CHAPTER 35

# デジタル証明書の設定

この章では、デジタル証明書の設定方法について説明します。次の項目を取り上げます。



## デジタル証明書に関する情報

CA は、証明書要求の管理とデジタル証明書の発行を行います。デジタル証明書には、名前、シリアル番号、会社、部門、または IP アドレスなど、ユーザまたはデバイスを識別する情報が含まれます。デジタル証明書には、ユーザまたはデバイスの公開キーのコピーも含まれています。CA は、信頼できるサードパーティ（VeriSign など）の場合もあれば、組織内に設置したプライベート CA（インハウス CA）の場合もあります。

**ヒント** 証明書コンフィギュレーションおよびロード バランシングの例は、次の URL を参照してください。https://supportforums.cisco.com/docs/DOC-5964

この項では、次のトピックについて取り上げます。

# 公開キー暗号化

デジタル署名は、公開キー暗号化によってイネーブルになり、デバイスおよびユーザを認証する手段です。RSA 暗号化システムなどの Public Key Cryptography では、各ユーザは、公開キーと秘密キーの両方を含むキー ペアを使用します。これらのキーは、補足として機能し、一方で暗号化されたものは、もう一方で復号化できます。

簡単に言えば、データが秘密キーで暗号化されたとき、署名が形成されます。署名はデータに付加されて受信者に送信されます。受信者は送信者の公開キーをデータに適用します。データとともに送信された署名が、公開キーをデータに適用した結果と一致した場合、メッセージの有効性が確立されます。

このプロセスは、受信者が送信者の公開キーのコピーを持っていること、およびその公開キーが送信者になりすました別人のものではなく、送信者本人のものであることを受信者が強く確信していることに依存しています。

通常、送信者の公開キーは外部で取得するか、インストール時の操作によって取得します。たとえば、ほとんどの Web ブラウザでは、いくつかの CA のルート証明書がデフォルトで設定されています。VPN の場合、IKE プロトコルは IPsec のコンポーネントであり、デジタル署名を使用してピア デバイスを認証した後で、セキュリティ アソシエーションをセットアップできます。

# 証明書のスケーラビリティ

デジタル証明書がない場合、通信するピアごとに各 IPsec ピアを手動で設定する必要があります。そのため、ネットワークにピアを新たに追加するたびに、安全に通信するために各ピアで設定変更を行わなければなりません。

デジタル証明書を使用している場合、各ピアは CA に登録されます。2 つのピアは、通信を試みるときに、証明書とデジタル署名されたデータを交換して、相互の認証を行います。新しいピアがネットワークに追加された場合は、そのピアを CA に登録するだけで済みます。他のピアを修正する必要はありません。新しいピアが IPSec 接続を試みると、証明書が自動的に交換され、そのピアの認証ができます。

CA を使用した場合、ピアはリモート ピアに証明書を送り、公開キー暗号化を実行することによって、そのリモート ピアに対して自分自身を認証します。各ピアから、CA によって発行された固有の証明書が送信されます。このプロセスが機能を果たすのは、関連付けられているピアの公開キーが各証明書にカプセル化され、各証明書が CA によって認証され、参加しているすべてのピアによって CA が認証権限者として認識されるためです。このプロセスは、RSA 署名付きの IKE と呼ばれます。

ピアは、証明書が期限満了になるまで、複数の IPSec セッションに対して、および複数の IPSec ピア宛てに証明書を送り続けることができます。証明書が期限満了になったときは、ピアの管理者は新しい証明書を CA から入手する必要があります。

CA は、IPSec に参加しなくなったピアの証明書を無効にすることもできます。無効にされた証明書は、他のピアからは有効な証明書とは認識されなくなります。無効にされた証明書は CRL に記載され、各ピアは別のピアの証明書を受け取る前に、CRL をチェックします。

CA の中には、実装の一部として RA を持つものもあります。RA は CA のプロキシの役割を果たすサーバであるため、CA が使用できないときも CA 機能は継続しています。

# キー ペア

キー ペアとは、次の特性を持つ RSA キーです。

- RSA キーは SSH や SSL に使用できます。
- SCEP 登録は、RSA キーの証明書をサポートしています。

- キー生成では、RSA キーの最大キー係数は 2048 ビットです。デフォルト サイズは 1024 です。1024 ビットを超える RSA キー ペアを持つ ID 証明書を使用している複数の SSL 接続によって、ASA での CPU 使用率が高くなり、クライアントレス ログインが拒否される可能性があります。
- 署名操作でサポートされているキーの最大サイズは 4096 ビットです。
- 署名にも暗号化にも使用できる汎用 RSA キー ペアを生成することも、署名用と暗号化用に別々の RSA キー ペアを生成することもできます。SSL では署名用ではなく暗号化用のキーが使用されるので、署名用と暗号化用にキーを分けると、キーが公開される頻度を少なくすることができます。ただし、IKE では暗号化用ではなく署名用のキーが使用されます。キーを用途別に分けることで、キーの公開頻度が最小化されます。

# トラストポイント

トラストポイントを使用すると、CA と証明書の管理とトレースができます。トラストポイントとは、CA または ID ペアを表現したものです。トラストポイントには、CA の ID、CA 固有のコンフィギュレーション パラメータ、登録されている ID 証明書とのアソシエーションが含まれています。

トラストポイントの定義が完了したら、CA の指定を必要とするコマンドで、名前によってトラストポイントを参照できます。トラストポイントは複数設定できます。

**（注）** ASA に同じ CA を共有するトラストポイントが複数ある場合、CA を共有するトラストポイントのうち、ユーザ証明書の検証に使用できるのは 1 つだけです。CA を共有するどのトラストポイントを使用して、その CA が発行したユーザ証明書を検証するかを制御するには、**support-user-cert-validation** コマンドを使用します。

自動登録の場合は、登録 URL がトラストポイントに設定されている必要があり、また、トラストポイントが示す CA がネットワーク上で使用可能であり、SCEP をサポートしている必要があります。

キー ペアと、トラストポイントに関連付けられている発行済み証明書は、PKCS12 形式でエクスポートとインポートができます。この形式は、異なる ASA 上のトラストポイント コンフィギュレーションを手動でコピーする場合に便利です。

## 証明書の登録

ASA は、トラストポイントごとに 1 つの CA 証明書が必要で、セキュリティ アプライアンス自体には、トラストポイントで使用するキーのコンフィギュレーションに応じて 1 つまたは 2 つの証明書が必要です。トラストポイントが署名と暗号化に別々の RSA キーを使用する場合、ASA には署名用と暗号化用の 2 つの証明書が必要になります。署名用と暗号化用のキーが同じである場合、必要な証明書は 1 つだけです。

ASA は、SCEP を使用した自動登録と、base-64-encoded 証明書を直接端末に貼り付けられる手動登録をサポートしています。サイトツーサイト VPN の場合は、各 ASA を登録する必要があります。リモート アクセス VPN の場合は、各 ASA と各リモート アクセス VPN クライアントを登録する必要があります。

## SCEP 要求のプロキシ

ASA は、AnyConnect とサードパーティ CA 間の SCEP 要求のプロキシとして動作することができます。プロキシとして動作する場合に必要なのは CA が ASA からアクセス可能であることのみです。ASA のこのサービスが機能するには、ASA が登録要求を送信する前に、ユーザが AAA でサポートされているいずれかの方法を使用して認証されている必要があります。また、ホスト スキャンおよびダイナミック アクセス ポリシーを使用して、登録資格のルールを適用することもできます。

ASA では、AnyConnect SSL または IKEv2 VPN セッションでのみこの機能をサポートしています。これは、IOS CS、Windows Server 2003 CA、および Windows Server 2008 CA を含む、すべての SCEP 準拠 CA をサポートしています。

クライアントレス（ブラウザベース）でのアクセスは SCEP プロキシをサポートしていませんが、WebLaunch（クライアントレス起動 AnyConnect）はサポートしていません。

ASA では、証明書のポーリングはサポートしていません。

ASA はこの機能に対するロード バランシングをサポートしています。

# 失効チェック

証明書は発行されると、一定期間有効です。CA は、安全上の問題や名前またはアソシエーションの変更などの理由で、期限が切れる前に証明書を無効にすることがあります。CA は、無効になった証明書の署名付きリストを定期的に発行します。失効確認をイネーブルにすることにより、CA が認証にその証明書を使用するたびに、その証明書が無効にされていないかどうか、ASA によってチェックされます。

失効確認をイネーブルにすると、PKI 証明書検証プロセス時に ASA によって証明書の失効ステータスがチェックされます。これには、CRL チェック、OCSP、またはその両方が使用されます。OCSP は、最初の方式がエラーを返した場合に限り使用されます（たとえば、サーバが使用不可であることを示すエラー）。

CRL チェックを使用すると、ASA によって、無効になった（および失効解除された）証明書とその証明書シリアル番号がすべてリストされている CRL が取得、解析、およびキャッシュされます。ASA では CRL（認証局の失効リストとも呼ばれます）に基づいて証明書を検証します。検証は、ID 証明書から下位証明書チェーンの認証局までさかのぼって行われます。

OCSP は、検証局に特定の証明書のステータスを問い合わせ、チェックを検証局が扱う範囲に限定するため、よりスケーラブルな方法を提供します。

## サポート対象の CA サーバ

ASA は次の CA サーバをサポートしています。

Cisco IOS CS、ASA ローカル CA、およびサードパーティの X.509 準拠 CA ベンダー（次のベンダーが含まれますが、これらに限定はされません）。

- Baltimore Technologies
- Entrust
- Digicert
- Geotrust
- GoDaddy
- iPlanet/Netscape
- Microsoft Certificate Services
- RSA Keon
- Thawte
- VeriSign

## CRL

CRL は、有効期間内の証明書が発行元の CA によって無効にされているかどうかを ASA が判断するための 1 つの方法です。CRL コンフィギュレーションは、トラストポイントのコンフィギュレーションの一部です。

証明書を認証するときに必ず **revocation-check crl** コマンドを使用して CRL チェックを行うように、ASA を設定できます。また、**revocation-check crl none** コマンドを使用することで、CRL チェックをオプションにすることもできます。こうすると、更新された CRL データが CA から提供されない場合でも、証明書認証は成功します。

ASA は HTTP、SCEP、または LDAP を使用して、CA から CRL を取得できます。トラストポイントごとに取得された CRL は、トラストポイントごとに設定可能な時間だけキャッシュされます。

ASA により、CRL のキャッシュに設定されている時間を超過してキャッシュされた CRL がある場合、ASA ではその CRL は古すぎて信頼できない、つまり「失効した」と見なされます。次回の証明書認証で失効した CRL のチェックが必要な場合に、ASA によってより新しいバージョンの CRL の取得が試みられます。

ASA によって CRL がキャッシュされる時間は、次の 2 つの要素によって決まります。

- **cache-time** コマンドで指定される分数。デフォルト値は 60 分です。
- 取得した CRL 中の NextUpdate フィールド。このフィールドが CRL にない場合もあります。ASA が NextUpdate フィールドを必要とするかどうか、およびこのフィールドを使用するかどうかは、**enforcenextupdate** コマンドで制御します。

ASA では、これらの 2 つの要素が次のように使用されます。

- NextUpdate フィールドが不要の場合、**cache-time** コマンドで指定された時間が経過すると、ASA は CRL に失効のマークを付けます。
- NextUpdate フィールドが必要な場合、ASA は、**cache-time** コマンドと NextUpdate フィールドで指定されている 2 つの時間のうち短い方の時間で、CRL に失効のマークを付けます。たとえば、**cache-time** コマンドによってキャッシュ時間が 100 分に設定され、NextUpdate フィールドによって次のアップデートが 70 分後に指定されている場合、ASA は 70 分後に CRL に失効のマークを付けます。

ASA がメモリ不足で、特定のトラストポイント用にキャッシュされた CRL をすべて保存することができない場合、使用頻度が最も低い CRL が削除され、新しく取得した CRL 用の空き領域が確保されます。

## OCSP

OCSP は、有効期間内の証明書が発行元の CA によって無効にされているかどうかを ASA が判断するための 1 つの方法です。OCSP のコンフィギュレーションは、トラストポイントのコンフィギュレーションの一部です。

OCSP によって、証明書のステータスをチェックする範囲が検証局（OCSP サーバ、*応答側*とも呼ばれます）に限定され、ASA によって検証局に特定の証明書のステータスに関する問い合わせが行われます。これは、CRL チェックよりもスケーラブルで、最新の失効ステータスを確認できる方法です。この方法は、PKI の導入規模が大きい場合に便利で、安全なネットワークを拡大できます。

**（注）** ASA では、OCSP 応答に 5 秒間のスキューを許可します。

証明書を認証するときに必ず **revocation-check ocsp** コマンドを使用して OCSP チェックを行うように、ASA を設定できます。また、**revocation-check ocsp none** コマンドを使用することで、OCSP チェックをオプションにすることもできます。こうすると、更新された OCSP データが検証局から提供されない場合でも、証明書認証は成功します。

OCSP を利用すると、OCSP サーバの URL を 3 つの方法で定義できます。ASA は、これらのサーバを次の順に使用します。

1. **match certificate** コマンドの使用による証明書の照合の上書きルールで定義されている OCSP サーバの URL
2. **ocsp url** コマンドを使用して設定されている OCSP サーバの URL
3. クライアント証明書の AIA フィールド

**（注）** トラストポイントで OCSP の応答側の自己署名した証明書を検証するように設定するには、信頼できる CA 証明書として、この自己署名した応答側の証明書をそのトラストポイントにインポートします。次に、クライアント証明書を検証するトラストポイントで **match certificate** コマンドを設定して、応答側の証明書を検証するために、OCSP の応答側の自己署名された証明書を含むトラストポイントを使用するようにします。クライアント証明書の検証パスの外部にある応答側の証明書を検証する場合も、同じ手順で設定します。

通常、OCSP サーバ（応答側）の証明書によって、OCSP 応答が署名されます。ASA が応答を受け取ると、応答側の証明書を検証しようとします。通常、CA は、侵害される危険性を最小限に抑えるために、OCSP レスポンダ証明書のライフタイムを比較的短い期間に設定します。CA は一般に、応答側証明書に ocsp-no-check 拡張を含めて、この証明書では失効ステータス チェックが必要ないことを示します。ただし、この拡張がない場合、ASA はトラストポイントで指定されている方法で失効ステータスをチェックします。応答側の証明書を検証できない場合、失効ステータスをチェックできなくなります。この可能性を防ぐには、**revocation-check none** コマンドを使用して応答側の証明書を検証するトラストポイントを設定し、**revocation-check ocsp** コマンドを使用してクライアント証明書を設定します。

# ローカル CA

ローカル CA では、次のタスクが実行されます。

- ASA の基本的な認証局の動作を統合する。
- 証明書を導入する。
- 発行済み証明書のセキュアな失効チェックを実行する。
- ブラウザベースとクライアントベースの両方で SSL VPN 接続とともに使用するために、ASA 上に認証局を提供する。
- 外部の証明書認証に依存することなく、ユーザに信頼できるデジタル証明書を提供する。
- 証明書認証のためのセキュアな内部認証局を提供し、Web サイト ログインを使用した簡単なユーザ登録を実現する。

## ローカル CA ファイル用のストレージ

ASA では、ユーザ情報、発行済み証明書、および失効リストへのアクセスと実装にローカル CA データベースが使用されます。このデータベースは、デフォルトでローカル フラッシュ メモリに存在するか、または、マウントされて ASA にアクセス可能な外部のファイル システム上に設定することもできます。

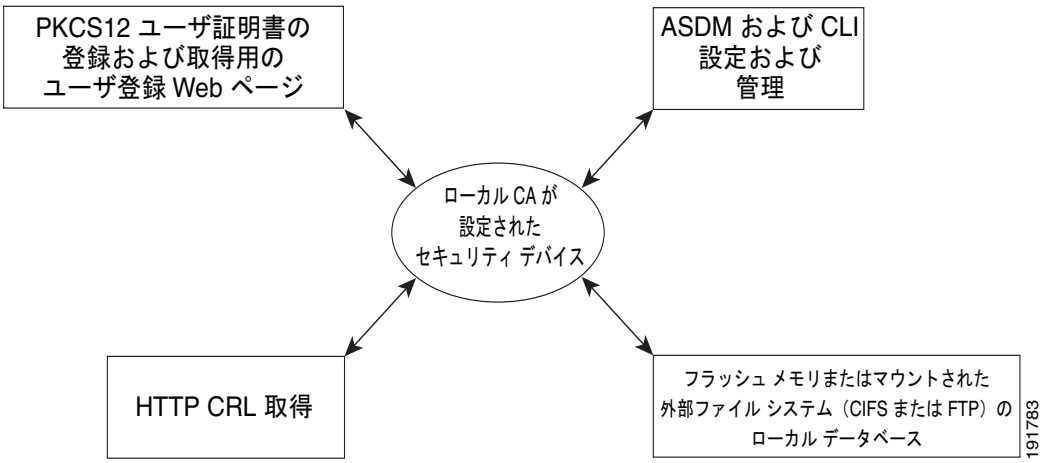
ローカル CA ユーザ データベースに保存できるユーザの数に制限はありませんが、フラッシュ メモリ ストレージに問題がある場合、管理者に対策を取るように警告する syslog が作成され、ローカル CA はストレージの問題が解決されるまでディセーブルになることがあります。フラッシュ メモリは、3500 人以下のユーザを持つデータベースを保存できますが、ユーザの数がそれを超えるデータベースでは外部ストレージが必要になります。

## ローカル CA サーバ

ASA にローカル CA サーバを設定すると、ユーザは、Web サイトにログインし、ユーザの登録資格を検証するためにローカル CA 管理者によって与えられたユーザ名とワンタイム パスワードを入力することで、証明書を登録できます。

図 35-1 に示すとおり、ローカル CA サーバは ASA に常駐し、Web サイト ユーザからの登録要求や、その他の証明書を検証するデバイスおよび ASA から発信された CRL の問い合わせを処理します。ローカル CA データベースおよびコンフィギュレーション ファイルは、ASA のフラッシュ メモリ（デフォルトのストレージ）または個別のストレージ デバイスに保持されます。

**図 35-1　　ローカル CA**




# 証明書とユーザ ログイン クレデンシャルの使用

この項では、認証と許可に証明書およびユーザ ログイン クレデンシャル（ユーザ名とパスワード）を使用する、さまざまな方法について説明します。これらの方式は、IPSec、AnyConnect、およびクライアントレス SSL VPN に適用されます。

すべての場合において、LDAP 許可では、パスワードをクレデンシャルとして使用しません。RADIUS 許可では、すべてのユーザの共通パスワードまたはユーザ名のいずれかを、パスワードとして使用します。

この項は、次の内容で構成されています。

## ユーザ ログイン クレデンシャルの使用

認証および許可のデフォルトの方法では、ユーザ ログイン クレデンシャルを使用します。

- 認証
  - トンネル グループ（ASDM 接続プロファイルとも呼ばれます）の認証サーバ グループ設定によりイネーブルにされます。
  - ユーザ名とパスワードをクレデンシャルとして使用します。
- 許可
  - トンネル グループ（ASDM 接続プロファイルとも呼ばれます）の許可サーバ グループ設定によりイネーブルにされます。
  - ユーザ名をクレデンシャルとして使用します。

## 証明書の使用

ユーザ デジタル証明書が設定されている場合、ASA によって最初に証明書が検証されます。ただし、証明書の DN は認証用のユーザ名として使用されません。

認証と許可の両方がイネーブルになっている場合、ASA によって、ユーザの認証と許可の両方にユーザ ログイン クレデンシャルが使用されます。

- 認証
  - 認証サーバ グループ設定によってイネーブルにされます。
  - ユーザ名とパスワードをクレデンシャルとして使用します。
- 許可
  - 許可サーバ グループ設定によってイネーブルにされます。
  - ユーザ名をクレデンシャルとして使用します。

認証がディセーブルで許可がイネーブルになっている場合、ASA によって許可にプライマリ DN のフィールドが使用されます。

- 認証
  - 認証サーバ グループ設定によってディセーブル（[None] に設定）になります。
  - クレデンシャルは使用されません。
- 許可
  - 許可サーバ グループ設定によってイネーブルにされます。
  - 証明書のプライマリ DN フィールドのユーザ名の値をクレデンシャルとして使用します。

（注）　証明書にプライマリ DN のフィールドが存在しない場合、ASA では、セカンダリ DN のフィールド値が許可要求のユーザ名として使用されます。

次のサブジェクト DN フィールドと値が含まれるユーザ証明書を例に挙げます。

```
Cn=anyuser,OU=sales;O=XYZCorporation;L=boston;S=mass;C=us;ea=anyuser@example.com
```

プライマリ DN = EA（電子メール アドレス）およびセカンダリ DN = CN（一般名）の場合、許可要求で使われるユーザ名は anyuser@example.com になります。

# デジタル証明書のライセンス要件

次の表に、この機能のライセンス要件を示します。

| モデル | ライセンス要件 |
|---|---|
| すべてのモデル | 基本ライセンス |

# ローカル証明書の前提条件

ローカル証明書には、次の前提条件があります。

- 証明書をサポートするように ASA が正しく設定されていることを確認します。ASA の設定に誤りがあると、登録に失敗したり、不正確な情報を含む証明書が要求されたりする可能性があります。
- ASA のホスト名とドメイン名が正しく設定されていることを確認します。現在設定されているホスト名とドメイン名を表示するには、**show running-config** コマンドを入力します。ホスト名とドメイン名を設定する方法の詳細については、「ホスト名、ドメイン名、およびパスワードの設定」（P.14-1）を参照してください。
- CA を設定する前に、ASA のクロックが正しく設定されていることを確認します。証明書には、有効になる日時と満了になる日時が指定されています。ASA が CA に登録して証明書を取得するとき、ASA は現在の時刻が証明書の有効期間の範囲内であるかどうかをチェックします。現在の時刻が有効期間の範囲外の場合、登録は失敗します。クロックを設定する方法の詳細については、「日付と時刻の設定」（P.14-3）を参照してください。

## SCEP プロキシ サポートの前提条件

サードパーティ製証明書の要求を送信するために ASA をプロキシとして設定するには、次の要件があります。

- AnyConnect セキュア モビリティ クライアント 3.0 以降がエンドポイントで実行中である必要があります。
- グループ ポリシーの接続プロファイルで設定される認証方式は、AAA 認証と証明書認証の両方を使用するように設定する必要があります。
- SSL ポートが、IKEv2 VPN 接続用に開いている必要があります。
- CA は、自動許可モードになっている必要があります。

# 注意事項と制限事項

この項では、この機能のガイドラインと制限事項について説明します。

### コンテキスト モードのガイドライン

- ローカル CA ではシングル コンテキスト モードとマルチ コンテキスト モードでサポートされています。
- サードパーティ CA ではシングル コンテキスト モードでのみサポートされています。

### ファイアウォール モードのガイドライン

ルーテッド ファイアウォール モードとトランスペアレント ファイアウォール モードでサポートされています。

### フェールオーバーのガイドライン

- ステートフル フェールオーバーではセッションの複製はサポートされません。
- Active/Active フェールオーバーおよび Active/Standby フェールオーバーはサポートされません。

### IPv6 のガイドライン

IPv6 をサポートします。

### その他のガイドライン

- ASA が CA サーバまたはクライアントとして設定されている場合、推奨される終了日（2038 年 1 月 19 日 03:14:08 UTC）を超えないよう、証明書の有効期を制限してください。このガイドラインは、サードパーティ ベンダーからインポートした証明書にも適用されます。
- フェールオーバーがイネーブルになっている場合、ローカル CA は設定できません。ローカル CA サーバを設定できるのは、フェールオーバーのないスタンドアロン ASA のみです。詳細については、「CSCty43366」を参照してください。
- 証明書の登録が完了すると、ASA により、ユーザのキー ペアと証明書チェーンを含む PKCS12 ファイルが保存されます。これには、登録ごとに約 2 KB のフラッシュ メモリまたはディスク領域が必要です。実際のディスク領域の量は、設定されている RSA キー サイズと証明書フィールドによって異なります。使用できるフラッシュ メモリの量が限られている ASA に、保留中の証明書登録を多数追加する場合には、このガイドラインに注意してください。これらの PKCS12 ファイルは、設定されている登録の取得タイムアウトの間、フラッシュ メモリに保存されます。
- **lifetime ca-certificate** コマンドは、ローカル CA サーバ証明書の初回生成時（初めてローカル CA サーバを設定し、**no shutdown** コマンドを発行するとき）に有効になります。CA 証明書の期限が切れると、設定されたライフタイム値を使用して新しい CA 証明書が生成されます。既存の CA 証明書のライフタイム値は変更できません。
- 管理インターフェイスに対する ASDM トラフィックおよび HTTPS トラフィックを保護するために、ID 証明書を使用するよう ASA を設定する必要があります。SCEP により自動的に生成される ID 証明書はリブートのたびに再生成されるため、必ず独自の ID 証明書を手動でインストールしてください。SSL のみに適用されるこの手順の例については、次の URL を参照してください。
http://www.cisco.com/en/US/products/ps6120/products_configuration_example09186a00809fcf91.shtml
- ASA および AnyConnect クライアントで検証できるのは、[X520Serialnumber] フィールド（[Subject Name] のシリアル番号）が PrintableString 形式である証明書のみです。シリアル番号の形式に UTF8 などのエンコーディングが使用されている場合、証明書認証は失敗します。

# デジタル証明書の設定

この項では、ローカル CA 証明書を設定する方法を説明します。このタイプのデジタル証明書を正しく設定するためには、必ず記載されている順にタスクを実行してください。この項では、次のトピックについて取り上げます。



## キー ペアの設定

キー ペアを生成するには、次の手順を実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto key generate rsa**<br><br>**例：**<br>`hostname/contexta(config)# crypto key generate rsa` | 1 つの汎用 RSA キー ペアを生成します。デフォルトのキー係数は 1024 です。その他の係数サイズを指定するには、**modulus** キーワードを使用します。<br>**（注）** 1024 ビットを超える RSA キー ペアを持つ ID 証明書を使用している複数の SSL 接続によって、ASA での CPU 使用率が高くなり、クライアントレス ログインが拒否される可能性があります。 |
| ステップ 2 | **crypto key generate rsa label** *key-pair-label*<br><br>**例：**<br>`hostname/contexta(config)# crypto key generate rsa label exchange` | （任意）ラベルを各キー ペアに割り当てます。このラベルは、キー ペアを使用するトラストポイントによって参照されます。ラベルを割り当てなかった場合、キー ペアには *Default-RSA-Key* というラベルが自動的に付けられます。 |

| | コマンド | 目的 |
|---|---|---|
| ステップ 3 | **show crypto key** *name of key*<br><br>**例：**<br>hostname/contexta(config)# show crypto key<br>examplekey | 生成したキー ペアを検証します。 |
| ステップ 4 | **write memory**<br><br>**例：**<br>hostname(config)# write memory | 生成したキー ペアを保存します。 |

# キー ペアの削除

キー ペアを削除するには、次の手順を実行します。

| コマンド | 目的 |
|---|---|
| **crypto key zeroize rsa**<br><br>**例：**<br>hostname(config)# crypto key zeroize rsa | キー ペアを削除します。 |

**例**

次に、キー ペアを削除する例を示します。

```
hostname(config)# crypto key zeroize rsa
WARNING: All RSA keys will be removed.
WARNING: All device certs issued using these keys will also be removed.

Do you really want to remove these keys?[yes/no] y
```

# トラストポイントの設定

トラストポイントを設定するには、次の手順を実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca trustpoint** *trustpoint-name*<br><br>**例：**<br>hostname/contexta(config)# crypto ca trustpoint Main | ASA が証明書を受け取る必要のある CA に対応するトラストポイントを作成します。crypto ca トラストポイント コンフィギュレーション モードに入り、ステップ 3 から設定できる CA 固有のトラストポイント パラメータを制御します。<br>**（注）** 接続しようとすると、トラストポイントからの ID 証明書の取得の試行時にそのトラストポイントに ID 証明書が含まれていないことを示す警告が表示されます。 |
| ステップ 2 | 次のいずれかのオプションを選択します。 | |

| | コマンド | 目的 |
|---|---|---|
| | **enrollment url** *url*<br><br>**例：**<br>hostname/contexta(config-ca-trustpoint)# enrollment url http://10.29.67.142:80/certsrv/mscep/mscep.dll | SCEP と指定のトラストポイントを使用して自動登録を要求し、登録用 URL を設定します。 |
| | **enrollment terminal**<br><br>**例：**<br>hostname/contexta(config-ca-trustpoint)# enrollment terminal | CA から取得した証明書を端末に貼り付けることによって、指定したトラストポイントで手動登録を要求します。 |
| ステップ 3 | **revocation-check crl none**<br>revocation-check crl<br>revocation-check none<br><br>**例：**<br>hostname/contexta(config-ca-trustpoint)# revocation-check crl none<br>hostname/contexta(config-ca-trustpoint)# revocation-check crl<br>hostname/contexta(config-ca-trustpoint)# revocation-check none | 使用可能な CRL コンフィギュレーション オプションを指定します。<br>**（注）** 必須または任意の CRL チェックをイネーブルにするには、証明書を取得してから、CRL 管理用のトラストポイントを設定します。 |
| ステップ 4 | **crl configure**<br><br>**例：**<br>hostname/contexta(config-ca-trustpoint)# crl configure | CRL コンフィギュレーション モードを開始します。 |
| ステップ 5 | **email** *address*<br><br>**例：**<br>hostname/contexta(config-ca-trustpoint)# email example.com | 登録時に、指定された電子メール アドレスを、証明書の Subject Alternative Name 拡張子に含めるように CA に要求します。 |
| ステップ 6 | **enrollment retry period**<br><br>**例：**<br>hostname/contexta(config-ca-trustpoint)# enrollment retry period 5 | （任意）再試行間隔を分単位で指定し、SCEP 登録だけに適用します。 |
| ステップ 7 | **enrollment retry count**<br><br>**例：**<br>hostname/contexta(config-ca-trustpoint)# enrollment retry period 2 | （任意）許可される再試行の最大数を指定し、SCEP 登録だけに適用します。 |
| ステップ 8 | **fqdn** *fqdn*<br><br>**例：**<br>hostname/contexta(config-ca-trustpoint)# fqdn example.com | 登録時に、指定された完全修飾ドメイン名を証明書の Subject Alternative Name 拡張子に含めるように CA に要求します。 |

| | コマンド | 目的 |
|---|---|---|
| ステップ 9 | **ip-address** *ip-address*<br><br>**例：**<br>`hostname/contexta(config-ca-trustpoint)# ip-address 10.10.100.1` | 登録時に、ASA の IP アドレスを証明書に含めるように CA に要求します。 |
| ステップ 10 | **keypair** *name*<br><br>**例：**<br>`hostname/contexta(config-ca-trustpoint)# keypair exchange` | 公開キーが認証の対象となるキー ペアを指定します。 |
| ステップ 11 | **match certificate map-name override ocsp**<br><br>**例：**<br>`hostname/contexta(config-ca-trustpoint)# match certificate examplemap override ocsp` | OCSP の URL の上書きと、OCSP の応答側の証明書の検証に使用するトラストポイントを設定します。 |
| ステップ 12 | **ocsp disable-nonce**<br><br>**例：**<br>`hostname/contexta(config-ca-trustpoint)# ocsp disable-nonce` | OCSP 要求の nonce 拡張をディセーブルにします。nonce 拡張は、リプレイ攻撃を防ぐために、要求と応答を暗号化してバインドします。 |
| ステップ 13 | **ocsp url**<br><br>**例：**<br>`hostname/contexta(config-ca-trustpoint)# ocsp url` | ASA で、トラストポイントに関連するすべての証明書をチェックするときに使用する OCSP サーバを設定します。クライアント証明書の AIA 拡張で指定されているサーバは使用しません。 |
| ステップ 14 | **password** *string*<br><br>**例：**<br>`hostname/contexta(config-ca-trustpoint)# password mypassword` | 登録時に CA に登録されるチャレンジ フレーズを指定します。CA は、通常、このフレーズを使用して、その後の失効要求を認証します。 |
| ステップ 15 | **revocation check**<br><br>**例：**<br>`hostname/contexta(config-ca-trustpoint)# revocation check` | 失効チェックの方法（CRL、OCSP、および none）を 1 つまたは複数設定します。 |
| ステップ 16 | **subject-name** *X.500 name*<br><br>**例：**<br>`hostname/contexta(config-ca-trustpoint)# myname X.500 examplename` | 登録時に、指定されたサブジェクト DN を証明書に含めるように CA に要求します。DN 文字列にカンマが含まれている場合は、値の文字列を二重引用符で囲みます（O="Company, Inc." など）。 |

| | コマンド | 目的 |
|---|---|---|
| ステップ 17 | **serial-number**<br><br>**例：**<br>`hostname/contexta(config-ca-trustpoint)# serial number JMX1213L2A7` | 登録時に、ASA のシリアル番号を証明書に含めるように CA に要求します。 |
| ステップ 18 | **write memory**<br><br>**例：**<br>`hostname/contexta(config)# write memory` | 実行コンフィギュレーションを保存します。 |

# トラストポイントの CRL の設定

証明書の認証時に必須またはオプションの CRL チェックを行うには、トラストポイントごとに CRL を設定する必要があります。トラストポイントの CRL を設定するには、次の手順を実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca trustpoint** *trustpoint-name*<br><br>**例：**<br>`hostname (config)# crypto ca trustpoint Main` | CRL コンフィギュレーションを変更するトラストポイントに対して、crypto ca trustpoint コンフィギュレーション モードに入ります。<br>**（注）** このコマンドを入力する前に、CRL がイネーブルであることを確認してください。また、認証が成功するためには、CRL が使用可能である必要があります。 |
| ステップ 2 | **crl configure**<br><br>**例：**<br>`hostname (config-ca-trustpoint)# crl configure` | 現在のトラストポイントで、crl コンフィギュレーション モードを開始します。<br>**ヒント** すべての CRL コンフィギュレーションのパラメータをデフォルト値に設定するには、**default** コマンドを使用します。CRL の設定中は、いつでもこのコマンドを入力して手順をやり直すことができます。 |
| ステップ 3 | 次のいずれかを実行します。 | |
| | **policy cdp**<br><br>**例：**<br>`hostname (config-ca-crl)# policy cdp` | 取得ポリシーを設定します。CRL は、認証済みの証明書で指定されている CRL 分散ポイントだけから取得できます。<br>**（注）** SCEP の取得は、証明書で指定されている分散ポイントではサポートされていません。<br><br>続行するには、ステップ 5 に進みます。 |
| | **policy static**<br><br>**例：**<br>`hostname (config-ca-crl)# policy static` | 取得ポリシーを設定します。CRL は、設定した URL だけから取得できます。<br>続行するには、ステップ 4 に進みます。 |

| | コマンド | 目的 |
|---|---|---|
| | **policy both**<br><br>**例：**<br>hostname (config-ca-crl)# policy both | 取得ポリシーを設定します。CRL は、認証済みの証明書で指定されている CRL 分散ポイントと、設定した URL の両方から取得できます。<br>続行するには、ステップ 4 に進みます。 |
| ステップ 4 | **url** *n url*<br><br>**例：**<br>hostname (config-ca-crl)# url 2<br>http://www.example.com | CRL ポリシーの設定時に **static** または **both** キーワードを使用する場合、CRL 取得用の URL を設定する必要があります。1 ～ 5 のランクを付けて、最大 5 つの URL を入力できます。*n* は、URL に割り当てるランクです。URL を削除するには、**no url** *n* コマンドを使用します。 |
| ステップ 5 | **protocol http \| ldap \| scep**<br><br>**例：**<br>hostname (config-ca-crl)# protocol http | 取得方法を設定します。CRL 取得方式として HTTP、LDAP、または SCEP を指定します。 |
| ステップ 6 | **cache-time** *refresh-time*<br><br>**例：**<br>hostname (config-ca-crl)# cache-time 420 | ASA が現在のトラストポイントの CRL をキャッシュしている時間を設定します。*refresh-time* は、CRL を失効と判断するまで ASA が待機する時間（分）です。 |
| ステップ 7 | 次のいずれかを実行します。 | |
| | **enforcenextupdate**<br><br>**例：**<br>hostname (config-ca-crl)# enforcenextupdate | CRL の NextUpdate フィールドを要求します。これがデフォルト設定です。 |
| | **no enforcenextupdate**<br><br>**例：**<br>hostname (config-ca-crl)# no enforcenextupdate | CRL に NextUpdate フィールドが存在しないことを許可します。 |
| ステップ 8 | **ldap-defaults** *server*<br><br>**例：**<br>hostname (config-ca-crl)# ldap-defaults ldap1 | LDAP が取得プロトコルとして指定されている場合に ASA に LDAP サーバを指定します。LDAP サーバは、DNS ホスト名または IP アドレスで指定できます。LDAP サーバがデフォルトの 389 以外のポートで LDAP クエリーを受信する場合は、ポート番号も指定できます。<br>**（注）** LDAP サーバを指定するために、IP アドレスの代わりにホスト名を使用する場合は、ASA が DNS を使用するように設定されていることを確認します。 |
| ステップ 9 | **ldap-dn** *admin-DN password*<br><br>**例：**<br>hostname (config-ca-crl)# ldap-dn<br>cn=admin,ou=devtest,o=engineering c00lRunZ | LDAP サーバでクレデンシャルを必要としている場合に、CRL の取得を許可します。 |

| | コマンド | 目的 |
|---|---|---|
| ステップ 10 | **crypto ca crl request** *trustpoint*<br><br>**例：**<br>hostname (config-ca-crl)# crypto ca crl request Main | 指定したトラストポイントによって示される CA から現在の CRL を取得し、現在のトラストポイントの CRL コンフィギュレーションをテストします。 |
| ステップ 11 | **write memory**<br><br>**例：**<br>hostname (config)# write memory | 実行コンフィギュレーションを保存します。 |

# トラストポイント コンフィギュレーションのエクスポート

トラスト ポイント設定をエクスポートするには、次のコマンドを入力します。

| コマンド | 目的 |
|---|---|
| **crypto ca export** *trustpoint*<br><br>**例：**<br>hostname(config)# crypto ca export Main | トラスト ポイント設定を関連するすべてのキーと PKCS12 形式の証明書とともにエクスポートします。ASA は PKCS12 データを端末に表示します。この表示されたデータはコピーできます。トラストポイント データはパスワードで保護されますが、このデータをファイルに保存する場合は、そのファイルがセキュアな場所にあることを確認してください。 |

**例**

次の例では、トラストポイント Main の PKCS12 データをパスフレーズ Wh0zits とともにエクスポートしています。

```
hostname (config)# crypto ca export Main pkcs12 Wh0zits

Exported pkcs12 follows:

[ PKCS12 data omitted ]

---End - This line not part of the pkcs12---
```

# トラストポイント コンフィギュレーションのインポート

トラスト ポイント設定をインポートするには、次のコマンドを入力します。

| コマンド | 目的 |
|---|---|
| **crypto ca import** *trustpoint* **pkcs12**<br><br>**例：**<br>hostname(config)# crypto ca import Main pkcs12 | キー ペアと、トラスト ポイント設定に関連付けられている発行済み証明書をインポートします。ASA では、Base-64 形式で端末にテキストを貼り付けるように要求されます。トラストポイントとともにインポートされるキー ペアには、作成するトラストポイントの名前と一致するラベルが割り当てられます。<br><br>**（注）** ASA に同じ CA を共有するトラストポイントが複数ある場合、CA を共有するトラストポイントでユーザ証明書の検証に使用できるのは 1 つだけです。CA を共有するどのトラストポイントを使用して、その CA が発行したユーザ証明書を検証するかを制御するには、**support-user-cert-validation** キーワードを使用します。 |

**例**

次の例では、パスフレーズ Wh0zits とともに PKCS12 データを手動でトラストポイント Main にインポートしています。

```
hostname (config)# crypto ca import Main pkcs12 Wh0zits

Enter the base 64 encoded pkcs12.
End with a blank line or the word "quit" on a line by itself:
[ PKCS12 data omitted ]
quit
INFO: Import PKCS12 operation completed successfully
```

次に、トラストポイント Main の証明書を手動でインポートする例を示します。

```
hostname (config)# crypto ca import Main certificate
% The fully-qualified domain name in the certificate will be:
securityappliance.example.com

Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself
[ certificate data omitted ]
quit
INFO: Certificate successfully imported
```

# CA 証明書マップ規則の設定

証明書の [Issuer] フィールドと [Subject] フィールドに基づいて、ルールを設定できます。作成した規則を使用すると、**tunnel-group-map** コマンドによって、IPSec ピアの証明書をトンネル グループにマッピングできます。ASA は CA 証明書マップを 1 つサポートしています。これには、複数の規則を設定できます。

CA 証明書マップ規則を設定するには、次の手順を実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca certificate map** *sequence-number*<br><br>**例：**<br>`hostname(config)# crypto ca certificate map 1` | 設定するルールの CA 証明書マップ コンフィギュレーション モードを開始し、ルール インデックス番号を指定します。 |
| ステップ 2 | **issuer-name** *DN-string*<br><br>**例：**<br>`hostname(config-ca-cert-map)# issuer-name cn=asa.example.com` | すべての発行済み証明書の認定者名を指定します。これは自己署名 CA 証明書のサブジェクト名 DN でもあります。属性と値のペアを区切るには、カンマを使用します。カンマを含む値は、引用符で囲んでください。発行者名は、英数字で 500 文字未満にする必要があります。デフォルトの発行者名は cn=*hostame.domain-name* です。 |
| ステップ 3 | **subject-name attr** *tag* **eq \| co \| ne \| nc** *string*<br><br>**例：**<br>`hostname(config-ca-cert-map)# subject-name attr cn eq mycert` | ASA が証明書の [Issuer] フィールドまたは [Subject] フィールドの値に適用できるテストを指定します。テストは、特定の属性または全体のフィールドに適用できます。ルールごとに複数のテストを設定できますが、これらのコマンドで指定するすべてのテストは、証明書と一致するルールで真である必要があります。有効な演算子は次のとおりです。<br>• eq：フィールドまたは属性が所定の値と一致する。<br>• ne：フィールドまたは属性が所定の値と一致しない。<br>• co：フィールドまたは属性の一部または全部が所定の値と一致する。<br>• nc：フィールドまたは属性の全部が所定の値と一致しない。 |
| ステップ 4 | **write memory**<br><br>**例：**<br>`hostname (config)# write memory` | 実行コンフィギュレーションを保存します。 |

# 手動での証明書の取得

証明書を手動で取得するには、次の手順を実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca authenticate** *trustpoint*<br><br>**例：**<br>hostname(config)# crypto ca authenticate Main<br>Enter the base 64 encoded CA certificate.<br>End with a blank line or the word "quit" on a line by itself<br>MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB<br>*[ certificate data omitted ]*<br>/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==<br>**quit**<br><br>INFO: Certificate has the following attributes:<br>Fingerprint:  24b81433 409b3fd5 e5431699 8d490d34<br>Do you accept this certificate? [yes/no]: **y**<br>Trustpoint CA certificate accepted.<br><br>% Certificate successfully imported | 設定したトラストポイントの CA 証明書をインポートします。<br><br>**（注）** この手順は、トラストポイントで示されている CA から、base-64 encoded CA 証明書を取得済みであることを前提としています。<br><br>トラストポイントが手動で証明書を取得する必要があるかどうかは、そのトラストポイントを設定するときに **enrollment terminal** コマンドを使用するかどうかによって決まります。詳細については、「トラストポイントの設定」（P.35-12）を参照してください。 |
| ステップ 2 | **crypto ca enroll** *trustpoint*<br><br>**例：**<br>hostname(config)# crypto ca enroll Main<br>% Start certificate enrollment ..<br><br>% The fully-qualified domain name in the certificate will be: securityappliance.example.com<br><br>% Include the device serial number in the subject name? [yes/no]: **n**<br><br>Display Certificate Request to terminal? [yes/no]: **y**<br>Certificate Request follows:<br><br>MIIBoDCCAQkCAQAwIzEhMB8GCSqGSIb3DQEJAhYSRmVyYWxQaXguY2lzY28uY29t<br>*[ certificate request data omitted ]*<br>jF4waw68eOxQxVmdgMWeQ+RbIOYmvt8g6hnBTrd0GdqjjVLt<br><br>---End - This line not part of the certificate request---<br><br>Redisplay enrollment request? [yes/no]: **n** | このトラストポイントを持つASA を登録します。署名データの証明書を生成し、設定したキーのタイプによっては暗号化データの証明書も生成します。<br><br>署名と暗号化に別々の RSA キーを使用する場合、**crypto ca enroll** コマンドは各キーに対する 2 つの証明書要求を表示します。署名と暗号化の両方に汎用の RSA キーを使用する場合、**crypto ca enroll** コマンドでは証明書要求が 1 つ表示されます。<br><br>登録を完了するには、該当するトラストポイントで示される CA から **crypto ca enroll** コマンドで生成されたすべての証明書要求に対する証明書を取得します。証明書が base-64 形式であることを確認してください。 |

| | コマンド | 目的 |
|---|---|---|
| ステップ 3 | **crypto ca import** *trustpoint* **certificate**<br><br>**例：**<br>hostname (config)# crypto ca import Main certificate<br>% The fully-qualified domain name in the certificate will be: securityappliance.example.com<br><br>Enter the base 64 encoded certificate.<br>End with a blank line or the word "quit" on a line by itself<br>*[ certificate data omitted ]*<br>quit<br>INFO: Certificate successfully imported | CA から受信する各証明書をインポートします。証明書を base-64 形式で端末に貼り付けることが求められます。 |
| ステップ 4 | **show crypto ca server certificate**<br><br>**例：**<br>hostname(config)# show crypto ca server certificate Main | ASA に発行された証明書の詳細とトラストポイントの CA 証明書を表示して、登録プロセスが成功したことを確認します。 |
| ステップ 5 | **write memory**<br><br>**例：**<br>hostname(config)# write memory | 実行コンフィギュレーションを保存します。<br><br>手動登録を設定したトラストポイントごとに、これらの手順を繰り返します。 |

# SCEP を使用した証明書の自動取得

SCEP を使用して証明書を自動的に取得するには、次の手順を実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca authenticate** *trustpoint*<br><br>**例：**<br>hostname/contexta(config)# crypto ca authenticate Main | 設定したトラストポイントの CA 証明書を取得します。<br><br>**（注）** この手順は、トラストポイントで示されている CA から、base-64 encoded CA 証明書を取得済みであることを前提としています。<br><br>トラスト ポイントを設定するときに、**enrollment url** コマンドを使用すると、SCEP を使用して証明書を自動的に取得する必要があるかどうかを判断できます。詳細については、「トラストポイントの設定」（P.35-12）を参照してください。 |
| ステップ 2 | **crypto ca enroll** *trustpoint*<br><br>**例：**<br>hostname/contexta(config)# crypto ca enroll Main | このトラストポイントを持つASA を登録します。署名データの証明書を取得し、設定したキーのタイプによっては暗号化データの証明書も取得します。CA の管理者は、CA が証明書を付与する前に手動で登録要求を認証しなければならない場合があるため、このコマンドを入力する前に CA の管理者に連絡してください。<br><br>ASA が証明書要求を送信してから 1 分（デフォルト）以内に CA から証明書を受け取らなかった場合は、証明書要求が再送信されます。ASA によって、証明書を受信するまで 1 分ごとに証明書要求が送信されます。<br><br>トラストポイントの完全修飾ドメイン名が ASA の完全修飾ドメイン名と一致しなかった場合（完全修飾ドメイン名が文字の場合も含む）、警告が表示されます。この問題を解決するには、登録プロセスを終了し、必要な修正を行ってから、**crypto ca enroll** コマンドを再入力します。<br><br>**（注）** **crypto ca enroll** コマンドを発行した後、証明書を受信する前に ASA がリブートされた場合は、**crypto ca enroll** コマンドを再入力して、CA 管理者に連絡してください。 |
| ステップ 3 | **show crypto ca server certificate**<br><br>**例：**<br>hostname/contexta(config)# show crypto ca server certificate Main | ASA に発行された証明書の詳細とトラストポイントの CA 証明書を表示して、登録プロセスが成功したことを確認します。 |
| ステップ 4 | **write memory**<br><br>**例：**<br>hostname/contexta(config)# write memory | 実行コンフィギュレーションを保存します。 |

# SCEP 要求のプロキシ サポートの設定

サードパーティの CA を使用してリモート アクセスのエンド ポイントを認証するように ASA を設定するには、次の手順を実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ikev2 enable outside client-services port** *portnumber*<br><br>**例：**<br>`hostname(config-tunnel-ipsec)# crypto ikev2 enable outside client-services` | クライアント サービスをイネーブルにします。<br>**（注）**　IKEv2 をサポートする場合にのみ必要です。<br>このコマンドをトンネル グループ ipsec 属性コンフィギュレーション モードで入力します。<br>デフォルトのポート番号は 443 です。 |
| ステップ 2 | **scep-enrollment enable**<br><br>**例：**<br>`hostname(config-tunnel-general)# scep-enrollment enable`<br>`INFO: 'authentication aaa certificate' must be configured to complete setup of this option.` | トンネル グループの SCEP 登録をイネーブルにします。<br>このコマンドを tunnel-group general-attributes コンフィギュレーション モードで入力します。 |
| ステップ 3 | **scep-forwarding-url value** *URL*<br><br>**例：**<br>`hostname(config-group-policy)# scep-forwarding-url value http://ca.example.com:80/` | グループ ポリシー用の SCEP CA を登録します。<br>このコマンドは、サードパーティのデジタル証明書をサポートするグループ ポリシーごとに 1 回入力します。また、このコマンドはグループ ポリシー一般属性コンフィギュレーション モードで入力します。<br>*URL* は CA の SCEP URL です。 |
| ステップ 4 | **secondary-pre-fill-username clientless hide use-common-password** *password*<br><br>**例：**<br>`hostname(config)# tunnel-group remotegrp webvpn-attributes`<br>`hostname(config-tunnel-webvpn)# secondary-pre-fill-username clientless hide use-common-password secret` | 証明書が SCEP プロキシの WebLaunch のサポートに使用できない場合は、共通のセカンダリ パスワードを使用します。<br>SCEP プロキシをサポートするには、**hide** キーワードを使用する必要があります。<br>たとえば、証明書は、それを要求するエンド ポイントでは使用できません。エンド ポイントに証明書が存在する場合、AnyConnect は ASA への接続を切断し、その後再接続して、内部ネットワーク リソースへのアクセスを提供する DAP ポリシーに適合するようにします。 |

| | コマンド | 目的 |
|---|---|---|
| ステップ 5 | **secondary-pre-fill-username ssl-client hide use-common-password** *password*<br><br>**例：**<br>hostname(config-tunnel-webvpn)# secondary-pre-fill-username ssl-client hide use-common-password secret | AnyConnect VPN セッションの事前入力されているセカンダリ ユーザ名を非表示にします。<br><br>以前のリリースから継承した **ssl-client** キーワードに関係なく、IKEv2 または SSL を使用する AnyConnect セッションをサポートするには、このコマンドを使用します。<br><br>SCEP プロキシをサポートするには、**hide** キーワードを使用する必要があります。 |
| ステップ 6 | **secondary-username-from-certificate** {**use-entire-name** \| **use-script** \| {*primary_attr* [*secondary-attr*]}} [**no-certificate-fallback cisco-secure-desktop machine-unique-id**]<br><br>**例：**<br>hostname(config-tunnel-webvpn)# secondary-username-from-certificate CN no-certificate-fallback cisco-secure-desktop machine-unique-id | 証明書が使用できないときにはユーザ名を指定します。 |

# ローカル CA サーバのイネーブル化

ローカル CA サーバをイネーブルにする前に、7 文字以上からなるパスフレーズを作成して、生成されるローカル CA 証明書とキー ペアを含む PKCS12 ファイルを符号化し、アーカイブしておく必要があります。CA 証明書またはキー ペアが失われた場合は、パスフレーズを使用して PKCS12 アーカイブをロック解除します。

ローカル CA サーバをイネーブルにするには、次のコマンドを実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>hostname (config)# crypto ca server | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **no shutdown**<br><br>**例：**<br>hostname (config-ca-server)# no shutdown | ローカル CA サーバをイネーブルにします。ローカル CA サーバの証明書、キー ペア、および必要なデータベース ファイルを生成し、ローカル CA サーバの証明書とキー ペアを PKCS12 ファイル内のストレージにアーカイブします。8 ～ 65 文字の英数字のパスワードが必要になります。初期スタートアップ後、パスフレーズを求めるプロンプトを表示せずにローカル CA をディセーブルにすることができます。<br><br>**（注）** ローカル CA サーバをイネーブルにしたら、コンフィギュレーションを保存して、リブート後にローカル CA 証明書とキー ペアが失われないようにします。 |

### 例

次の例では、ローカル CA サーバをイネーブルにします。

```
hostname (config)# crypto ca server
hostname (config-ca-server)# no shutdown

% Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key
% or type Return to exit

Password: caserver

Re-enter password: caserver

Keypair generation process begin.Please wait...
```

次に、ローカル CA サーバのコンフィギュレーションとステータスを表示するサンプル出力を示します。

```
Certificate Server LOCAL-CA-SERVER:
    Status: enabled
    State: enabled
    Server's configuration is locked (enter "shutdown" to unlock it)
    Issuer name: CN=wz5520-1-16
    CA certificate fingerprint/thumbprint: (MD5)
        76dd1439 ac94fdbc 74a0a89f cb815acc
    CA certificate fingerprint/thumbprint: (SHA1)
        58754ffd 9f19f9fd b13b4b02 15b3e4be b70b5a83
    Last certificate issued serial number: 0x6
    CA certificate expiration timer: 14:25:11 UTC Jan 16 2008
    CRL NextUpdate timer: 16:09:55 UTC Jan 24 2007
    Current primary storage dir: flash:
```

## ローカル CA サーバの設定

ローカル CA サーバを設定するには、次のコマンドを実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>`hostname (config)# crypto ca server` | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA を作成します。 |
| ステップ 2 | **smtp from-address** *e-mail_address*<br><br>**例：**<br>`hostname (config-ca-server) # smtp from-address SecurityAdmin@example.com` | SMTP from-address を指定します。これはローカル CA がユーザに登録案内用の OTP を送る電子メール メッセージを送信するときに、発信元アドレスとして使用する有効な電子メール アドレスを指定です。 |

| | コマンド | 目的 |
|---|---|---|
| ステップ 3 | **subject-name-default** *dn*<br><br>**例：**<br>`hostname (config-ca-server)# subject-name-default cn=engineer, o=asc systems, c="US"` | （任意）発行された証明書のユーザ名に付加する subject-name DN を指定します。<br><br>subject-name DN とユーザ名は結合して、ローカル CA サーバによって発行されたすべてのユーザ証明書の DN を形成します。subject-name DN を指定しない場合、ユーザ データベースにユーザを追加するたびに、ユーザ証明書に含めるサブジェクト名 DN を正確に指定する必要があります。<br><br>**（注）** ローカル CA をイネーブルにした後は、issuer-name 値および keysize server 値は変更できないため、設定したローカル CA をイネーブルにする前に、オプションのすべてのパラメータを慎重に見直してください。 |
| ステップ 4 | **no shutdown**<br><br>**例：**<br>`hostname (config-ca-server)# no shutdown` | 自己署名した証明書を作成し、ASA のローカル CA に関連付けます。自己署名した証明書のキーの使用拡張には、キー暗号化、キー シグニチャ、CRL 署名、および証明書署名機能があります。<br><br>**（注）** 自己署名したローカル CA 証明書が生成された後、特性を変更するには、既存のローカル CA サーバを削除して、完全に作成し直す必要があります。<br><br>ローカル CA サーバはユーザ証明書を把握しているため、管理者は、必要に応じて特権を無効にしたり元に戻したりできます。 |

## 例

次の例は、必要なパラメータすべてで事前定義済みのデフォルト値を使用してローカル CA サーバを設定する方法を示しています。

```
hostname (config)# crypto ca server
hostname (config-ca-server) # smtp from-address SecurityAdmin@example.com
hostname (config-ca-server)# subject-name-default cn=engineer, o=asc Systems, c=US
hostname (config-ca-server)# no shutdown
```

# ローカル CA サーバのカスタマイズ

カスタマイズされたローカル CA グループ サーバを設定するには、次のコマンドを実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>`hostname (config)# crypto ca server` | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **issuer-name** *DN-string*<br><br>**例：**<br>`hostname (config-ca-server)# issuer-name cn=xx5520,cn=30.132.0.25,ou=DevTest,ou=QA,o=ASC Systems` | デフォルト値のないパラメータを指定します。 |
| ステップ 3 | **smtp subject** *subject-line*<br><br>**例：**<br>`hostname (config-ca-server) # smtp subject Priority E-Mail: Enclosed Confidential Information is Required for Enrollment` | ローカル CA サーバから送信されるすべての電子メールの件名フィールドに表示するテキストをカスタマイズします。 |
| ステップ 4 | **smtp from-address** *e-mail_address*<br><br>**例：**<br>`hostname (config-ca-server) # smtp from-address SecurityAdmin@example.com` | ローカル CA サーバによって生成されるすべての電子メールの [From:] フィールドに使用する電子メール アドレスを指定します。 |
| ステップ 5 | **subject-name-default** *dn*<br><br>**例：**<br>`hostname (config-ca-server) # subject-name default cn=engineer, o=ASC Systems, c=US` | 発行された証明書のユーザ名に追加するオプションの subject-name DN を指定します。デフォルトの subject-name DN は、ローカル CA サーバによって発行されたすべてのユーザ証明書でユーザ名の一部になります。<br><br>許可される DN 属性キーワードは次のとおりです。<br>• C = 国<br>• CN = 通常名<br>• EA = 電子メール アドレス<br>• L = 地名<br>• O = 組織名<br>• OU = 組織ユニット<br>• ST = 州 / 都道府県<br>• SN = 姓名の姓<br>• ST = 州 / 都道府県<br><br>**（注）** subject-name-default を標準の subject-name のデフォルト値として機能するように指定しない場合、ユーザを追加するたびに DN を指定する必要があります。 |

# ローカル CA サーバのデバッグ

新たに設定されたローカル CA サーバをデバッグするには、次のコマンドを実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>`hostname (config)# crypto ca server` | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **debug crypto ca server**<br><br>**例：**<br>`hostname (config-ca-server)# debug crypto ca server` | ローカル CA サーバを設定およびイネーブルにするときに、デバッグ メッセージを表示します。レベル 1 のデバッグ機能を実行します。レベル 1 ～ 255 を使用できます。<br>**（注）** デバッグ コマンドによって、ビジー状態のネットワークのトラフィックが低速化することがあります。レベル 5 以上は、未加工データのダンプのために予約されており、出力が多くなりすぎるため、通常のデバッグ時は避ける必要があります。 |

# ローカル CA サーバのディセーブル化

ローカル CA サーバをディセーブルにするには、次のコマンドを実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>`hostname (config)# crypto ca server` | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **shutdown**<br><br>**例：**<br>`hostname (config-ca-server)# shutdown`<br>`INFO: Local CA Server has been shutdown.` | ローカル CA サーバをディセーブルにします。Web サイト登録をディセーブルにして、ローカル CA サーバ コンフィギュレーションの修正を可能にします。現在のコンフィギュレーションと関連付けられたファイルを保存します。初期スタートアップ後、パスフレーズを求めるプロンプトを表示せずにローカル CA を再びイネーブルにすることができます。 |

# ローカル CA サーバの削除

既存のローカル CA サーバ（イネーブル状態またはディセーブル状態）を削除するには、次のいずれかのコマンドを入力します。

| コマンド | 目的 |
|---|---|
| 次のいずれかを実行します。 | |
| **no crypto ca server**<br><br>**例：**<br>`hostname (config)# no crypto ca server` | 既存のローカル CA サーバ（イネーブル状態またはディセーブル状態）を削除します。<br><br>**（注）** ローカル CA サーバを削除すると、ASA からコンフィギュレーションが削除されます。削除されたコンフィギュレーションは元に戻せません。 |
| **clear configure crypto ca server**<br><br>**例：**<br>`hostname (config)# clear config crypto ca server` | 関連付けられたローカル CA サーバのデータベースとコンフィギュレーション ファイル（つまり、ワイルドカード名が LOCAL-CA-SERVER.* のすべてのファイル）も必ず削除してください。 |

# ローカル CA 証明書の特性の設定

次のローカル CA 証明書の特性を設定できます。

- すべてのユーザ証明書に表示されるとおりの証明書発行元の名前。
- ローカル CA 証明書（サーバおよびユーザ）および CRL のライフタイム。
- ローカル CA とユーザ証明書に関連付けられている公開キーペアおよび秘密キーペアの長さ。

この項では、次のトピックについて取り上げます。

- 「ローカル CA サーバ証明書およびキー ペアのアーカイブ」（P.35-43）

## 発行元名の設定

証明書の発行元名を設定するには、次のコマンドを実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>hostname (config)# crypto ca server | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **issuer-name** *DN-string*<br><br>**例：**<br>hostname (config-ca-server)# issuer-name CN=xx5520,CN=30.132.0.25,ou=DevTest,ou=QA,O=ABC Systems | ローカル CA 証明書のサブジェクト名を指定します。設定した証明書発行元名は、自己署名したローカル CA 証明書のサブジェクト名および発行元名の両方であり、発行済みクライアント証明書すべておよび発行済み CRL の発行元名でもあります。ローカル CA のデフォルトの発行元名の形式は、*hostname.domainname* です。<br>**（注）** ローカル CA が最初にイネーブルになった後に発行元名の値を変更することはできません。 |

## CA 証明書のライフタイムの設定

ローカル CA サーバ証明書のライフタイムを設定するには、次のコマンドを実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>hostname (config)# crypto ca server | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |

| | コマンド | 目的 |
|---|---|---|
| ステップ 2 | **lifetime ca-certificate** *time*<br><br>**例：**<br>`hostname (config-ca-server)# lifetime ca-certificate 365` | 証明書に含まれる有効期限を指定します。ローカル CA 証明書のデフォルトのライフタイムは 3 年間です。<br><br>推奨される終了日（2038 年 1 月 19 日 03:14:08 UTC）を超えないよう、証明書の有効期間を制限します。 |
| ステップ 3 | **no lifetime ca-certificate**<br><br>**例：**<br>`hostname (config-ca-server)# no lifetime ca-certificate` | （任意）ローカル CA 証明書のライフタイムをデフォルトの 3 年にリセットします。<br><br>ローカル CA サーバでは、CA 証明書の期限が切れる 30 日前に後継の CA 証明書が自動的に生成されます。この証明書をエクスポートし、他のデバイスにインポートすることにより、ローカル CA 証明書が発行したユーザ証明書のローカル CA 証明書検証を、期限が切れた後に行うことができます。次のような preexpiration syslog メッセージが生成されます。<br><br>`%ASA-1-717049: Local CA Server certificate is due to expire in days days and a replacement certificate is available for export.`<br><br>**（注）** この自動ロールオーバーが通知されたら、管理者は、新しいローカル CA 証明書が有効期限の前に必要なすべてのデバイスにインポートされるようにする必要があります。 |

## ユーザ証明書のライフタイムの設定

ユーザ証明書のライフタイムを設定するには、次のコマンドを実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>`hostname (config)# crypto ca server` | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **lifetime certificate** *time*<br><br>**例：**<br>`hostname (config-ca-server)# lifetime certificate 60` | ユーザ証明書の有効期間の時間の長さを設定します。<br><br>**（注）** ユーザ証明書の期限が満了になる前に、ローカル CA サーバは、証明書の有効期限の数日前にそのユーザに登録特権を付与し、更新の注意を設定し、証明書更新用の登録ユーザ名および OTP を電子メールで配信することで、証明書の更新プロセスを自動的に開始します。推奨される終了日（2038 年 1 月 19 日 03:14:08 UTC）を超えないよう、証明書の有効期間を制限します。 |

## CRL のライフタイムの設定

CRL ライフタイムを設定するには、次のコマンドを実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>`hostname (config)# crypto ca server` | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **lifetime crl** *time*<br><br>**例：**<br>`hostname (config-ca-server)# lifetime crl 10` | CRL の有効期間の時間の長さを設定します。<br>ローカル CA では、ユーザ証明書が失効または失効解除されるたびに CRL をアップデートおよび再発行しますが、失効状態に変更がない場合、CRL の再発行は、そのライフタイムの中で 1 回だけ自動的に行われます。CRL のライフタイムを指定しない場合、デフォルトの期間は 6 時間になります。 |
| ステップ 3 | **crypto ca server crl issue**<br><br>**例：**<br>`hostname(config)# crypto ca server crl issue`<br>`A new CRL has been issued.` | CRL を任意のタイミングで強制的に発行します。現在の CRL がただちに更新および再生成され、既存の CRL が上書きされます。<br>**（注）** CRL ファイルがエラーで削除されたり、壊れたりして、再生成が必要になった場合以外は、このコマンドを使用しないでください。 |

## サーバのキーサイズの設定

サーバのキーサイズを設定するには、次のコマンドを実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>`hostname (config)# crypto ca server` | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **keysize server**<br><br>**例：**<br>`hostname (config-ca-server)# keysize server 2048` | ユーザ証明書登録で生成される公開キーと秘密キーのサイズを指定します。キーペア サイズのオプションは 512、768、1024、2048 ビットで、デフォルト値は 1024 ビットです。<br>**（注）** ローカル CA をイネーブルにした後でローカル CA のキーサイズを変更することはできません。発行済み証明書すべてが無効になるためです。ローカル CA キーサイズを変更するには、現在のローカル CA を削除して新しいローカル CA を再設定する必要があります。 |

### 例

次は、データベースの 2 つのユーザ証明書の出力例です。

```
Username: user1
```

```
Renewal allowed until: Not Allowed
Number of times user notified: 0
PKCS12 file stored until: 12:45:52 UTC Fri Jan 4 2017
Certificates Issued:
serial:   0x71
issued:   12:45:52 UTC Thu Jan 3 2008
expired:  12:17:37 UTC Sun Dec 31 2017
status:   Not Revoked
Username: user2
Renewal allowed until: Not Allowed
Number of times user notified: 0
PKCS12 file stored until: 12:27:59 UTC Fri Jan 4 2008
Certificates Issued:
serial:   0x2
issued:   12:27:59 UTC Thu Jan 3 2008
expired:  12:17:37 UTC Sun Dec 31 2017
status:   Not Revoked
<--- More --->
```

## 外部ローカル CA ファイル ストレージの設定

ローカル CA サーバ データベースのローカル CA サーバ設定、ユーザ、発行済み証明書、CRL をフラッシュ メモリまたは外部ローカル CA ファイル システムのいずれかに保存できます。外部ローカル CA ファイル ストレージを設定するには、次の手順を実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **mount** *name* **type**<br><br>**例：**<br>`hostname (config)# mount mydata type cifs` | 特定のファイル システム タイプでコンフィギュレーション モードにアクセスします。 |
| ステップ 2 | **mount** *name* **type cifs**<br><br>**例：**<br>`hostname (config-mount-cifs)# mount mydata type cifs`<br>`server 10.1.1.10 share myshare`<br>`domain example.com`<br>`username user6`<br>`password ********`<br>`status enable` | CIFS ファイル システムをマウントします。<br>**（注）** ファイル システムをマウントするユーザだけが、**no mount** コマンドを使ってアンマウントできます。 |
| ステップ 3 | **crypto ca server**<br><br>**例：**<br>`hostname (config)# crypto ca server` | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |

| | コマンド | 目的 |
|---|---|---|
| ステップ 4 | **database path** *mount-name directory-path*<br><br>**例：**<br>`hostname (config-ca-server)# database path mydata:newuser` | ローカル CA サーバ データベースで使用するマウント済みの CIFS ファイル システムである *mydata* の場所を指定します。サーバへのパスを確立して、ストレージおよび取得に使用するローカル CA ファイルまたはフォルダ名を指定します。ローカル CA ファイル ストレージを ASA フラッシュ メモリに戻すには、**no database path** コマンドを使用します。<br><br>**（注）** 外部サーバに保存されているローカル CA ファイルは、ユーザ名とパスワードが保護されているファイル タイプが CIFS または FTP のマウント済みファイル システムが必要です。 |
| ステップ 5 | **write memory**<br><br>**例：**<br>`hostname (config)# write memory` | 実行コンフィギュレーションを保存します。<br><br>外部ローカル CA ファイル ストレージでは、ASA 設定を保存するたびに、ユーザ情報が ASA からマウント済みのファイル システムおよびファイルの場所である *mydata:newuser* に保存されます。<br><br>フラッシュ メモリ ストレージの場合、ユーザ情報は、スタートアップ コンフィギュレーションのデフォルトの場所に自動的に保存されます。 |

## 例

次の例は、フラッシュ メモリまたは次の外部ストレージに表示されるローカル CA ファイルの例です。

hostname (config-ca-server)# **dir LOCAL* //**

Directory of disk0:/LOCAL*

75　-rwx　32　　13:07:49 Jan 20 2007　LOCAL-CA-SERVER.ser

77　-rwx　229　　13:07:49 Jan 20 2007　LOCAL-CA-SERVER.cdb

69　-rwx　0　　01:09:28 Jan 20 2007　LOCAL-CA-SERVER.udb

81　-rwx　232　　19:09:10 Jan 20 2007　LOCAL-CA-SERVER.crl

72　-rwx　1603　　01:09:28 Jan 20 2007　LOCAL-CA-SERVER.p12

127119360 bytes total (79693824 bytes free)

## CRL のダウンロード

特定のインターフェイスまたはポートで、CRL を HTTP ダウンロードできるようにするには、次のコマンドを実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>`hostname (config)# crypto ca server` | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **publish-crl interface** *interface* **port** *portnumber*<br><br>**例：**<br>`hostname (config-ca-server)# publish-crl outside 70` | インターフェイスのポートを開き、CRL をインターフェイスからアクセスできるようにします。指定したインターフェイスおよびポートを使用して、CRL の着信要求をリッスンします。選択できるインターフェイスと任意のポートは、次のとおりです。<br>• inside：interface/GigabitEthernet0/1 の名前<br>• management：interface/Management0/0 の名前<br>• outside：interface/GigabitEthernet0/0 の名前<br>• ポート番号の範囲は 1 ～ 65535 です。TCP ポート 80 は、HTTP のデフォルト ポート番号です。<br>**（注）** インターフェイスを開いて CRL ファイルをダウンロードするにはこのコマンドが必要であるため、このコマンドを指定しないと、CDP の場所から CRL にアクセスできません。<br><br>CDP URL でインターフェイスの IP アドレスを使用するように設定し、CDP URL およびファイル名のパスも設定できます（http://10.10.10.100/user8/my_crl_file など）。<br>この場合、その IP アドレスが設定されたインターフェイスだけが CRL 要求をリッスンします。要求を受信すると、ASA によってパス /user8/my_crl_file と設定済み CDP URL が照合されます。パスが一致すると、ASA から、保存されている CRL ファイルが返されます。<br>**（注）** プロトコルは必ず HTTP にします。したがって、プレフィックスは http:// です。 |

## CRL の保存

自動的に生成されるローカル CA の CRL に特定の場所を設定するには、シングル コンテキスト モードまたはマルチ コンテキスト モードで次のサイト間タスクを実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>`hostname (config)# crypto ca server` | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **cdp-url** *url*<br><br>**例：**<br>`hostname(config-ca-server)# cdp-url http://172.16.1.1/pathname/myca.crl` | 対象となるすべての証明書に含まれる Cisco Discovery Protocol（CDP）を指定します。CDP に特定の場所を設定しない場合、デフォルトの URL は http://*hostname.domain*/+CSCOCA+/asa_ca.crl になります。<br><br>ローカル CA は、ユーザ証明書が無効化または無効化解除されるたびに、CRL を更新および再発行します。無効化に変更がない場合、CRL のライフタイムごとに 1 回 CRL が再発行されます。<br><br>このコマンドがローカル CA ASA から直接 CRL を処理するように設定されている場合に、インターフェイスのポートを開き、CRL をインターフェイスからアクセスできるようにする手順については、「CRL のダウンロード」（P.35-35）を参照してください。<br><br>CRL は、ローカル CA によって発行された証明書の失効を検証する他のデバイスのためにあります。また、ローカル CA は、自らの証明書データベース内にあるすべての発行済み証明書とステータスを追跡します。検証する機関が、外部サーバから失効ステータスを取得してユーザ証明書を検証する必要がある場合、失効チェックが行われます。この場合、外部サーバは、証明書を発行した CA、または CA が指定したサーバである可能性があります。 |

## 登録パラメータの設定

登録パラメータを設定するには、次のコマンドを実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>`hostname (config)# crypto ca server` | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **otp expiration** *timeout*<br><br>**例：**<br>`hostname(config-ca-server)# otp expiration 24` | ローカル CA 登録ページに対して発行された OTP が有効である期間を時間数で指定します。デフォルトの有効期間は 72 時間です。<br>**（注）** 登録 Web サイトで証明書に登録するためのユーザ OTP をパスワードとして使用して、指定したユーザの発行済み証明書およびキーペアが含まれる PKCS12 ファイルをロック解除することもできます。 |
| ステップ 3 | **enrollment-retrieval** *timeout*<br><br>**例：**<br>`hostname(config-ca-server)# enrollment-retrieval 120` | 登録済みユーザが PKCS12 登録ファイルを取得できる期間を時間単位で指定します。この期間は、ユーザが正常に登録されたときに開始します。デフォルトの取得期間は 24 時間です。取得期間の有効値の範囲は 1 ～ 720 時間です。登録取得期間は、OTP の有効期間とは関係ありません。<br>登録取得期間が過ぎた後、ユーザ証明書とキー ペアは無効になります。ユーザが証明書を受け取る唯一の方法は、管理者が証明書の登録を再開し、ユーザの再ログインを許可することです。 |

## ユーザの追加と登録

ローカル CA データベースに登録できるユーザを追加するには、次のコマンドを実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | `crypto ca server user-db add` *username* `[dn` *dn*`] [email` *emailaddress*`]`<br><br>**例：**<br>`hostname (config-ca-server)# crypto ca server user-db add user1 dn user1@example.com, Engineer, Example Company, US, email user1@example.com` | ローカル CA サーバ ユーザ データベースに新規ユーザを追加します。オプションは次のとおりです。<br>• *username*：4 ～ 64 文字の文字列で、追加するユーザの単純なユーザ名です。ユーザ名には、電子メール アドレスを指定できます。この電子メール アドレスを使用して、登録案内の際に必要に応じてユーザに連絡を取ることができます。<br>• *dn*：認定者名。OSI ディレクトリ（X.500）内のグローバルな正規のエントリ名です（たとえば、cn=user1@example.com、cn=Engineer、o=Example Company、c=US のようになります）。<br>• *e-mail-address*：ワンタイム パスワード（OTP）および通知が送信される、新しいユーザの電子メール アドレスです。 |
| ステップ 2 | `crypto ca server user-db allow` *user*<br><br>**例：**<br>`hostname (config-ca-server)# crypto ca server user-db allow user6` | 新たに追加したユーザにユーザ特権を付与します。 |
| ステップ 3 | `crypto ca server user-db email-otp` *username*<br><br>**例：**<br>`hostname (config-ca-server)# crypto ca server user-db email-otp exampleuser1` | ローカル CA データベースのユーザに、ユーザ証明書を登録およびダウンロードするように通知します。そのユーザには、OTP が自動的に電子メールで送信されます。<br>**（注）** 管理者は、電子メールでのユーザ通知が必要である場合、ユーザを追加するときに、ユーザ名フィールドまたは電子メール フィールドに電子メール アドレスを指定する必要があります。 |

| | コマンド | 目的 |
|---|---|---|
| ステップ 4 | **crypto ca server user-db show-otp**<br><br>**例：**<br>`hostname (config-ca-server)# crypto ca server user-db show-otp` | 対象の OTP を表示します。 |
| ステップ 5 | **otp expiration** *timeout*<br><br>**例：**<br>`hostname (config-ca-server)# otp expiration 24` | 登録の時間制限を時間単位で設定します。デフォルトの有効期間は 72 時間です。**otp expiration** コマンドは、OTP がユーザ登録に有効な期間を定義します。この期間は、ユーザが登録を許可されたときに開始します。<br><br>ユーザが正しい OTP を使って時間制限内に正常に登録すると、ローカル CA サーバによって PKCS12 ファイルが作成されます。これには、そのユーザのキー ペア、生成されたキー ペアの公開キーに基づいたユーザ証明書、およびユーザを追加したときに指定した subject-name DN が含まれます。PKCS12 ファイルの内容は、OTP と呼ばれるパスフレーズによって保護されます。OTP は手動で処理できます。または、管理者が登録を許可した後、このファイルをローカル CA からユーザに電子メールで送信し、ダウンロードすることもできます。<br><br>PKCS12 ファイルは、*username.p12* という名前で一時的なストレージに保存されます。ストレージ内の PKCS12 ファイルを使用して、登録取得期間内に戻り、PKCS12 ファイルを必要な回数だけダウンロードすることができます。登録取得期間が過ぎると、PKCS12 ファイルがストレージから自動的に削除され、ダウンロードできなくなります。<br><br>**（注）** ユーザ証明書が含まれる PKCS12 ファイルを取得する前に登録の有効期間が切れた場合、登録は許可されません。 |

## ユーザの更新

更新通知のタイミングを指定するには、次の手順を実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>`hostname (config)# crypto ca server` | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **renewal-reminder** *time*<br><br>**例：**<br>`hostname (config-ca-server)# renewal-reminder 7` | ローカル CA 証明書の有効期限までの日数（1 ～ 90）を指定します。この日数が経過すると、再登録に関する最初の通知が証明書所有者に送信されます。証明書は、有効期限を過ぎると無効になります。<br><br>電子メールでユーザに送信される更新通知のタイプや送信時機の設定は各種あり、ローカル CA サーバの設定中に管理者が設定できます。<br><br>3 種類の通知が送信されます。ユーザ データベースに電子メール アドレスが指定されている場合、3 種類ある通知ごとに、電子メールが自動的に証明書所有者に送信されます。ユーザの電子メール アドレスを指定していない場合、syslog メッセージが更新要件を警告します。<br><br>ユーザがユーザ データベース内に存在する限り、ASA によって、有効期限間近の有効な証明書を持つすべてのユーザに、証明書の更新特権が自動的に付与されます。したがって、管理者がユーザに自動更新を許可しない場合、更新期間の前にそのユーザをデータベースから削除する必要があります。 |

## ユーザの復元

ローカル CA サーバによって発行され、以前無効にした証明書とユーザを復元するには、次の手順を実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>`hostname (config)# crypto ca server` | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **crypto ca server unrevoke** *cert-serial-no*<br><br>**例：**<br>`hostname (config)# crypto ca server unrevoke 782ea09f` | ユーザを復元し、ローカル CA サーバによって発行され、以前無効にした証明書を無効化解除します。<br><br>ローカル CA では、CRL は、無効になったすべてのユーザ証明書のシリアル番号で保持されます。このリストは外部デバイスで使用でき、**cdp-url** コマンドや **publish-crl** コマンドなどで設定されている場合に、ローカル CA から直接取得することができます。証明書のシリアル番号で、現在の証明書を無効化（または無効化解除）すると、CRL にはそれらの変更が自動的に反映されます。 |

## ユーザの削除

ユーザ データベースからユーザ名によってユーザを削除するには、次の手順を実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>`hostname (config)# crypto ca server` | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **crypto ca server user-db remove** *username*<br><br>**例：**<br>`hostname (config)# crypto ca server user-db remove user1` | ユーザ データベースからユーザを削除し、そのユーザに発行された有効な証明書の無効化を許可します。 |

## 証明書の無効化

ユーザ証明書を無効にするには、次の手順を実行します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **crypto ca server**<br><br>**例：**<br>hostname (config)# crypto ca server | ローカル CA サーバ コンフィギュレーション モードに入ります。ローカル CA の設定と管理を許可します。 |
| ステップ 2 | **crypto ca server revoke** *cert-serial-no*<br><br>**例：**<br>hostname (config-ca-server)# crypto ca server revoke 782ea09f | 16 進数の形式で証明書のシリアル番号を入力します。ローカル CA サーバ上の証明書データベースと CRL で証明書に無効のマークを付けます。CRL は、自動的に再発行されます。<br>**（注）** ASA の証明書を無効にするには、パスワードも必要なので、パスワードは必ず記録し、安全な場所に保管してください。 |

## ローカル CA 証明書データベースのメンテナンス

ローカル CA 証明書データベースを維持するため、データベースに変更が加えられるたびに **write memory** コマンドを使用して、証明書データベース ファイル LOCAL-CA-SERVER.cdb を保存してください。ローカル CA 証明書データベースには、次のファイルが含まれます。

- LOCAL-CA-SERVER.p12 は、ローカル CA サーバを最初にイネーブルにしたときに生成されたローカル CA 証明書とキー ペアのアーカイブです。
- LOCAL-CA-SERVER.crl ファイルは、実際の CRL です。
- LOCAL-CA-SERVER.ser ファイルでは、発行済み証明書のシリアル番号が追跡されます。

## ローカル CA 証明書のロールオーバー

ローカル CA 証明書の有効期限の 30 日前に、ロールオーバー代替証明書が生成され、syslog メッセージ情報で管理者にローカル CA のロールオーバーの時期であることが知らされます。新しいローカル CA 証明書は、現在の証明書が有効期限に達する前に、必要なすべてのデバイスにインポートする必要があります。管理者が、新しいローカル CA 証明書としてロールオーバー証明書をインストールして応答しない場合、検証が失敗する可能性があります。

ローカル CA 証明書は、同じキー ペアを使用して期限満了後に自動的にロールオーバーします。ロールオーバー証明書は、base 64 形式でエクスポートに使用できます。

### 例

次に、base 64 で符号化されたローカル CA 証明書の例を示します。

```
MIIXlwIBAzCCF1EGCSqGSIb3DQEHAaCCF0IEghc+MIIXOjCCFzYGCSqGSIb3DQEHBqCCFycwghcjAgEAMIIXHAYJKo
ZIhvcNAQcBMBsGCiqGSIb3DQEMAQMwDQQIjph4SxJoyTgCAQGAghbw3v4bFy+GGG2dJnB4OLphsUM+IG3SDOiDwZG9
n1SvtMieoxd7Hxknxbum06JDrujWKtHBIqkrm+td34qlNE1iGeP2YC94/NQ2z+4kS+uZzwcRhl1KEZTS1E4L0fSaC3
uMTxJq2NUHYWmoc8pi4CIeLj3h7VVMy6qbx2AC8I+q57+QG5vG5l5Hi5imwtYfaWwPEdPQxaWZPrzoG1J8BFqdPa1j
BGhAzzuSmElm3j/2dQ3Atro1G9nIsRHgV39fcBgwz4fEabHG7/Vanb+fj81d5nlOiJjDYYbP86tvbZ2yOVZR6aKFVI
0b2AfCr6PbwfC9U8Z/aF3BCyM2sN2xPJrXva94CaYrqyotZdAkSYA5KWScyEcgdqmuBeGDKOncTknfgy0XM+fG5rb3
qAXy1GkjyFI5Bm9Do6RUROoG1DSrQrKeq/hj….

END OF CERTIFICATE
```

## ローカル CA サーバ証明書およびキー ペアのアーカイブ

ローカル CA サーバ認証とキー ペアをアーカイブするには、次のコマンドを入力します。

| コマンド | 目的 |
|---|---|
| **copy**<br><br>**例：**<br>`hostname# copy LOCAL-CA-SERVER_0001.pl2 tftp://10.1.1.22/user6/` | FTP または TFTP を使用して、ローカル CA サーバ証明書とキー ペア、およびASAからのすべてのファイルをコピーします。<br>**（注）** すべてのローカル CA ファイルをできるだけ頻繁にバックアップしてください。 |

**（注）**

# デジタル証明書のモニタリング

証明書のコンフィギュレーションとデータベース情報を表示するには、次のコマンドの 1 つまたは複数を入力します。

| コマンド | 目的 |
|---|---|
| **show crypto ca server** | ローカル CA のコンフィギュレーションとステータスを表示します。 |
| **show crypto ca server cert-db** | ローカル CA によって発行されたユーザ証明書を表示します。 |
| **show crypto ca server certificate** | コンソールに base 64 形式でローカル CA 証明書を表示し、使用可能な場合は、他のデバイスへのインポート時に新しい証明書の検証に使うためのロールオーバー証明書のサムプリントを含むロールオーバー証明書の情報を表示します。 |
| **show crypto ca server crl** | CRL を表示します。 |
| **show crypto ca server user-db** | ユーザとユーザのステータスを表示します。この情報に次の修飾子を使用して、表示されるレコード数を減らすことができます。<br>• allowed：現在登録が許可されているユーザだけを表示します。<br>• enrolled：登録され、有効な証明書を持つユーザだけを表示します。<br>• expired：期間満了になった証明書を持つユーザだけを表示します。<br>• on-hold：証明書を持たず現在登録が許可されていないユーザだけを表示します。 |
| **show crypto ca server user-db allowed** | 登録できるユーザを表示します。 |
| **show crypto ca server user-db enrolled** | 有効な証明書を持つ登録済みユーザを表示します。 |
| **show crypto ca server user-db expired** | 期間満了した証明書を持つユーザを表示します。 |
| **show crypto ca server user-db on-hold** | 証明書がなく、登録が許可されていないユーザを表示します。 |

| コマンド | 目的 |
|---|---|
| **show crypto key** *name of key* | 生成したキー ペアを表示します。 |
| **show running-config** | ローカル CA 証明書マップ ルールを表示します。 |

## 例

次の例では、汎用 RSA キーを表示します。

```
hostname/contexta(config)# show crypto key mypubkey
Key pair was generated at: 16:39:47 central Feb 10 2010
Key name: <Default-RSA-Key>
 Usage: General Purpose Key
 Modulus Size (bits): 1024
 Key Data:

30819f30 0d06092a 864886f7 0d010101 05000381 8d003081 89028181 00ea51b7
  0781848f 78bccac2 4a1b5b8d 2f3e30b4 4cae9f86 f4485207 159108c9 f5e49103
  9eeb0f5d 45fd1811 3b4aafce 292b3b64 b4124a6f 7a777b08 75b88df1 8092a9f8
  5508e9e5 2c271245 7fd1c0c3 3aaf1e04 c7c4efa4 600f4c4a 6afe56ad c1d2c01c
  e08407dd 45d9e36e 8cc0bfef 14f9e6ac eca141e4 276d7358 f7f50d13 79020301 0001
Key pair was generated at: 16:34:54 central Feb 10 2010
```

次に、ローカル CA CRL を表示する例を示します。

```
hostname (config)# show crypto ca server crl
Certificate Revocation List:
    Issuer: cn=xx5520-1-3-2007-1
    This Update: 13:32:53 UTC Jan 4 2010
    Next Update: 13:32:53 UTC Feb 3 2010
    Number of CRL entries: 2
    CRL size: 270 bytes
Revoked Certificates:
    Serial Number: 0x6f
    Revocation Date: 12:30:01 UTC Jan 4 2010
    Serial Number: 0x47
    Revocation Date: 13:32:48 UTC Jan 4 2010
```

次に、1 人の保留中のユーザを表示する例を示します。

```
hostname (config)# show crypto ca server user-db on-hold
username: wilma101
email:    <None>
dn:       <None>
allowed:  <not allowed>
notified: 0
hostname (config)#
```

次に、**show running-config** コマンドの出力例を示します。この出力には、ローカル CA 証明書マップ ルールが表示されています。

```
crypto ca certificate map 1
 issuer-name co asc
 subject-name attr ou eq Engineering
```

# 証明書管理の機能履歴

表 35-1 に、各機能変更と、それが実装されたプラットフォーム リリースを示します。

**表 35-1　証明書管理の機能履歴**

| 機能名 | プラットフォーム リリース | 機能情報 |
|---|---|---|
| 証明書管理 | 7.0(1) | デジタル証明書（CA 証明書、ID 証明書、およびコード署名者証明書など）は、認証用のデジタル ID を提供します。デジタル証明書には、名前、シリアル番号、会社、部門、または IP アドレスなど、ユーザまたはデバイスを識別する情報が含まれます。CA は、証明書に「署名」してその認証を確認することで、デバイスまたはユーザのアイデンティティを保証する、信頼できる機関です。CA は、公開キーまたは秘密キーの暗号化を使用してセキュリティを保証する PKI コンテキストで、デジタル証明書を発行します。 |
| 証明書管理 | 7.2(1) | 次のコマンドを導入しました。<br>**issuer-name** *DN-string*、**revocation-check crl none**、**revocation-check crl**、**revocation-check none**。<br>**crl** {**required** \| **optional** \| **nocheck**} コマンドが非推奨になりました。 |

表 35-1　証明書管理の機能履歴 （続き）

| 機能名 | プラットフォーム リリース | 機能情報 |
|---|---|---|
| 証明書管理 | 8.0(2) | 次のコマンドを導入しました。<br>**cdp-url**、**crypto ca server**、**crypto ca server crl issue**、**crypto ca server revoke** *cert-serial-no*、**crypto ca server unrevoke** *cert-serial-no*、**crypto ca server user-db add** *user* [**dn** *dn*] **[email** *e-mail-address*]、**crypto ca server user-db allow** {*username* \| **all-unenrolled** \| **all-certholders**} [**display-otp**] [**email-otp**] [**replace-otp**]、**crypto ca server user-db email-otp** {*username* \| **all-unenrolled** \| **all-certholders**}、**crypto ca server user-db remove** *username*、**crypto ca server user-db show-otp** {*username* \| **all-certholders** \| **all-unenrolled**}、**crypto ca server user-db write**、**[no] database path** *mount-name directory-path*、**debug crypto ca server** [*level*]、**lifetime {ca-certificate** \| **certificate** \| **crl**} *time*、no shutdown、**otp expiration** *timeout*、**renewal-reminder** *time*、**show crypto ca server**、**show crypto ca server cert-db [user** *username* **\| allowed \| enrolled \| expired \| on-hold] [serial** *certificate-serial-number*]、**show crypto ca server certificate**、**show crypto ca server crl**、**show crypto ca server user-db** [**expired** \| **allowed** \| **on-hold** \| **enrolled**]、**show crypto key** *name of key*、**show running-config**、**shutdown**。 |
| SCEP プロキシ | 8.4(1) | サードパーティ CA からのデバイス証明書を安全に構成できる機能を導入しました。<br>次のコマンドを導入しました。<br>**crypto ikev2 enable outside client-services port** *portnumber*, **scep-enrollment enable**, **scep-forwarding-url value** *URL*, **secondary-pre-fill-username clientless hide use-common-password** *password*, **secondary-pre-fill-username ssl-client hide use-common-password** *password*, **secondary-username-from-certificate** {**use-entire-name** \| **use-script** \| {*primary_attr* [*secondary-attr*]}} [**no-certificate-fallback cisco-secure-desktop machine-unique-id**]. |