Oracle® Solaris 11.3 での FIPS 140-2 対応システムの使用

2017 年 3 月

Part No: E62764

この記事では、Oracle Solaris システムを、暗号化のカーネルレベルとユーザーレベルのコンシューマ (Kerberos、SunSSH、Apache HTTP Server など) に FIPS 140-2 レベル 1 暗号化を提供するように構成する方法について説明します。プロバイダとコンシューマを有効にする方法について説明するとともに、SunSSH と Apache HTTP Server を FIPS 140-2 モードで動作できるようにする例が含まれています。

Oracle Solaris での FIPS 140-2 レベル 1 暗号化の概要

2016 年 8 月、米国商務省国立標準技術研究所 (NIST) は、Oracle Solaris の暗号化フレームワーク機能を FIPS 140-2 レベル 1 標準に対して検証する 2 つの証明書を発行しました。これらの Oracle Solaris 証明書は 2698 および 2699 の番号が付けられ、Oracle Solaris 11.3 SRU 5.6 リリースに基づいています。

Oracle Solaris 11.3 上で動作する OpenSSL モジュールは 2013 年 11 月に FIPS 140-2 で検証され、証明書 1747 が発行されました。FIPS 140-2 で検証された OpenSSL を暗号化として使用するアプリケーションはすべて、このモジュールを使用できます。これらの証明書へのリンクについては、16 ページの「Oracle Solaris システムでの FIPS 140-2 レベル 1 証明書のリファレンス」を参照してください。

FIPS 140-2 (米国連邦情報処理標準) は、機密情報ではあるが、機密扱いを受けていない情報を処理する多くの規制産業および米国政府機関にとっての要件です。FIPS 140-2 の目的は、そのシステムが暗号化を正しく実装しているという一定程度の保証を提供することにあります。コンピュータシステム上で FIPS 140-2 レベル 1 暗号化を提供することを「FIPS 140-2 モードでの動作」と呼びます。

アプリケーションと FIPS 140-2

FIPS 140-2 モードで動作しているシステムは、FIPS 140-2 暗号化の少なくとも 1 つのプロバイダを有効にしています。一部のアプリケーション (コンシューマ) は、自動的に FIPS 140-2 暗号化を呼び出します (`passwd` コマンドなど)。一部のアプリケーションは、動的に FIPS 140-2 暗号化プロバイダを呼び出します (SunSSH など)。その他のアプリケーションは、プロバイダが有効であり、管理者が FIPS 140-2 暗号化 (Kerberos、IPsec、Apache HTTP Server など) のみが使用されるようにアプリケーションを構成した場合に FIPS 140-2 モードで実行されます。

2013 年 12 月検証からの FIPS 140-2 2016 年 8 月検証の変更

2013 年 12 月から 2016 年 8 月までの間に、NIST では FIPS 140-2 暗号化およびハードウェアの要件が更新されました。これらの更新によって、Oracle Solaris の暗号フレームワーク機能で複数の項目の検証ステータスが変更されました。

2016 年 8 月 FIPS 140-2 検証では、次のメカニズムのステータスが変更されました。

- SHA512/224 が承認されています。

- SHA512/256 が承認されています。
- AES-GMAC は承認されていません。
- `libucrypto` の SHA1 と HMAC-SHA1 が承認されています。
- PKCS #11 ソフトトークンキーストアの SHA1 と HMAC-SHA1 は承認されていません。

2013 年 12 月の証明書の場合と異なり、ソフトウェア検証が特定のハードウェアに関連付けられなくなりました。承認されているハードウェアのリストについては、17 ページの「FIPS 140-2 用に検証されている Oracle Solaris システムハードウェア」を参照してください。

Oracle Solaris システム上での FIPS 140-2 プロバイダの有効化

FIPS 140-2 プロバイダモジュールは CPU を集中的に使用するため、デフォルトでは有効になっていません。管理者には、FIPS 140-2 モードでこれらのプロバイダを有効にし、コンシューマを構成する責任があります。

Oracle Solaris OS では、FIPS 140-2 レベル 1 で検証された、暗号化アルゴリズムの 2 つのプロバイダが提供されます。

- Oracle Solaris の暗号化フレームワーク機能は Oracle Solaris システム上の中央の暗号化ストアであり、2 つの FIPS 140-2 モジュールを提供します。ユーザーランドモジュールは、ユーザー空間で動作するアプリケーションに暗号化を提供し、カーネルモジュールは、カーネルレベルのプロセスに暗号化を提供します。使用可能な場合、両方のモジュールが SPARC および x86 プロセッサのアルゴリズムアクセラレーションを活用できます。
  - Oracle Solaris ユーザーランド暗号化フレームワークモジュールでは、その呼び出し元であるアプリケーション用の暗号化が提供されます。このモジュールでは、暗号化、復号化、ハッシュ処理、セキュアな乱数生成、署名の生成と検証、証明書の生成と検証、メッセージ認証機能、および RSA と DSA 用の鍵ペアの生成が提供されます。ユーザーランド暗号化フレームワーク の呼び出し元であるユーザーレベルのアプリケーション (`passwd` コマンドや IKEv2 など) は、FIPS 140-2 モードで実行されます。
  - Oracle Solaris カーネル暗号化フレームワークモジュールでは、カーネルモジュール用の暗号化が提供されます。このモジュールでは、暗号化、復号化、ハッシュ処理、セキュアな乱数生成、署名の生成と検証、およびメッセージ認証機能が提供されます。カーネルレベルのコンシューマ, (Kerberos や IPsec など) は、独自の API を使用してカーネル暗号化フレームワークを呼び出します。
- OpenSSL オブジェクトモジュールは、SunSSH および Web アプリケーションに暗号化を提供します。

  OpenSSL は、Secure Sockets Layer (SSL v2/v3) および Transport Layer Security (TLS v1) プロトコルのためのオープンソースのツールキットであり、暗号化ライブラリを提供します。Oracle Solaris では、SunSSH と Apache HTTP Server は OpenSSL FIPS 140-2 モジュールを使用し、その利点を活用できます。Apache HTTP Server はまた、暗号化フレームワークを使用して FIPS 140-2 モードで動作することもできます。

  Oracle Solaris 11.3 には、コードが FIPS 140-2 をサポートしているすべてのコンシューマで使用できる OpenSSL の FIPS 140-2 バージョンが含まれています。
  - SunSSH は、Oracle Solaris 11.3 に含まれている OpenSSL の FIPS 140-2 バージョンを使用します。
  - Apache HTTP Server Version 2.4 は、Oracle Solaris 11.3 に含まれている OpenSSL の FIPS 140-2 バージョンを使用できます。

    Apache HTTP Server Version 2.2 は、FIPS 140-2 をサポートしていない OpenSSL のバージョンを使用します。Version 2.2 を FIPS 140-2 モードで実行するには、OpenSSL ではなく PKCS #11 エンジンを使用します。

Oracle Solaris で FIPS 140-2 プロバイダを有効にする方法

FIPS 140-2 モードでプロバイダを有効にし、アプリケーションでそれらのプロバイダを使用できるようにする例については、8 ページの「Oracle Solaris 11.3 SRU 5.6 システム上で FIPS 140-2 モードで実行する例」を参照してください。

- 暗号化フレームワークを FIPS 140-2 モードで実行するには、『Oracle Solaris 11.3 での暗号化と証明書の管理』 の 「FIPS 140-2 が有効になったブート環境を作成する方法」を参照してください。
- OpenSSL を FIPS 140-2 モードで実行するには、『Oracle Solaris 11.3 での暗号化と証明書の管理』 の 「Oracle Solaris での OpenSSL のサポート」を参照してください。

FIPS 140-2 モードの暗号化フレームワークについて

暗号化フレームワークは、さまざまな鍵の長さを持つ多くの暗号化アルゴリズムを実装します。アルゴリズムの各バリアントは、メカニズムと呼ばれます。すべてのメカニズムが FIPS 140-2 で検証されているわけではありません。

FIPS 140-2 モードで動作しているとき、ユーザーランド暗号化フレームワークは、FIPS 140-2 で承認されたアルゴリズムの使用を強制しません。この設計の選択により、ユーザー独自のセキュリティーポリシーを適用できます。

ヒント - 従来のシステム、準拠していないアプリケーション、または問題の解決に対応するために、すべての暗号化フレームワークアルゴリズムを有効なままにしておくことができます。FIPS 140-2 モードの厳格な強制のために、暗号化フレームワークでの FIPS 140-2 以外のアルゴリズムを無効にできます。例については、8 ページの「Oracle Solaris 11.3 SRU 5.6 システム上で FIPS 140-2 モードで実行する例」の最後のステップを参照してください。

FIPS 140-2 モードでプロバイダを有効にしたあと、FIPS 140-2 アルゴリズムを使用するようにアプリケーションとプログラムを構成する必要があります。

`cryptoadm` および `pktool` コマンドは、暗号化フレームワークがサポートするアルゴリズムを一覧表示します。

- 暗号化メカニズムの完全なリストを表示するには、`cryptoadm list -vm` コマンドを使用します。`cryptoadm(1M)` のマニュアルページを参照してください。
- ECC アルゴリズムの曲線のリストを表示するには、`pktool gencert listcurves` コマンドを使用します。`pktool(1)` のマニュアルページを参照してください。

  Oracle Solaris に対して FIPS 140-2 で検証された Oracle Solaris での ECC 曲線については、14 ページの「暗号化フレームワークでの FIPS 140-2 アルゴリズム」を参照してください。
- 暗号化フレームワークに対して検証された FIPS 140-2 アルゴリズムについては、16 ページの「Oracle Solaris システムでの FIPS 140-2 レベル 1 証明書のリファレンス」に一覧表示されている Oracle Solaris セキュリティーポリシーを確認してください。カーネル暗号化フレームワーク と ユーザーランド暗号化フレームワーク では、サポートされているアルゴリズムが若干異なります。

Oracle Solaris での FIPS 140-2 モードの OpenSSL について

FIPS 140-2 モードで動作している OpenSSL では、FIPS 140-2 で検証されたアルゴリズムが強制的に使用されます。そのため、OpenSSL を FIPS 140-2 モードで使用するアプリケーションは、FIPS 140-2 アルゴリズムにしかアクセスできません。

詳細および例については、次を参照してください。

- 『Oracle Solaris 11.3 での暗号化と証明書の管理』 の 「Oracle Solaris での OpenSSL のサポート」
- Oracle Solaris 11.2 上での OpenSSL (`http://blogs.oracle.com/observatory/entry/openssl_on_solaris_11_2`)
- `openssl`(5) のマニュアルページ

注記 - FIPS 140-2 モードでの OpenSSL の構成の例については、8 ページの「Oracle Solaris 11.3 SRU 5.6 システム上で FIPS 140-2 モードで実行する例」を参照してください。

ハードウェアアクセラレーションと FIPS 140-2 のパフォーマンス

最高のパフォーマンスを得るには、FIPS 140-2 プロバイダのコンシューマは可能な場合、ハードウェアで高速化された暗号化を使用するようにしてください。暗号化フレームワークは、17 ページの「FIPS 140-2 用に検証されている Oracle Solaris システムハードウェア」に一覧表示されているシステム上で、ハードウェアアクセラレーションとともに FIPS 140-2 モードで実行されます。

OpenSSL を FIPS 140-2 モードで実行している場合に SPARC T4 または SPARC T5 サーバー上でハードウェアアクセラレーションを実現するには、`pkcs11` エンジンを使用します。

注記 - SPARC システムでは、Oracle Solaris 11.3 リリースに含まれている OpenSSL の FIPS 140-2 バージョンはアセンブリ言語の最適化を利用しますが、ハードウェアアクセラレーションは利用しません。Intel システムでは、Oracle Solaris 11.3 に含まれている OpenSSL の FIPS 140-2 バージョンは、AES-NI のハードウェアアクセラレーションとアセンブリ言語の最適化を利用します。

詳細は、『Oracle Solaris 11.3 での暗号化と証明書の管理』 の 「SPARC T4 システムの暗号化の最適化」を参照してください。例については、8 ページの「Oracle Solaris 11.3 SRU 5.6 システム上で FIPS 140-2 モードで実行する例」を参照してください。

Oracle Solaris システム上での FIPS 140-2 コンシューマの有効化

FIPS 140-2 モードで動作するには、FIPS 140-2 対応システム上のアプリケーションは、米国政府が Oracle Solaris 上で FIPS 140-2 モードで検証したアルゴリズムを使用する必要があります。FIPS 140-2 プロバイダが有効になっている場合、一部のコンシューマは、デフォルトで FIPS 140-2 アルゴリズムを使用します (`passwd` コマンドなど)。その他のコンシューマでは、構成で FIPS 140-2 アルゴリズムのみを使用する必要があります。

管理者には、Oracle Solaris に対して検証された FIPS 140-2 アルゴリズムを使用するようにコンシューマを構成し、無効なアルゴリズムを避ける責任があります。次のガイドラインに従ってください。

- Oracle Solaris 上で使用できるが、Oracle Solaris の FIPS 140-2 検証に含まれていないアルゴリズム (2 つの鍵のトリプル DES など) は避けてください。
- Oracle Solaris の FIPS 140-2 証明書に含まれているが、鍵の長さが FIPS 140-2 に必要な長さより短いアルゴリズムは避けてください (1024 ビット RSA など)。

- Oracle Solaris の FIPS 140-2 証明書に含まれているが、コンシューマで使用できないアルゴリズム (IKEv2 での Koblitz 曲線に基づく楕円曲線暗号化 (ECC) など) は避けてください。IKEv2 は、素数に基づく ECC のみをサポートしています。
- Oracle Solaris の FIPS 140-2 証明書に含まれていないが、暗号化フレームワークには含まれているアルゴリズム (`MD5` 対称鍵アルゴリズムや、その他の対称アルゴリズムの弱いバージョンなど) は避けてください。
- コンシューマでその他のアルゴリズムを使用できる場合でも、FIPS 140-2 アルゴリズムのみを指定してください。多くのコンシューマがこのカテゴリに分類されます。

注記 - アプリケーションが、FIPS 140-2 で検証されたアルゴリズムを使用できないモジュール (インターネット鍵交換プロトコルバージョン 1 (IKEv1) や OpenSSH など) を使用している場合、FIPS 140-2 システム上ではこれらのアプリケーションを実行しないようにしてください。

FIPS 140-2 コンシューマとしての Apache HTTP Server

Oracle Solaris 11.3 は、Apache HTTP Server の 2 つのバージョンを提供します。Version 2.4 はパッケージ `pkg:/web/server/apache-24` としてインストールされ、Version 2.2 はパッケージ `pkg:/web/server/apache-22` としてインストールされます。FIPS 140-2 モードで動作するには、Version 2.4 は、FIPS 140-2 OpenSSL プロバイダまたは PKCS #11 エンジンオプションを使用できます。Version 2.2 は、暗号化フレームワークである PKCS #11 エンジンオプションを使用する必要があります。

注記 - 各バージョンを別のポート上で待機するように構成した場合は、Web サーバーの両方のバージョンを FIPS 140-2 モードで実行できます。

暗号化フレームワーク (`pktool gencert` コマンド) または OpenSSL の FIPS 140-2 バージョン (`openssl -newkey` コマンド) を使用して、Web サーバー証明書を生成できます。

構成ステップについては、8 ページの「Oracle Solaris 11.3 SRU 5.6 システム上で FIPS 140-2 モードで実行する例」を参照してください。

関連項目:

- `openssl`(1openssl) のマニュアルページ
- `openssl`(5) のマニュアルページ
- 『Oracle Solaris 11.3 でのネットワークのセキュリティー保護』 の 「SSL カーネルプロキシ を使用するように Apache 2.2 Web サーバーを構成する方法」
- `ksslcfg`(1M) のマニュアルページ

FIPS 140-2 コンシューマとしての SunSSH

Oracle Solaris 11.3 は、Secure Shell の 2 つのバージョンである SunSSH と OpenSSH を提供します。FIPS 140-2 モードで動作できるのは、Secure Shell の SunSSH バージョンだけです。

管理者は、明示的に SunSSH を FIPS 140-2 モードで動作できるようにする必要があります。FIPS 140-2 モードでは、SunSSH は FIPS 140-2 で検証されていないアルゴリズムを使用するように構成されている場合、エラーで失敗します。手順については、『Oracle Solaris 11.3 での Secure Shell アクセスの管理』 の 「SunSSH と FIPS 140-2」を参照してください。この手順には、検証された FIPS 140-2 アルゴリズムのリストが含まれています。

サンプル構成については、8 ページの「Oracle Solaris 11.3 SRU 5.6 システム上で FIPS 140-2 モードで実行する例」を参照してください。

関連項目:

- sshd(1M) および ssh(1) のマニュアルページ
- sshd_config(4) および ssh_config(4) のマニュアルページ
- ssh-keygen(1) のマニュアルページ

FIPS 140-2 コンシューマとしての IPsec と IKEv2

IP セキュリティーアーキテクチャー (IPsec) は、IPv4 および IPv6 ネットワークで IP パケットを暗号化して保護します。インターネット鍵管理 (IKE) は、IPsec のための自動化された鍵管理を提供します。Oracle Solaris では、IPsec はカーネル暗号化フレームワークのコンシューマであり、IKE バージョン 2 (IKEv2) はユーザーランド暗号化フレームワークのコンシューマです。IPsec および IKE 管理者には、IPsec とともに IKEv2 を使用し、Oracle Solaris に対して検証された FIPS 140-2 アルゴリズムを選択する責任があります。

注記 - IKEv1 では、FIPS 140-2 用に検証された暗号化フレームワークの暗号化アルゴリズムが使用されません。そのため、IKEv1 を FIPS 140-2 モードで実行しないようにしてください。

関連項目:

- 『Oracle Solaris 11.3 でのネットワークのセキュリティー保護』 の 「IPsec を使用してほかのサーバーとの Web サーバー通信を保護する方法」
- 『Oracle Solaris 11.3 でのネットワークのセキュリティー保護』 の 「自己署名付き公開鍵証明書により IKEv2 を構成する方法」
- 『Oracle Solaris 11.3 でのネットワークのセキュリティー保護』 の 「ハードウェア上で IKEv2 の公開鍵証明書を生成および格納する方法」
- ipsecconf(1M)、ikev2cert(1M)、ikev2.config(4)、および pktool(1) のマニュアルページ

FIPS 140-2 コンシューマとしての Kerberos

Kerberos クライアントはパッケージ `pkg:/service/security/kerberos-5` としてインストールされ、KDC マネージャーはパッケージ `pkg:/system/security/kerberos-5` としてインストールされます。Kerberos 管理者には、Kerberos サーバー、Kerberos データベース、および Kerberos クライアントで、Oracle Solaris に対して検証された FIPS 140-2 アルゴリズムを使用できるようにする責任があります。

KDC データベースと Kerberos クライアントに使用する暗号化タイプは、複数の Kerberos 構成ファイルで指定されます。これらのファイルでは、FIPS 140-2 暗号化タイプのみが使用され、弱い鍵が許可されないように Kerberos を構成する必要があります。

手順については、『Oracle Solaris 11.3 での Kerberos およびその他の認証サービスの管理』 の 「FIPS 140-2 モードで実行するように Kerberos を構成する方法」を参照してください。

関連項目:

- kdc.conf(4) および krb5.conf(4) のマニュアルページ
- kdb5_util(1M) および krb5kdc(1M) のマニュアルページ

FIPS 140-2 コンシューマとしての鍵管理フレームワーク

鍵管理フレームワーク (KMF) は、Oracle Solaris で暗号化鍵と暗号化ポリシーを管理します。`pktool` は、対称鍵と非対称鍵を作成するための KMF コマンドです。KMF 管理者には、

Oracle Solaris に対して検証された FIPS 140-2 アルゴリズムを選択する責任があります。『Oracle Solaris 11.3 での暗号化と証明書の管理』 の 「pktool gencert コマンドを使って証明書を作成する方法」の例および pktool(1) のマニュアルページを参照してください。

FIPS 140-2 コンシューマとしての passwd コマンド

passwd コマンドは、ユーザーランド暗号化フレームワークのコンシューマです。2 つの構成ファイル /etc/security/crypt.conf と /etc/security/policy.conf によって、システムがどのパスワードハッシュを使用するかが決定されます。

passwd コマンドは、PAM モジュール pam_authtok_store.so.1 と pam_unix_auth.so.1 を使用して crypt() 関数を呼び出します。crypt() 関数は、crypt.conf ファイル内のエントリに基づいて、メッセージダイジェストライブラリ libmd() からパスワードハッシュのプラグインを動的にロードします。使用可能なプラグインには、SHA256、SHA512、および MD5 が含まれます。policy.conf ファイルには、許可されているプラグインが一覧表示されています。デフォルトでは、policy.conf ファイルで MD5 を使用することが許可されません。

注記 - /etc/security/policy.conf ファイル内の暗号化パスワードハッシュポリシーは、パスワードハッシュとして Blowfish を使用するシステムとの相互運用性を拡張します。FIPS 140-2 セキュリティーを向上させるには、policy.conf ファイル内の CRYPT_ALGORITHMS_ALLOW=2a,5,6 エントリから Blowfish アルゴリズム (2a) を削除してください。

例については、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』 の 「信頼できるユーザーのログインの作成」および『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』 の 「役割の作成」を参照してください。

関連項目:

- crypt(3C) および libmd(3LIB) のマニュアルページ
- crypt.conf(4) および policy.conf(4) のマニュアルページ
- passwd(1) および passwd(4) のマニュアルページ

FIPS 140-2 コンシューマとしての encrypt、decrypt、digest、および mac コマンド

ユーザーコマンド encrypt、decrypt、digest、および mac は、暗号化フレームワークのコンシューマです。サイトのセキュリティーチームは、通常のユーザーが検証された鍵の長さの FIPS 140-2 アルゴリズムを選択するように指導してください。

例については、次を参照してください。

- 『Oracle Solaris 11.3 での暗号化と証明書の管理』 の 「暗号化フレームワークによるファイルの保護」
- encrypt(1)、decrypt(1)、digest(1)、および mac(1) のマニュアルページ

Oracle Solaris 11.3 SRU 5.6 システム上で FIPS 140-2 モードで実行する例

このセクションの例では、Apache HTTP Server Version 2.4 を FIPS 140-2 モードで実行するように Oracle Solaris システムを構成します。システムは SPARC T5-2 サーバーであり、これにより SPARC5 プロセッサでの暗号化アクセラレーションが提供されます。

主な手順は次のとおりです。

1. FIPS 140-2 レベル 1 を構成する BE を作成し、そこにブートします。
2. 新しい BE で、FIPS 140-2 プロバイダを有効にします。
3. FIPS 140-2 モードで 2 つのコンシューマ SunSSH と Apache HTTP Server Version 2.4 を有効にします。
4. `policy.conf` ファイルを変更して、FIPS 140-2 パスワードハッシュを使用しないシステムとの相互運用性を削除します。
5. FIPS 140-2 BE にブートします。
6. テストします。

次の例では、この構成を実現するために実行する詳細なアクションについて説明します。

1. 現在の構成に基づいて BE を作成し、それをブートします。

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

2. 新しい BE で、暗号化フレームワークでの FIPS 140-2 モードを有効にします。

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

`fips-140` パッケージがまだロードされていない場合は、このコマンドでそのパッケージもロードされます。

3. OpenSSL の FIPS 140-2 モードバージョンを有効にします。

   a. OpenSSL の FIPS 140-2 モードバージョンがシステム上に存在することを確認します。

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

注意 - OpenSSL モジュールに切り替えようとしたときに、そのモジュールがシステム上に存在しないと、システムを使用できなくなる可能性があります。

   b. FIPS 140-2 OpenSSL プロバイダを有効にします。

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

4. SunSSH コンシューマを構成し、FIPS 140-2 モードで有効にします。

注記 - OpenSSH を Oracle Solaris 上で FIPS 140-2 モードで動作するように構成することはできません。

   a. FIPS 140-2 モードを使用するには、`sshd_config` および `ssh_config` ファイルの最後に次の情報を追加します。

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

   b. FIPS 140-2 モードで SunSSH で使用する PKCS #8 形式の非公開鍵を生成します。

      Oracle Solaris 11 上で Secure Shell のための X.509 を設定する方法 (`http://www.oracle.com/technetwork/articles/servers-storage-admin/howto-setup-x509-sunssh-`

1929594.html)の手順に従います。次に、ssh-keygen コマンドを使用して非公開鍵を作成します。

ssh-keygen コマンドを使用する場合、デフォルトの鍵の長さは 1024 であり、これは検証された長さではありません。-b オプションを使用して、有効な鍵の長さを指定する必要があります。

5. FIPS 140-2 モードで Apache HTTP Server を構成します。

   a. 検証された鍵の長さで FIPS 140-2 アルゴリズムを使用して、Web サーバー証明書を生成します。

      たとえば、pktool コマンドを使用し、2048 ビットの RSA 鍵と SHA-384 ハッシュを指定します。

```
# 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. ssl.conf 構成ファイルを作成します。

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

   c. わかりやすくするために、FIPS 140-2 モードでの OpenSSL の使用についてコメントします。

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

注記 - Apache HTTP Server Version 2.2 を構成していた場合、SSLCryptoDevice の値は pkcs11 になります。

---

   d. その他の鍵情報がサイトポリシーに対して正しく構成されていることを確認します。

```
# 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. Web サーバーのサイト構成を完了します。

      たとえば、SSL プロトコルバージョンを指定します。

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

6. 許容可能なハッシュとして 2a を削除することにより、FIPS 140-2 以外のパスワードハッシュが使用されないようにします。

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

7. (オプション) すべてのログインが正しいハッシュを使用することを確認します。

   a. この BE にログインできるすべてのユーザーを一覧表示します。

```
# 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 Administrator:/home/admin:/usr/bin/bash:PS ...
```

ヒント - LDAP リポジトリ内のすべてのユーザーを検索するには、`-S ldap` オプションを使用します。

---

b. 各ユーザーにログイン時に新しいパスワードを強制的に作成させます。

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

ヒント - すべてのユーザーにログイン時にパスワードを強制的に変更させるスクリプトを記述できます。

---

8. コンシューマが構成されたら、BE をリブートします。

```
# reboot
```

9. 構成をテストします。

- プロバイダが FIPS 140-2 モードで動作していることを確認します。

  次の出力は、暗号化フレームワークが FIPS 140-2 モードで動作していることを示します。

```
# 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.
```

  次の出力は、OpenSSL が FIPS 140-2 モードで動作していることを示します。

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

- Apache HTTP Server の暗号化の使用をトレースします。

  a. 端末ウィンドウで、Apache HTTP Server Version 2.4 プロセス上の OpenSSL 暗号化呼び出しをトレースします。

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

注記 - このコマンドは、`/usr/lib/libcrypto.so.1` ライブラリへの FIPS 140-2 エンベロープ (evp) 関数呼び出しをトレースします。

---

b. Web サーバー要求を送信し、その出力で FIPS 140-2 エンベロープの使用を確認します。

```
# 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
...
```

- FIPS 140-2 以外のシステムと FIPS 140-2 システムから Secure Shell ログインをテストします。
- ログファイルで Secure Shell と Apache HTTP Server にエラーがないか確認します。

  FIPS 140-2 アルゴリズムが使用されていない場合、Secure Shell はエラーを返します。

10. (オプション) FIPS 140-2 以外のアルゴリズムがすべての暗号化フレームワークコンシューマで使用されないようにするには、FIPS 140-2 以外のメカニズムを無効にします。

ヒント - 暗号化フレームワークコンシューマの厳格なポリシーを実装するには、そのポリシーを実装するスクリプトを作成してから、FIPS 140-2 モードの厳格なポリシーバージョンのための 2 番目の BE を作成します。

次の一連のコマンドにより、FIPS 140-2 モードで検証されていないカーネルアルゴリズムが使用されなくなります。

```
# cryptoadm -vm      /** 短縮リストには、非 FIPS 140 アルゴリズムのメカニズムのみが表示されます **/
...
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
```

次のコマンドは、FIPS 140-2 以外のメカニズムを無効にしたあとのカーネル暗号化フレームワークプロバイダのポリシーを表示します。

```
# 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.
```

ユーザーランドメカニズムが使用されないようにするには、プロバイダとして `/usr/lib/security/$ISA/pkcs11_softtoken.so` を指定したあと、それらのメカニズムを指定します。ユーザーランドのメカニズムを一覧表示するには、次のコマンドを使用します。

```
# 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  ...
```

たとえば、次のコマンドは、ユーザーランドの Camellia メカニズムを無効にします。

```
# 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.
```

注意 - 厳格なポリシーのための BE は、本番環境で使用する前に徹底的にテストしてください。

11. FIPS 140-2 モードの使用を停止するには、元の BE をアクティブ化してリブートします。

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

Oracle Solaris システムでの FIPS 140-2 アルゴリズムのリストと証明書のリファレンス

このセクションでは、FIPS 140-2 モードで使用できるアルゴリズムと、避けるべきアルゴリズムのリストを示します。

注記 - これらのリストは、利便性のためにのみ提供されています。正式な米国 FIPS 140-2 認定およびガイドラインドキュメントが最終的なソースです。

暗号化フレームワークでの FIPS 140-2 アルゴリズム

暗号化フレームワークのコンシューマが FIPS 140-2 で検証されたアルゴリズムを確実に使用するようにするには、検証されたアルゴリズム、モード、および鍵の長さの次のサマリーのアルゴリズムを選択します。

アルゴリズムの最終的なリストについては、16 ページの「Oracle Solaris システムでの FIPS 140-2 レベル 1 証明書のリファレンス」にあるセキュリティーポリシーのリファレンスを確認してください。

注記 - アルゴリズムの鍵の長さが重要になる場合があります。鍵の長さが短いと、FIPS 140-2 で検証されない可能性があります。

- AES – 次のモードと鍵の長さを持つもののみ。
  - CBC モード – 128 ビット、192 ビット、および 256 ビットの鍵の長さ
  - CCM モード – 128 ビット、192 ビット、および 256 ビットの鍵の長さ
  - CFB モード – 128 ビットの鍵の長さ
  - CTR モード – 128 ビット、192 ビット、および 256 ビットの鍵の長さ
  - ECB モード – 128 ビット、192 ビット、および 256 ビットの鍵の長さ
  - GCM モード – 128 ビット、192 ビット、および 256 ビットの鍵の長さ
  - XTS モード - 128 ビットおよび 256 ビットの鍵の長さ (データストレージのみ)
- 3DES – 鍵オプション 1 の CBC および ECB モードで。
- Diffie-Hellman - 鍵合意で使用されます (2048 ビットから 5012 ビットまでの鍵の長さ、ユーザーランド暗号化フレームワークのみ)。
- 楕円曲線 Diffie-Hellman - 鍵合意での使用が許可されます (2048 ビットから 5012 ビットまでの鍵の長さ、ユーザーランド暗号化フレームワークのみ)。
- DSA – 2048 ビット以上の鍵の長さ。
- ECC – 次の曲線を持つもののみ。ECC によって ECDSA と ECDH が影響を受けます。最初の名前は NIST の名前であり、2 番目の名前は Oracle Solaris での同等の名前です。
  - 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 – バリアントはありません。
- HMAC SHA2 – 224 ビットから 512 ビットまでの鍵の長さ。
- ECDSA SHA1 - 署名検証。

- RSA – SHA1 では 2048 ビット以上の鍵の長さ、SHA2 では 256 ビットから 512 ビットまでの鍵の長さ。
- SHA1 – バリアントはありません。
- SHA2 – 224 ビットから 512 ビットまでの鍵の長さ。
- SHA512/224 – SHA-512 の切り詰められたバージョン。ここで、初期値は、2012 年 5 月の ITL 公報 (`http://csrc.nist.gov/publications/nistbul/may-2012_itl-bulletin.pdf`)で説明されている方法を使用して生成されます。
- SHA512/256 – SHA-512 の切り詰められたバージョン。ここで、初期値は、2012 年 5 月の ITL 公報 (`http://csrc.nist.gov/publications/nistbul/may-2012_itl-bulletin.pdf`)で説明されている方法を使用して生成されます。
- `swrand` - カーネル暗号化フレームワーク のソフトウェアエントロピーソース。カーネル暗号化フレームワークとユーザーランドの両方に、NIST で承認された DRBG (Deterministic Random Bit Generator) があります。NIST 特殊出版物 800-90A (`http://csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf`) を参照してください。
- `intelrd` - カーネル暗号化フレームワークのハードウェアエントロピープロバイダ。カーネル暗号化フレームワークとユーザーランドの両方に、NIST で承認された DRBG (Deterministic Random Bit Generator) があります。NIST 特殊出版物 800-90A (`http://csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf`) を参照してください。

FIPS 140-2 構成では、鍵の長さが指定された次のアルゴリズムが許可されます。

- RSA 鍵ラッピング - 112 ビットよりも長い鍵の長さが許可されます。
- Diffie-Hellman 鍵合意 - 112 ビットよりも長い鍵の長さが許可されます (ユーザーランド暗号化フレームワークのみ)。
- Elliptic Curve Diffie-Hellman (ECDH) 鍵合意 - 112 ビットよりも長い鍵の長さが許可されます (ユーザーランド暗号化フレームワークのみ)。

暗号化フレームワークでの FIPS 140-2 で承認されていないアルゴリズム

FIPS 140-2 モードでは、次のサマリーリストのアルゴリズムは使用できません(それが暗号化フレームワークで実装されていたり、またはほかのプロバイダに対して FIPS 140-2 で検証されたアルゴリズムである場合でも)。

アルゴリズムの最終的なリストについては、16 ページの「Oracle Solaris システムでの FIPS 140-2 レベル 1 証明書のリファレンス」にあるセキュリティーポリシーのリファレンスを確認してください。

- 2 つの鍵のトリプル DES - 80 ビットのセキュリティーのみを提供する弱いアルゴリズムです (3DES とも表記)。
- MD4 – 1990 年に Ronald Rivest によって開発されたメッセージダイジェストアルゴリズム (Message Digest Algorithm) 4 は、明らかに脆弱なアルゴリズムです。
- MD5 および HMAC MD5 - メッセージダイジェストアルゴリズム 5 は、TLS でのみ FIPS 140-2 モードで使用できます。

  1991 年に Ron Rivest によって開発された MD5 アルゴリズムは、128 ビットのハッシュ値を生成します。MD5 は一般に、データの整合性を検証するために使用されます。MD5 は、デジタルセキュリティーのための衝突耐性に依存する SSL 証明書やデジタル署名などのアプリケーションには適していません。
- RC4 – ARCFOUR または ARC4 とも呼ばれる RC4 は、インターネットトラフィックを保護するために Transport Layer Security (TLS) で、またワイヤレスネットワークをセキュリティー保護するために WEP で使用されるソフトウェアストリーム暗号化方式です。RC4 は、出力鍵ストリームの先頭が破棄されないか、または鍵がランダムでない場合は明らかに脆弱です。

- AES - 明示的に検証されていないモード (XCBC-MAC や CTS など)。
- Blowfish – 1993 年に Bruce Schneier によって設計された対称鍵ブロック暗号化方式。これは独自仕様ではありません。
- Camellia - 日本で開発され、AES との互換性があり、ソフトウェアとハードウェアの両方の実装 (低価格のスマートカードから高速のネットワークシステムまで) に適するように設計されています。
- DES – IBM によって開発されたデータ暗号化規格 (Data Encryption Standard) は、1977 年に米国連邦情報処理標準 (FIPS) として公開されました。今日のコンピューティング環境では、その 56 ビットの鍵の長さは弱くなっています。
- DSA 鍵生成 – 512 ビットおよび 1024 ビットの鍵の長さは弱くなっています。さらに長い鍵の長さが FIPS 140-2 で検証されています。
- DSA 署名生成 – 512 ビットおよび 1024 ビットの鍵の長さは弱くなっています。さらに長い鍵の長さが FIPS 140-2 で検証されています。
- DSA 署名検証 – 512 ビットの鍵の長さは弱くなっています。さらに長い鍵の長さが FIPS 140-2 で検証されています。
- RSA 署名生成 – 256 ビット、512 ビット、および 1024 ビットの鍵の長さは弱くなっています。さらに長い鍵の長さが FIPS 140-2 で検証されています。
- RSA 署名検証 – 256 ビットおよび 512 ビットの鍵の長さは弱くなっています。さらに長い鍵の長さが FIPS 140-2 で検証されています。
- RSA 鍵ラッピング - 112 ビット未満の鍵の長さでは弱くなっています。FIPS 140-2 では、さらに長い鍵の長さが許可されます。
- Diffie-Hellman - 112 ビット未満の鍵の長さでは弱くなっています。鍵合意では、さらに長い鍵の長さが許可されます (ユーザーランド暗号化フレームワークのみ)。
- ECDH - 112 ビット未満の鍵の長さでは弱くなっています。鍵合意では、さらに長い鍵の長さが許可されます (ユーザーランド暗号化フレームワークのみ)。

Oracle Solaris システムでの FIPS 140-2 レベル 1 証明書のリファレンス

Oracle Solaris システム上の FIPS 140-2 プロバイダのセキュリティーポリシーには、モジュールの仕様とインタフェースが記載され、FIPS 140-2 モードで動作するために検証された暗号化メカニズムの完全なリストが提供されています。

表 1 Oracle Solaris でのプロバイダモジュールの FIPS 140-2 証明書とセキュリティーポリシー

証明書	プロバイダモジュール	セキュリティーポリシー
2698	Oracle Solaris カーネル暗号化フレームワーク (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2016.htm#2698)	http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp2698.pdf
2699	Oracle Solaris ユーザーランド暗号化フレームワーク (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2016.htm#2699)	http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp2699.pdf
1747	OpenSSL FIPS オブジェクトモジュールバージョン 2	http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1747.pdf
1051	OpenSSL FIPS オブジェクトモジュールバージョン 1.2	http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1051.pdf

次の FIPS 140-2 標準ドキュメントと移行ドキュメントは、FIPS 140-2 プロセス、および非推奨または制限されたアルゴリズムとそのさらに弱いバリアントに関するガイダンスを提供します。

- NIST 標準: FIPS PUB 140-2 (http://csrc.nist.gov/groups/STM/cmvp/standards.html)

- 移行: 暗号化アルゴリズムおよび鍵の長さの使用を移行するための推奨事項 (http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf)

Oracle Solaris システムでの FIPS 140-2 レベル 1 証明書のリファレンス

Oracle Solaris システム上の FIPS 140-2 プロバイダのセキュリティーポリシーには、モジュールの仕様とインタフェースが記載され、FIPS 140-2 モードで動作するために検証された暗号化メカニズムの完全なリストが提供されています。

次の FIPS 140-2 標準ドキュメントと移行ドキュメントは、FIPS 140-2 プロセス、および非推奨または制限されたアルゴリズムとそのさらに弱いバリアントに関するガイダンスを提供します。

- NIST 標準: FIPS PUB 140-2 (http://csrc.nist.gov/groups/STM/cmvp/standards.html)
- 移行: 暗号化アルゴリズムおよび鍵の長さの使用を移行するための推奨事項 (http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf)

FIPS 140-2 用に検証されている Oracle Solaris システムハードウェア

次の Oracle Solaris システムハードウェアおよびプロセッサは、FIPS 140-2 に準拠しています。ハードウェアアクセラレーションの有無に関係なく、すべてのシステムが検証されています。

最終的なプラットフォームのリストについては、16 ページの「Oracle Solaris システムでの FIPS 140-2 レベル 1 証明書のリファレンス」にあるセキュリティーポリシーのリファレンスを確認してください。

- Oracle SPARC T4、T5、および T7 シリーズサーバー
- Oracle SPARC M5、M6、および M7 シリーズサーバー
- Oracle SPARC S7 シリーズサーバー
- Oracle Minicluster S7-2 エンジニアドシステム
- Oracle Netra SPARC T4-1B および T5-1B サーバー
- Oracle Sun Blade X3 および X4 シリーズサーバー
- Oracle Sun Server X3、X4、および X5 シリーズ
- Oracle Netra Server X3-2 および X5-2
- Oracle Server X6-2 および X6-2L
- 富士通 M10 シリーズサーバー

Oracle Solaris 11.3 での FIPS 140-2 対応システムの使用

Part No: E62764

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWeb サイト(http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc)を参照してください。

Oracle Supportへのアクセス

サポートをご契約のお客様には、My Oracle Supportを通して電子支援サービスを提供しています。詳細情報は(http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info) か、聴覚に障害のあるお客様は (http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs)を参照してください。

Part No: E62764