter your Administrative Username and Password.
9. Click Continue Installation with the understanding that the system must be rebooted once the installation is complete.
10. Read the information on the Summary page and then click Restart.

© Copyright 2012 Apple Inc.

How to verify that the FIPS Administration Tools were successfully installed

There are two places to check when verifying that the FIPS Administration Tools were installed successfully.

The first place to check is in `/System/Library/LaunchDaemons/` for the `com.apple.fipspost.plist` file. This is the control file for the FIPS Launch Daemon.

```
/System/Library/LaunchDaemons/com.apple.fipspost.plist
```

The second place to check is in the `/usr/sbin/fips` folder that is created during the installation. The key executables installed in that folder are:

```
1. FIPSPerformSelfTest      (POST Tool)
2. cryptoKAT                (CRYPTO Algorithm Tests)
3. postsig                  (DSA/ECDSA Sig Tests)
```

The following lists all files installed in the `/usr/sbin/fips` folder.

```
-r--------    1 root  wheel     128 Jul  6  2007 128_incr
-r--------    1 root  wheel     128 Jul  6  2007 128_rand
-r--------    1 root  wheel     128 Jul  6  2007 128_zeroes
-r--------    1 root  wheel      16 Jul  6  2007 16_incr
-r--------    1 root  wheel      16 Jul  6  2007 16_rand
-r--------    1 root  wheel      16 Jul  6  2007 16_zeroes
-r--------    1 root  wheel      24 Jul  6  2007 24_incr
-r--------    1 root  wheel      24 Jul  6  2007 24_rand
-r--------    1 root  wheel      24 Jul  2  2007 24_zeroes
-r--------    1 root  wheel      32 Jul  6  2007 32_incr
-r--------    1 root  wheel      32 Jul  6  2007 32_rand
-r--------    1 root  wheel      32 Jul  6  2007 32_zeroes
-r--------    1 root  wheel       4 Jul  6  2007 4_incr
-r--------    1 root  wheel       4 Jul  6  2007 4_rand
-r--------    1 root  wheel       4 Jul  6  2007 4_zeroes
-r--------    1 root  wheel      60 Jul  6  2007 60_incr
-r--------    1 root  wheel      60 Jul  6  2007 60_rand
-r--------    1 root  wheel      60 Jul  6  2007 60_zeroes
-r--------    1 root  wheel      64 Jul  6  2007 64_incr
-r--------    1 root  wheel      64 Jul  6  2007 64_rand
-r--------    1 root  wheel      64 Jul  6  2007 64_zeroes
-r--------    1 root  wheel       6 Jul  6  2007 6_incr
-r--------    1 root  wheel       6 Jul  6  2007 6_zeroes
-r--------    1 root  wheel       8 Jul  6  2007 8_incr
-r--------    1 root  wheel       8 Jul  6  2007 8_rand
-r--------    1 root  wheel       8 Jul  2  2007 8_zeroes
-r-xr-xr-x    1 root  wheel  151008 Dec 21 13:01 FIPSPerformSelfTest
-r-x------    1 root  wheel   29568 Jul  7 16:58 cryptoKAT
-r-x------    1 root  wheel  303344 Jul  8 13:26 postsig
-r--------    1 root  wheel     635 Jul  6  2007 rsa1024_priv.der
-r--------    1 root  wheel     140 Jul  6  2007 rsa1024_pub.der
```

Once the above files are installed, the installer then runs the `FIPSPerformSelfTest create` command and the following Reference Files are created for validation in the same `/usr/sbin/fips` folder.

© Copyright 2012 Apple Inc.

```
-rw-------    1 root  wheel      64 Dec 19 22:35 ctext_3des_cbc_64_incr
-rw-------    1 root  wheel       8 Dec 19 22:35 ctext_3des_cbc_8_zeroes
-rw-------    1 root  wheel      64 Dec 19 22:35 ctext_3des_cbc_pad_60_incr
-rw-------    1 root  wheel      16 Dec 19 22:35 ctext_3des_cbc_pad_8_zeroes
-rw-------    1 root  wheel      64 Dec 19 22:35 ctext_3des_ecb_64_incr
-rw-------    1 root  wheel       8 Dec 19 22:35 ctext_3des_ecb_8_zeroes
-rw-------    1 root  wheel      16 Dec 19 22:35 ctext_aes192_cbc_16_zeroes
-rw-------    1 root  wheel      64 Dec 19 22:35 ctext_aes192_cbc_64_incr
-rw-------    1 root  wheel      32 Dec 19 22:35 ctext_aes192_cbc_pad_16_zeroes
-rw-------    1 root  wheel      64 Dec 19 22:35 ctext_aes192_cbc_pad_60_incr
-rw-------    1 root  wheel      16 Dec 19 22:35 ctext_aes192_ecb_16_zeroes
-rw-------    1 root  wheel      64 Dec 19 22:35 ctext_aes192_ecb_64_incr
-rw-------    1 root  wheel      16 Dec 19 22:35 ctext_aes256_cbc_16_zeroes
-rw-------    1 root  wheel      64 Dec 19 22:35 ctext_aes256_cbc_64_incr
-rw-------    1 root  wheel      32 Dec 19 22:35 ctext_aes256_cbc_pad_16_zeroes
-rw-------    1 root  wheel      64 Dec 19 22:35 ctext_aes256_cbc_pad_60_incr
-rw-------    1 root  wheel      16 Dec 19 22:35 ctext_aes256_ecb_16_zeroes
-rw-------    1 root  wheel      64 Dec 19 22:35 ctext_aes256_ecb_64_incr
-rw-------    1 root  wheel      16 Dec 19 22:35 ctext_aes_cbc_16_zeroes
-rw-------    1 root  wheel      64 Dec 19 22:35 ctext_aes_cbc_64_incr
-rw-------    1 root  wheel      32 Dec 19 22:35 ctext_aes_cbc_pad_16_zeroes
-rw-------    1 root  wheel      64 Dec 19 22:35 ctext_aes_cbc_pad_60_incr
-rw-------    1 root  wheel      16 Dec 19 22:35 ctext_aes_ecb_16_zeroes
-rw-------    1 root  wheel      64 Dec 19 22:35 ctext_aes_ecb_64_incr
-rw-------    1 root  wheel     128 Dec 19 22:35 ctext_rsa1024_incr
-rw-------    1 root  wheel     128 Dec 19 22:35 ctext_rsa1024_rand
-rw-------    1 root  wheel      20 Dec 19 22:35 digest_sha1_24_incr
-rw-------    1 root  wheel      20 Dec 19 22:35 digest_sha1_60_incr
-rw-------    1 root  wheel      20 Dec 19 22:35 digest_sha1_8_zeroes
-rw-------    1 root  wheel      28 Dec 19 22:35 digest_sha224_24_incr
-rw-------    1 root  wheel      28 Dec 19 22:35 digest_sha224_60_incr
-rw-------    1 root  wheel      28 Dec 19 22:35 digest_sha224_8_zeroes
-rw-------    1 root  wheel      32 Dec 19 22:35 digest_sha256_24_incr
-rw-------    1 root  wheel      32 Dec 19 22:35 digest_sha256_60_incr
-rw-------    1 root  wheel      32 Dec 19 22:35 digest_sha256_8_zeroes
-rw-------    1 root  wheel      48 Dec 19 22:35 digest_sha384_24_incr
-rw-------    1 root  wheel      48 Dec 19 22:35 digest_sha384_60_incr
-rw-------    1 root  wheel      48 Dec 19 22:35 digest_sha384_8_zeroes
-rw-------    1 root  wheel      64 Dec 19 22:35 digest_sha512_24_incr
-rw-------    1 root  wheel      64 Dec 19 22:35 digest_sha512_60_incr
-rw-------    1 root  wheel      64 Dec 19 22:35 digest_sha512_8_zeroes
-rw-------    1 root  wheel      20 Dec 19 22:35 hmac_sha1_16_incr
-rw-------    1 root  wheel      20 Dec 19 22:35 hmac_sha1_24_incr
-rw-------    1 root  wheel      20 Dec 19 22:35 hmac_sha1_24_incr_edc
-rw-------    1 root  wheel      20 Dec 19 22:35 hmac_sha1_4_incr
-rw-------    1 root  wheel      20 Dec 19 22:35 hmac_sha1_6_incr
-rw-------    1 root  wheel      20 Dec 19 22:35 hmac_sha1_8_zeroes
-rw-------    1 root  wheel     128 Dec 19 22:35 sig_rsa_60_incr
-rw-------    1 root  wheel     128 Dec 19 22:35 sig_rsa_8_zeroes
-rw-------    1 root  wheel     128 Dec 19 22:35 sig_rsa_sha224_60_incr
-rw-------    1 root  wheel     128 Dec 19 22:35 sig_rsa_sha256_60_incr
-rw-------    1 root  wheel     128 Dec 19 22:35 sig_rsa_sha384_60_incr
-rw-------    1 root  wheel     128 Dec 19 22:35 sig_rsa_sha512_60_incr
```

Also note that the contents of the `hmac_sha1_24_incr_edc` reference file was created using an HMAC-SHA1 against the PRNG (`/mach_kernel`). The contents of the reference file will be compared against an HMAC-SHA1 calculation of the PRNG during the POST.

The FIPS Administration Tools installation is successful if all files listed above are found in the `/usr/sbin/fips` folder on the system.

© Copyright 2012 Apple Inc.

The FIPS Power-On-Self-Test (POST) process flow

1. Apple Mac system is physically Powered on

2. Operating System (OS X Lion) begins bootstrap process

3. OS X Lion automatically launches FIPSPerformSelfTest

   3.1. FIPSPerformSelfTest is launched via the `launchd` process with parameters defined in the LaunchDaemon "plist" file:

   ```
   /System/Library/LaunchDaemons/com.apple.fipspost.plist
   ```

4. FIPSPerformSelfTest controls the Self-Test Process

   4.1. Notify of Power-On-Self-Test (POST) Launch

   Logged to `/var/log/secure.log`

   ```
   ****************************************
   *** RUNNING FIPS Power-On-Self-Test     ***
   ***         FIPSPerformSelfTest         ***
   ***         12/20/2010 11:19:37         ***
   ****************************************
   ```

   4.2. Validate the Crypto Module and POST tools' RSA Digital Signatures and the PRNG with an HMAC-SHA1 data authentication code (DAC).

   ```
   [FIPSPerformSelfTest][ValidSignatures]: Module Integrity using
   "codesign" Validation - Digital Signature (signed code)
   [FIPSPerformSelfTest][ValidSignatures]: Module Path: /System/
   Library/Frameworks/Security.framework/Versions/A/Security
   [FIPSPerformSelfTest][ValidSignatures]: Security Module Validation:
   PASSED
   [FIPSPerformSelfTest][ValidSignatures]: PRNG Module Validation:
   PASSED
   [FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/
   FIPSPerformSelfTest =>> PASSED
   [FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/cryptoKAT
   =>> PASSED
   [FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/postsig =>>
   PASSED
   ```

   Refer to the following for more information:
   • "How to verify the integrity of the FIPS Administration Tools" on page 8.
   • "How to verify the integrity of the Crypto Module (security framework)" on page 12.
   • "How to verify the integrity of the PRNG (mach_kernel) component of the Crypto Module" on page 14.

   4.3. Execute Validation of Algorithms with KAT Files

   The `FIPSPerformSelfTest` calls the two KAT tools (`cryptoKAT` and `postsig`) which processes the encryption algorithms and compares the results against the known answers.

© Copyright 2012 Apple Inc.

Refer to "List of Algorithm Known Answer Tests (KAT) performed" tables on page 19 for more information.

4.4. Notify the Successful Completion of the Known Answer Tests and of the complete Power-On-Self-Test process

```
[FIPSPerformSelfTest][POST KAT]: Algorithm Tests: STARTED
[FIPSPerformSelfTest] :AES-128:  COMPLETE
[FIPSPerformSelfTest] :AES-192:  COMPLETE
[FIPSPerformSelfTest] :AES-256:  COMPLETE
[FIPSPerformSelfTest] :3DES:     COMPLETE
[FIPSPerformSelfTest] :SHA-1:    COMPLETE
[FIPSPerformSelfTest] :SHA-224:  COMPLETE
[FIPSPerformSelfTest] :SHA-256:  COMPLETE
[FIPSPerformSelfTest] :SHA-384:  COMPLETE
[FIPSPerformSelfTest] :SHA-512:  COMPLETE
[FIPSPerformSelfTest] :HMAC/SHA1:COMPLETE
[FIPSPerformSelfTest] :RSAwSHA-X:COMPLETE
[FIPSPerformSelfTest] :RSA RAW:  COMPLETE
[FIPSPerformSelfTest] :DSA:      COMPLETE
[FIPSPerformSelfTest] :ECDSA:    COMPLETE
[FIPSPerformSelfTest][POST KAT]: Algorithm Tests: PASSED
[FIPSPerformSelfTest][FIPS POST/KAT] Completion : PASSED
12/20/2010 11:19:37
```

4.5. Exit the Power-On-Self-Test (POST) and continue normal bootup process

4.6. Halt upon failure of "ANY" of the Tests

If "ANY" phase or step of testing components fails, the system will log the failure and Halt/Shutdown the Operating System immediately.

The logging messages are sent to the `/var/log/secure.log` file which requires Administrator privileges to view its contents.

Refer to "How to identify a FIPS issue in the secure.log" on page 15 for more information.

© Copyright 2012 Apple Inc.

How to verify the integrity of the FIPS Administration Tools

The tool used to verify the integrity of the FIPS Administration Tools is called `FIPSPerformSelfTest`.

To see what options are available with `FIPSPerformSelfTest`, launch Terminal in the `/Applications/Utilities` folder and run the following command :

`/usr/sbin/fips/FIPSPerformSelfTest`

The result should be:

```
Usage:

FIPSPerformSelfTest version    # Display the current version            (User / Crypto Officer)
FIPSPerformSelfTest status     # Display the current FIPS MODE Status    (User / Crypto Officer)
FIPSPerformSelfTest signatures # Validate Tool Code Signatures           (Crypto Officer)
FIPSPerformSelfTest full       # Run the full suite of tests             (Crypto Officer)
FIPSPerformSelfTest create     # Forces creation of new KAT files        (Crypto Officer)
```

1. **FIPSPerformSelfTest VERSION**

   Verify the FIPS Tools Version :

   `/usr/sbin/fips/FIPSPerformSelfTest version`

   The result should be similar to the following:

   ```
   [FIPSPerformSelfTest][ToolVersion] FIPSPerformSelfTest v1.3, December 16, 2011
   ```

2. **FIPSPerformSelfTest STATUS**

   Verify that FIPS Mode Status :

   `/usr/sbin/fips/FIPSPerformSelfTest status`

   The result should be:

   ```
   [FIPSPerformSelfTest][ModeStatus] FIPS Mode Status : ENABLED
   ```

3. **FIPSPerformSelfTest SIGNATURES**

   Verify the FIPS Signatures :

   `/usr/sbin/fips/FIPSPerformSelfTest signatures`

   The result should be:

   ```
   [FIPSPerformSelfTest][ValidSignatures]: Module Integrity using "codesign"
   Validation - Digital Signature (signed code)
   [FIPSPerformSelfTest][ValidSignatures]: Module Path: /System/Library/Frameworks/
   Security.framework/Versions/A/Security
   [FIPSPerformSelfTest][ValidSignatures]: Security Module Validation:  PASSED
   [FIPSPerformSelfTest][ValidSignatures]: PRNG Module Validation:      PASSED
   [FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/FIPSPerformSelfTest =>>
   PASSED
   [FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/cryptoKAT =>> PASSED
   [FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/postsig =>> PASSED
   ```

© Copyright 2012 Apple Inc.

4. **FIPSPerformSelfTest FULL**

   Run Full FIPS Self Test at anytime :

   `/usr/sbin/fips/FIPSPerformSelfTest full`

   The results should be similar to the following information. This information is written to the `var/log/secure.log` when the POST runs at boot up.

   Sample `/var/log/secure.log` output:

```
*****************************************
*** RUNNING FIPS Power-On-Self-Test     ***
***         FIPSPerformSelfTest         ***
***         12/20/2011 11:19:37         ***
*****************************************
[FIPSPerformSelfTest][ValidSignatures]: Module Integrity using "codesign"
Validation - Digital Signature (signed code)
[FIPSPerformSelfTest][ValidSignatures]: Module Path: /System/Library/
Frameworks/Security.framework/Versions/A/Security
[FIPSPerformSelfTest][ValidSignatures]: Security Module Validation:  PASSED
[FIPSPerformSelfTest][ValidSignatures]: PRNG Module Validation:      PASSED
[FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/FIPSPerformSelfTest
=>> PASSED
[FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/cryptoKAT =>> PASSED
[FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/postsig =>> PASSED
[FIPSPerformSelfTest][POST KAT]: Algorithm Tests: STARTED
[FIPSPerformSelfTest] :AES-128:  COMPLETE
[FIPSPerformSelfTest] :AES-192:  COMPLETE
[FIPSPerformSelfTest] :AES-256:  COMPLETE
[FIPSPerformSelfTest] :3DES:     COMPLETE
[FIPSPerformSelfTest] :SHA-1:    COMPLETE
[FIPSPerformSelfTest] :SHA-224:  COMPLETE
[FIPSPerformSelfTest] :SHA-256:  COMPLETE
[FIPSPerformSelfTest] :SHA-384:  COMPLETE
[FIPSPerformSelfTest] :SHA-512:  COMPLETE
[FIPSPerformSelfTest] :HMAC/SHA1:COMPLETE
[FIPSPerformSelfTest] :RSAwSHA-X:COMPLETE
[FIPSPerformSelfTest] :RSA RAW:  COMPLETE
[FIPSPerformSelfTest] :DSA:      COMPLETE
[FIPSPerformSelfTest] :ECDSA:    COMPLETE
[FIPSPerformSelfTest][POST KAT]: Algorithm Tests: PASSED
[FIPSPerformSelfTest][FIPS POST/KAT] Completion : PASSED  12/20/2011
11:19:37
```

5. **FIPSPerformSelfTest CREATE**

   Force the creation of new KAT files:

   `/usr/sbin/fips/FIPSPerformSelfTest create`

   The result should be similar to the following:

```
...wrote 16 bytes to /usr/sbin/fips/ctext_aes_ecb_16_zeroes
...wrote 16 bytes to /usr/sbin/fips/ctext_aes_cbc_16_zeroes
...wrote 32 bytes to /usr/sbin/fips/ctext_aes_cbc_pad_16_zeroes
...wrote 64 bytes to /usr/sbin/fips/ctext_aes_ecb_64_incr
...wrote 64 bytes to /usr/sbin/fips/ctext_aes_cbc_64_incr
...wrote 64 bytes to /usr/sbin/fips/ctext_aes_cbc_pad_60_incr
[FIPSPerformSelfTest] :AES-128:  COMPLETE
```

© Copyright 2012 Apple Inc.

```
...wrote 16 bytes to /usr/sbin/fips/ctext_aes192_ecb_16_zeroes
...wrote 16 bytes to /usr/sbin/fips/ctext_aes192_cbc_16_zeroes
...wrote 32 bytes to /usr/sbin/fips/ctext_aes192_cbc_pad_16_zeroes
...wrote 64 bytes to /usr/sbin/fips/ctext_aes192_ecb_64_incr
...wrote 64 bytes to /usr/sbin/fips/ctext_aes192_cbc_64_incr
...wrote 64 bytes to /usr/sbin/fips/ctext_aes192_cbc_pad_60_incr
[FIPSPerformSelfTest] :AES-192:  COMPLETE
...wrote 16 bytes to /usr/sbin/fips/ctext_aes256_ecb_16_zeroes
...wrote 16 bytes to /usr/sbin/fips/ctext_aes256_cbc_16_zeroes
...wrote 32 bytes to /usr/sbin/fips/ctext_aes256_cbc_pad_16_zeroes
...wrote 64 bytes to /usr/sbin/fips/ctext_aes256_ecb_64_incr
...wrote 64 bytes to /usr/sbin/fips/ctext_aes256_cbc_64_incr
...wrote 64 bytes to /usr/sbin/fips/ctext_aes256_cbc_pad_60_incr
[FIPSPerformSelfTest] :AES-256:  COMPLETE
...wrote 8 bytes to /usr/sbin/fips/ctext_3des_ecb_8_zeroes
...wrote 8 bytes to /usr/sbin/fips/ctext_3des_cbc_8_zeroes
...wrote 16 bytes to /usr/sbin/fips/ctext_3des_cbc_pad_8_zeroes
...wrote 64 bytes to /usr/sbin/fips/ctext_3des_ecb_64_incr
...wrote 64 bytes to /usr/sbin/fips/ctext_3des_cbc_64_incr
...wrote 64 bytes to /usr/sbin/fips/ctext_3des_cbc_pad_60_incr
[FIPSPerformSelfTest] :3DES:     COMPLETE
...wrote 20 bytes to /usr/sbin/fips/digest_sha1_8_zeroes
...wrote 20 bytes to /usr/sbin/fips/digest_sha1_24_incr
...wrote 20 bytes to /usr/sbin/fips/digest_sha1_60_incr
[FIPSPerformSelfTest] :SHA-1:    COMPLETE
...wrote 28 bytes to /usr/sbin/fips/digest_sha224_8_zeroes
...wrote 28 bytes to /usr/sbin/fips/digest_sha224_24_incr
...wrote 28 bytes to /usr/sbin/fips/digest_sha224_60_incr
[FIPSPerformSelfTest] :SHA-224:  COMPLETE
...wrote 32 bytes to /usr/sbin/fips/digest_sha256_8_zeroes
...wrote 32 bytes to /usr/sbin/fips/digest_sha256_24_incr
...wrote 32 bytes to /usr/sbin/fips/digest_sha256_60_incr
[FIPSPerformSelfTest] :SHA-256:  COMPLETE
...wrote 48 bytes to /usr/sbin/fips/digest_sha384_8_zeroes
...wrote 48 bytes to /usr/sbin/fips/digest_sha384_24_incr
...wrote 48 bytes to /usr/sbin/fips/digest_sha384_60_incr
[FIPSPerformSelfTest] :SHA-384:  COMPLETE
...wrote 64 bytes to /usr/sbin/fips/digest_sha512_8_zeroes
...wrote 64 bytes to /usr/sbin/fips/digest_sha512_24_incr
...wrote 64 bytes to /usr/sbin/fips/digest_sha512_60_incr
[FIPSPerformSelfTest] :SHA-512:  COMPLETE
...wrote 20 bytes to /usr/sbin/fips/hmac_sha1_8_zeroes
...wrote 20 bytes to /usr/sbin/fips/hmac_sha1_4_incr
...wrote 20 bytes to /usr/sbin/fips/hmac_sha1_6_incr
...wrote 20 bytes to /usr/sbin/fips/hmac_sha1_16_incr
...wrote 20 bytes to /usr/sbin/fips/hmac_sha1_24_incr
[FIPSPerformSelfTest] :HMAC/SHA1:COMPLETE
...wrote 128 bytes to /usr/sbin/fips/sig_rsa_8_zeroes
...wrote 128 bytes to /usr/sbin/fips/sig_rsa_60_incr
...wrote 128 bytes to /usr/sbin/fips/sig_rsa_sha224_60_incr
...wrote 128 bytes to /usr/sbin/fips/sig_rsa_sha256_60_incr
...wrote 128 bytes to /usr/sbin/fips/sig_rsa_sha384_60_incr
...wrote 128 bytes to /usr/sbin/fips/sig_rsa_sha512_60_incr
[FIPSPerformSelfTest] :RSAwSHA-X:COMPLETE
```

© Copyright 2012 Apple Inc.

```
...wrote 128 bytes to /usr/sbin/fips/ctext_rsa1024_incr
...wrote 128 bytes to /usr/sbin/fips/ctext_rsa1024_rand
[FIPSPerformSelfTest] :RSA RAW:  COMPLETE
[FIPSPerformSelfTest]KAT Reference File Creation:   PASSED
...wrote 20 bytes to /usr/sbin/fips/hmac_sha1_24_incr_edc
[FIPSPerformSelfTest]PRNG Integrity File Creation:  PASSED
```

© Copyright 2012 Apple Inc.

How to verify the integrity of the CSP (security framework) component of the Crypto Module

The Crypto Officer can manually verify the integrity of the CSP (security framework) component of the Crypto Module with the following information. This integrity verification is run automatically when executing the `/usr/sbin/fips/FIPSPerformSelfTest signatures` command.

1. Ensure you have obtained root privileges
2. Launch the Terminal Utility (located at `/Applications/Utilities/Terminal`)
    - 2.1. At the Command Prompt "$", enter the following: `sudo -s`
    - 2.2. Enter your password
    - 2.3. The Command Prompt should now be "#" indicating root privileges. All commands below begin with "**#**" which indicates the Command Line Prompt (not to be entered on keyboard) for root.
    - 2.4. At the Command Prompt "#", enter the following: `cd /usr/sbin/fips`
3. Testing CSP (Security Framework) Integrity:
    - 3.1. At the Command Prompt "#", enter the following command (a single line):
      ```
      codesign -v -R="identifier com.apple.security and anchor
      apple" /System/Library/Frameworks/Security.framework/
      Versions/A/Security
      ```
    - 3.2. A successful Module Integrity Check returns the user to the Command Prompt "#" with no additional response
    - 3.3. A Module Integrity Check failure will return the following error:
      ```
      test-requirement: failed to satisfy code requirement(s)
      ```
    - 3.4. If additional information is needed about the Module's digital signature, run the following command:
      ```
      codesign -dvvvv -R="identifier com.apple.security and anchor
      apple" /System/Library/Frameworks/Security.framework/
      Versions/A/Security
      ```
        - 3.4.1. The following information shows an example of what the correct output can look like for the Module's digital signature when the integrity has been maintained. Note the unique Identifier should be com.apple.security and the Authority should resolve to Apple Root CA.

          ```
          Identifier=com.apple.security
          Format=bundle with Mach-O universal (i386 ppc7400 x86_64)
          CodeDirectory v=20100 size=16607 flags=0x0(none)
          hashes=824+3 location=embedded
          ```

© Copyright 2012 Apple Inc.

```
CDHash=b5862ac6d8bda5c35f6f92f605be8f7baca1c138
Signature size=4064
Authority=Software Signing
Authority=Apple Code Signing Certification Authority
Authority=Apple Root CA
Info.plist entries=8
Sealed Resources rules=9 files=21
Internal requirements count=1 size=148
```

3.4.2. Verify Module State Changes by examining the Secure.log File (`var/log/secure.log`)

3.4.3. The Secure.log (`var/log/secure.log`) should show status for each *Module Test Session* initiated on the computer system.

© Copyright 2012 Apple Inc.

How to verify the integrity of the PRNG (mach_kernel) component of the Crypto Module

The Crypto Officer can manually verify the integrity of the PRNG (mach_kernel) component of the Crypto Module with the following information. This integrity verification is run automatically when executing the `/usr/sbin/fips/FIPSPerformSelfTest signatures` command.

1. Ensure you have obtained root privileges
2. Launch the Terminal Utility (located at `/Applications/Utilities/Terminal`)
    - 2.1. At the Command Prompt "$", enter the following: `sudo -s`
    - 2.2. Enter your password
    - 2.3. The Command Prompt should now be "#" indicating root privileges All commands below begin with "**#**" which indicates the Command Line Prompt (not to be entered on keyboard) for root.
    - 2.4. At the Command Prompt "#", enter the following: `cd /usr/sbin/fips`
3. Testing PRNG (mach_kernel) Integrity:
    - 3.1. Run the following command: `/usr/sbin/fips/cryptoKAT -a hmacsha1 -p /mach_kernel -k /usr/sbin/fips/24_incr -c /usr/sbin/fips/hmac_sha1_24_incr_edc`
    - 3.2. A successful Module Integrity Check returns the user to the Command Prompt "#" with no additional response
    - 3.3. A Module Integrity Check failure will return the following error:

```
***Ciphertext data mismatch
***Error on algorithm hmacsha1. Aborting.
```

© Copyright 2012 Apple Inc.

How to identify a FIPS issue in the secure.log

There are two basic situations where the Crypto Officer may need to identify an integrity issue or take remedial action on a FIPS enabled system. The first such situation is when the Crypto Officer would like to perform a full ad-hoc test of the crypto module for purposes of assurance and the second would be with a system that shuts down automatically after power up, indicating a possible identification of integrity failure by the FIPS Power On Self-Test (POST).

In the first case, the Crypto Officer requires assurance that the Crypto Module integrity has remained intact. Running the `/usr/sbin/fips/FIPSPerformSelfTest full` command at this point will either verify that everything is working correctly or automatically shut the system down, indicating a FIPS related problem. This integrity validation command can be leveraged by any client management and reporting systems for compliance purposes.

In the second case, a system is shutting down automatically after boot up and the end user cannot access the login window. This is a strong indication of a forced shutdown due to an integrity failure found during the FIPS Power-On-Self-Test (POST).

There are many possible steps to take depending on the problem encountered. The following information provides the most common issues and their solutions. Additional guidance can be found from Apple's Technical Support Knowledge Base System found at http://support.apple.com/ with a keyword search for FIPS.

- **Situation :** FIPS enabled system is on and running

If the Crypto Officer requires assurance that the Crypto Module integrity has remained intact, the first thing to do is to run the `/usr/sbin/fips/FIPSPerformSelfTest` command with the "`full`" flag :

```
/usr/sbin/fips/FIPSPerformSelfTest full
```

If the resulting information is similar to the following output (also found in `var/log/secure.log`) then the Crypto Module integrity remains intact and passes all validation tests:

```
****************************************
*** RUNNING FIPS Power-On-Self-Test     ***
***         FIPSPerformSelfTest         ***
***         12/20/2011 11:19:37         ***
****************************************
[FIPSPerformSelfTest][ValidSignatures]: Module Integrity using "codesign"
Validation - Digital Signature (signed code)
[FIPSPerformSelfTest][ValidSignatures]: Module Path: /System/Library/
Frameworks/Security.framework/Versions/A/Security
[FIPSPerformSelfTest][ValidSignatures]: Security Module Validation:  PASSED
[FIPSPerformSelfTest][ValidSignatures]: PRNG Module Validation:      PASSED
[FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/FIPSPerformSelfTest
=>> PASSED
[FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/cryptoKAT =>> PASSED
[FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/postsig =>> PASSED
[FIPSPerformSelfTest][POST KAT]: Algorithm Tests: STARTED
[FIPSPerformSelfTest] :AES-128:  COMPLETE
[FIPSPerformSelfTest] :AES-192:  COMPLETE
[FIPSPerformSelfTest] :AES-256:  COMPLETE
[FIPSPerformSelfTest] :3DES:     COMPLETE
[FIPSPerformSelfTest] :SHA-1:    COMPLETE
[FIPSPerformSelfTest] :SHA-224:  COMPLETE
[FIPSPerformSelfTest] :SHA-256:  COMPLETE
[FIPSPerformSelfTest] :SHA-384:  COMPLETE
```

© Copyright 2012 Apple Inc.

```
[FIPSPerformSelfTest] :SHA-512:  COMPLETE
[FIPSPerformSelfTest] :HMAC/SHA1:COMPLETE
[FIPSPerformSelfTest] :RSAwSHA-X:COMPLETE
[FIPSPerformSelfTest] :RSA RAW:  COMPLETE
[FIPSPerformSelfTest] :DSA:      COMPLETE
[FIPSPerformSelfTest] :ECDSA:    COMPLETE
[FIPSPerformSelfTest][POST KAT]: Algorithm Tests: PASSED
[FIPSPerformSelfTest][FIPS POST/KAT] Completion : PASSED  12/20/2011 11:19:37
```

If an integrity failure has been detected when running the `/usr/sbin/fips/FIPSPerformSelfTest full` command, the system will automatically shutdown. If the system is powered up, and any Crypto Module issue has not been resolved, the system will once again automatically shutdown again.

• **Situation :** FIPS enabled system has suddenly shut down and will automatically shut down during a restart

If the FIPS enabled system will not fully boot up after the system is powered on (i.e. the system automatically shuts itself back down) then you must start the system in Single User Mode, mount the hard drive, and check the security.log to see if the system's FIPS POST tests have identified an integrity issue.

1. Press the power button to start the computer.
2. Immediately press and hold the Command (Apple) key and the "s" key for single-user mode. (Command-S)
3. Once the computer gets to the `:/ root#` prompt, type the following command to mount the Hard Drive:

   ```
   /sbin/mount -uw /
   ```

4. To view full contents of the current secure log, type the following command:

   ```
   cat var/log/secure.log
   ```

   Note: The system may contain multiple Secure Log files if additional maintenance has not been performed. Additional log files would have the file name structure such as secure.log.0.bz2.

5. If you see successful FIPS POST status messages in the `/var/log/secure.log`, then the integrity of the crypto module is not the cause of the shutdown.

   However, if there is a failure identified during the FIPS POST, you'll see failed FIPS POST messages similar to the examples below.

   ```
   ****************************************
   *** RUNNING FIPS Power-On-Self-Test     ***
   ***         FIPSPerformSelfTest         ***
   ***         12/21/2011 14:12:55         ***
   ****************************************
   [FIPSPerformSelfTest][ValidSignatures]: Module Integrity using "codesign" Valida-
   tion - Digital Signature (signed code)
   ```

© Copyright 2012 Apple Inc.

```
[FIPSPerformSelfTest][ValidSignatures]: Module Path:
/System/Library/Frameworks/Security.framework/Versions/A/Security
[FIPSPerformSelfTest][ValidSignatures]: Security Module Validation:  PASSED
[FIPSPerformSelfTest][ValidSignatures]: PRNG Module Validation:      PASSED
[FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/FIPSPerformSelfTest =>>
PASSED
[FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/cryptoKAT =>> PASSED
[FIPSPerformSelfTest][ValidSignatures]: /usr/sbin/fips/postsig =>> PASSED
[FIPSPerformSelfTest][POST KAT]: Algorithm Tests: STARTED
[FIPSPerformSelfTest] :AES-128:  COMPLETE
[FIPSPerformSelfTest] :AES-192:  COMPLETE
[FIPSPerformSelfTest] :AES-256:  COMPLETE
[FIPSPerformSelfTest] :3DES:     COMPLETE
[FIPSPerformSelfTest] :SHA-1:    COMPLETE
[FIPSPerformSelfTest] :SHA-224:  COMPLETE
[FIPSPerformSelfTest] :SHA-256:  COMPLETE
[FIPSPerformSelfTest] :SHA-384:  COMPLETE
[FIPSPerformSelfTest] :SHA-512:  COMPLETE
[FIPSPerformSelfTest][ValidateAlgorithms][ABORT] Error while processing [HMAC/SHA-
1] KAT files.
[FIPSPerformSelfTest][POST KAT]: Algorithm Tests: FAILED
[FIPSPerformSelfTest][FIPS POST/KAT] Completion : FAILED  12/21/2010 14:12:55
*****************************************
*** CRITICAL: FIPS POST/KAT FAILED      ***
*****************************************
***  Halting all processes and          ***
***  Shutting System Down immediately   ***
***                                     ***
*** TO resolve problems with module:    ***
***   1) Reboot into Single User Mode   ***
***   2) Review /var/log/secure.log     ***
***   3) Make appropriate changes       ***
***   4) Reboot machine                 ***
*****************************************
***         12/21/2011 14:12:56         ***
Dec 21 14:13:02 Macintosh HD shutdown[132]: halt by _usbmuxd:
Shutdown NOW!
```

Note that the error message indicates where the problem occurred, `[ABORT] Error while processing [HMAC/SHA-1] KAT files`, and this is the first place to start looking when addressing the issue.

Situation**:** If any situations occur that are not documented here, refer to the Apple Knowledge Base at http://www.apple.com/support for additional information.

© Copyright 2012 Apple Inc.

How to mitigate a crypto module integrity issue

If a crypto module integrity issue has been identified in the /var/log/secure.log, then there are several steps that should be taken to allow further research by the Crypto Officer or to allow the Crypto Officer to reset the FIPS installation.

Disable the Launch Daemon for the FIPS POST so that the system will boot normally

In order to do further analysis of a system that may be experiencing a crypto module integrity issue, it will be necessary to disable the Launch Daemon for the FIPS POST. This will allow the system to boot normally for additional tests and possible installations to be run on the system. To accomplish this, you will need to boot into single user mode and move the Launch Daemon for the FIPS POST (renaming it will not work, the file needs to be moved).

1. Press the power button to start the computer.
2. Immediately press and hold the Command (Apple) key and the "s" key for single-user mode. (Command-S)
3. Once the computer gets to the `:/ root#` prompt, type the following command to mount the Hard Drive:
   `/sbin/mount -uw /`
4. Type the following command:
   `mv /System/Library/LaunchDaemons/com.apple.fipspost.plist /`
5. This will move the `com.apple.fipspost.plist` Launch Daemon to the root of the hard drive.
6. Type the following command to restart the system:
   `reboot`
7. The system will now boot normally and the Crypto Module installation and integrity can be checked along with the system.

Reinstall the FIPS Administration Tools

If the decision is made by the Crypto Officer to reinstall the FIPS Administration Tools, it is possible to reinstall over the existing installation. However, it is recommended to delete the `com.apple.fipspost.plist` as well as the FIPS folder in `/usr/sbin/fips/` as this always ensures a clean install.

Reinstall OS X Lion

If there is an integrity issue found with the security framework or other issues that the Crypto Officer discovers that cannot be easily remediated, simple reinstallation of the FIPS Administration Tools may not be appropriate. If this is the case, reinstalling OS X Lion may be the correct next step. After a reinstallation of OS X Lion, the Crypto Officer must repeat the installation of the FIPS Administration Tools.

© Copyright 2012 Apple Inc.

List of Algorithm Known Answer Tests (KAT) performed

AES 128-bit

PlainText (File)	Raw Key Bits (File)	Cipher Text (File)	Block Size (Bytes)	Description
16_zeroes	16_zeroes	ctext_aes_ecb_16_zeroes	16	AES - 128 - ECB - No Padding
16_zeroes	16_zeroes	ctext_aes_cbc_16_zeroes	16	AES - 128 - CBC - No Padding
16_zeroes	16_zeroes	ctext_aes_cbc_pad_16_zeroes	16	AES - 128 - CBC - Padding
64_incr	16_incr	ctext_aes_ecb_64_incr	16	AES - 128 - ECB - No Padding
64_incr	16_incr	ctext_aes_cbc_64_incr	16	AES - 128 - CBC - No Padding
60_incr	16_incr	ctext_aes_cbc_pad_60_incr	16	AES - 128 - CBC - Padding

AES 192-bit

PlainText (File)	Raw Key Bits (File)	Cipher Text (File)	Block Size (Bytes)	Description
16_zeroes	24_zeroes	ctext_aes192_ecb_16_zeroes	16	AES - 192 - ECB - No Padding
16_zeroes	24_zeroes	ctext_aes192_cbc_16_zeroes	16	AES - 192 - CBC - No Padding
16_zeroes	24_zeroes	ctext_aes192_cbc_pad_16_zeroes	16	AES - 192 - CBC - Padding
64_incr	24_incr	ctext_aes192_ecb_64_incr	16	AES - 192 - ECB - No Padding
64_incr	24_incr	ctext_aes192_cbc_64_incr	16	AES - 192 - CBC - No Padding
60_incr	24_incr	ctext_aes192_cbc_pad_60_incr	16	AES - 192 - CBC - Padding

AES 256-bit

PlainText (File)	Raw Key Bits (File)	Cipher Text (File)	Block Size (Bytes)	Description
16_zeroes	32_zeroes	ctext_aes256_ecb_16_zeroes	16	AES - 256 - ECB - No Padding
16_zeroes	32_zeroes	ctext_aes256_cbc_16_zeroes	16	AES - 256 - CBC - No Padding
16_zeroes	32_zeroes	ctext_aes256_cbc_pad_16_zeroes	16	AES - 256 - CBC - Padding
64_incr	32_incr	ctext_aes256_ecb_64_incr	16	AES - 256 - ECB - No Padding
64_incr	32_incr	ctext_aes256_cbc_64_incr	16	AES - 256 - CBC - No Padding
60_incr	32_incr	ctext_aes256_cbc_pad_60_incr	16	AES - 256 - CBC - Padding

© Copyright 2012 Apple Inc.

3DES

Plain-Text (File)	Raw Key Bits (File)	Cipher Text (File)		Description
8_zeroes	24_zeroes	ctext_3des_ecb_8_zeroes	----	3DES - ECB - No Padding
8_zeroes	24_zeroes	ctext_3des_cbc_8_zeroes	----	3DES - CBC - No Padding
8_zeroes	24_zeroes	ctext_3des_cbc_pad_8_zeroes	----	3DES - CBC - Padding
64_incr	24_incr	ctext_3des_ecb_64_incr	----	3DES - ECB - No Padding
64_incr	24_incr	ctext_3des_cbc_64_incr	----	3DES - CBC - No Padding
60_incr	24_incr	ctext_3des_cbc_pad_60_incr	----	3DES - CBC - Padding

SHA-1

Plain-Text (File)		Cipher Text (File)		Description
8_zeroes	----	digest_sha1_8_zeroes	----	SHA1 -
24_incr	----	digest_sha1_24_incr	----	SHA1 -
60_incr	----	digest_sha1_60_incr	----	SHA1 -

SHA-224

Plain-Text (File)		Cipher Text (File)		Description
8_zeroes	----	digest_sha224_8_zeroes	----	SHA224 -
24_incr	----	digest_sha224_24_incr	----	SHA224 -
60_incr	----	digest_sha224_60_incr	----	SHA224 -

SHA-256

Plain-Text (File)		Cipher Text (File)		Description
8_zeroes	----	digest_sha256_8_zeroes	----	SHA256 -
24_incr	----	digest_sha256_24_incr	----	SHA256 -
60_incr	----	digest_sha256_60_incr	----	SHA256 -

© Copyright 2012 Apple Inc.

SHA-384

Plain-Text (File)		Cipher Text (File)		Description
8_zeroes	----	digest_sha384_8_zeroes	----	SHA384 -
24_incr	----	digest_sha384_24_incr	----	SHA384 -
60_incr	----	digest_sha384_60_incr	----	SHA384 -

SHA-512

Plain-Text (File)		Cipher Text (File)		Description
8_zeroes	----	digest_sha512_8_zeroes	----	SHA512 -
24_incr	----	digest_sha512_24_incr	----	SHA512 -
60_incr	----	digest_sha512_60_incr	----	SHA512 -

HMAC/SHA-1

Plain-Text (File)	Raw Key Bits (File)	Cipher Text (File)		Description
8_zeroes	8_zeroes	hmac_sha1_8_zeroes	----	HMAC/SHA1 -
60_incr	4_incr	hmac_sha1_4_incr	----	HMAC/SHA1 -
60_incr	6_incr	hmac_sha1_6_incr	----	HMAC/SHA1 -
60_incr	16_incr	hmac_sha1_16_incr	----	HMAC/SHA1 -
60_incr	24_incr	hmac_sha1_24_incr	----	HMAC/SHA1 -

© Copyright 2012 Apple Inc.

RSA

Plain-Text (File)	Raw Key Bits (File)	Cipher Text (File)		Signature Algorithm
8_zeroes	rsa1024_priv.der	sig_rsa_8_zeroes	----	RSA/SHA-1
60_incr	rsa1024_priv.der	sig_rsa_60_incr		RSA/SHA-1
60_incr	rsa1024_priv.der	sig_rsa_sha224_60_incr		RSA/SHA-224
60_incr	rsa1024_priv.der	sig_rsa_sha256_60_incr		RSA/SHA-256
60_incr	rsa1024_priv.der	sig_rsa_sha384_60_incr		RSA/SHA-384
60_incr	rsa1024_priv.der	sig_rsa_sha512_60_incr		RSA/SHA-512
128_incr	rsa1024_pub.der	ctext_rsa1024_incr		RSA1024
128_rand	rsa1024_pub.der	ctext_rsa1024_rand		RSA1024

© Copyright 2012 Apple Inc.