デジタル証明書

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。

デジタル証明書の概要

デジタル証明書は、認証に使用されるデジタル ID を保持しています。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。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。

デジタル証明書を使用して認証を行う場合は、ASA に 1 つ以上の ID 証明書と、その発行元の CA 証明書が必要です。この設定では、複数のアイデンティティ、ルート、および証明書の階層が許可されます。ASA では CRL（認証局の失効リストとも呼ばれます）に照らしてサードパーティの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。

次に、使用可能な各種デジタル証明書について説明します。

- CA 証明書は、他の証明書に署名するために使用されます。これは自己署名され、ルート証明書と呼ばれます。別の CA 証明書により発行される証明書は、下位証明書と呼ばれます。
- ID 証明書は、特定のシステムまたはホストの証明書です。この証明書も CA により発行されます。

- コード署名者証明書は、コードに署名するためのデジタル署名を作成する際に使用される特殊な証明書であり、署名されたコードそのものが証明書の作成元を示しています。

ローカルCAは、ASAの独立認証局機能を統合したもので、証明書の配布と、発行された証明書に対するセキュアな失効チェックを行います。Web サイトのログイン ページからユーザ登録を行う場合には、ローカル CA により実現されるセキュアで設定可能な内部認証局機能によって、証明書の認証を行うことができます。

（注） CA 証明書および ID 証明書は、サイトツーサイト VPN 接続およびリモート アクセス VPN 接続の両方に適用されます。このマニュアルに記載の手順は、ASDM GUI でリモート アクセス VPN を使用する場合の手順です。

デジタル証明書は、認証に使用されるデジタル ID を保持しています。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。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。

デジタル証明書を使用して認証を行う場合は、ASA に 1 つ以上の ID 証明書と、その発行元の CA 証明書が必要です。この設定では、複数のアイデンティティ、ルート、および証明書の階層が許可されます。次に、使用可能な各種デジタル証明書について説明します。

- CA 証明書は、他の証明書に署名するために使用されます。これは自己署名され、ルート証明書と呼ばれます。
- 別の CA 証明書により発行される証明書は、下位証明書と呼ばれます。

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

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 ペアは、RSA または楕円曲線署名アルゴリズム（ECDSA）キーであり、次の特性があります。

- 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。
- RSA キー サイズの最大値は 4096 で、デフォルトは 2048 です。
- ECDSA キー長の最大値は 521 で、デフォルトは 384 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） Cisco 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。

認証登録

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 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、Cisco 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 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。失効確認を有効にすることにより、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。

失効確認を有効にすると、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 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** コマンドを使用して、CRL チェックをオプションにすることもできます。オプションにすると、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。

CRL のキャッシュに設定された時間を超過して ASA にキャッシュされている CRL がある場合、ASA はその CRL を、古すぎて信頼できない、つまり「失効した」と見なします。ASA は、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、より新しいバージョンの CRL を取得しようとします。

CRL のエントリ制限を超えると、ユーザ接続/証明書で失効チェックエラーが表示されることがあります。CRL あたりの最大エントリ数が 65534 を超えている場合、処理するエントリ数が多すぎることを示すメッセージが syslog から返されます。

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** コマンドを使用して、OCSP チェックをオプションにすることもできます。オプションにすると、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 で基本的な証明機関動作を統合する。
- 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

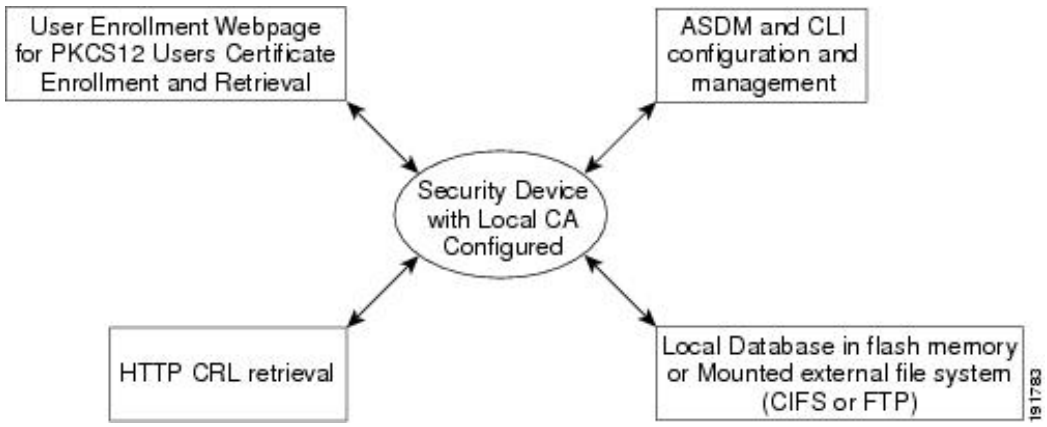
ラッシュ メモリは、3500 人以下のユーザを持つデータベースを保存できますが、ユーザの数がそれを超えるデータベースでは外部ストレージが必要になります。

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。

次の図に示すように、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 *1 :* ローカル *CA*



証明書とユーザ ログイン クレデンシャル

この項では、認証と認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 認可では、パスワードをクレデンシャルとして使用しません。RADIUS 認可では、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。

ユーザ ログイン クレデンシャル

認証および認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。

- 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

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。

認証と認可の両方がイネーブルになっている場合、ASAによって、ユーザの認証と認可の両方にユーザ 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。

- 認証
  - 認可サーバ 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認可がイネーブルになっている場合、ASA によって認可にプライマリ 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。

- 認証
  - 認可サーバ 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、セカンデリ DN のフィールド値が認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）およびセカンデリ 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。

デジタル証明書のガイドライン

この項では、デジタル証明書を設定する前に確認する必要のあるガイドラインおよび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

- 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。
- ローカル CA の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 はサポートされません。

ローカル CA 証明書

- 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 の設定が正しくないと、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。
- 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。
- 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。

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+MIIXOjCCFzYGCSqGSIb3DQEHBqCCFycwghcjAgEAMIIXHA
YJKoZIhvcNAQcBMBsGCiqGSIb3DQEMAQMwDQQIjph4SxJoyTgCAQGAghbw3v4bFy+GGG2dJnB4OLphsUM+IG3S
DOiDwZG9n1SvtMieoxd7Hxknxbum06JDrujWKtHBIqkrm+td34qlNE1iGeP2YC94/NQ2z+4kS+uZzwcRhl1KEZ
TS1E4L0fSaC3uMTxJq2NUHYWmoc8pi4CIeLj3h7VVMy6qbx2AC8I+q57+QG5vG5l5Hi5imwtYfaWwPEdPQxaWZ
PrzoG1J8BFqdPa1jBGhAzzuSmElm3j/2dQ3Atro1G9nIsRHgV39fcBgwz4fEabHG7/Vanb+fj81d5nlOiJjDYY
bP86tvbZ2yOVZR6aKFVI0b2AfCr6PbwfC9U8Z/aF3BCyM2sN2xPJrXva94CaYrqyotZdAkSYA5KWScyEcgdqmu
BeGDKOncTknfgy0XM+fG5rb3qAXy1GkjyFI5Bm9Do6RUROoG1DSrQrKeq/hj….

END OF CERTIFICATE
```

SCEP プロキシ サポート

- ASA と Cisco ISE ポリシー ノードが、同じ NTP サーバを使用して同期されていることを確認します。
- 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。

ローカル CA 証明書データベース

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。

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。キー サイズは 2048 以上を使用することをお勧めします。
- **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への ASDM トラフィックと HTTPS トラフィックを保護するために、アイデンティティ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プロシージャの例については、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。
- ASA でのインポート時は、有効な文字と値だけを証明書パラメータに使用してください。
- ワイルドカード（*）記号を使用するには、文字列値でこの文字を使用できるエンコードを CA サーバで使用していることを確認してください。RFC 5280 では UTF8String または PrintableString を使用することを推奨していますが、PrintableString ではこのワイルドカード文字を有効であると認識しないため UTF8String を使用する必要があります。ASA は、インポート中に無効な文字または値が見つかると、インポートした証明書を拒否します。次に例を示します。

```
ERROR: Failed to parse or verify imported certificate ciscoasa(config)# Read 162*H÷ytes
 as CA certificate:0U0= \Ivr"phÕV°3é¼þ0 CRYPTO_PKI(make trustedCerts list)
CERT-C: E ../cert-c/source/certlist.c(302): Error #711h
CRYPTO_PKI: Failed to verify the ID certificate using the CA certificate in trustpoint
 mm.
CERT-C: E ../cert-c/source/p7contnt.c(169): Error #703h
crypto_certc_pkcs7_extract_certs_and_crls failed (1795):
crypto_certc_pkcs7_extract_certs_and_crls failed
CRYPTO_PKI: status = 1795: failed to verify or insert the cert into storage
```

デジタル証明書の設定

ここでは、デジタル証明書の設定方法について説明します。

キーペアの設定

キー ペアを作成または削除するには、次の手順を実行します。

手順

ステップ 1 1 つのデフォルト汎用 RSA キー ペアを生成します。

crypto key generate rsa modulus 2048

例：

```
ciscoasa(config)# crypto key generate rsa modulus 2048
```

デフォルト キー モジュラスは 2048 ですが、必要なサイズを確実に取得するために、明示的にモジュラスを指定する必要があります。キーの名前は Default-RSA-Key になります。

楕円曲線デジタル署名アルゴリズム（ECDSA）キーも必要な場合は、Default-ECDSA-Key を生成できます。デフォルトの長さは 384 ですが、256 または 521 も使用できます。

crypto key generate ecdsa elliptic-curve 384

ステップ 2 （オプション）一意の名前で追加のキーを作成します。

crypto key generate rsa label *key-pair-label* **modulus** *size*

crypto key generate ecdsa label *key-pair-label* **elliptic-curve** *size*

例：

```
ciscoasa(config)# crypto key generate rsa label exchange modulus 2048
```

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。

RSA キーの場合、モジュラスは 512、768、1024、2048、4096 ビットのいずれかです。

ECDSA キーの場合、楕円曲線は 256、384、521 ビットのいずれかです。

ステップ 3 生成したキー ペアを検証します。

show crypto key mypubkey {**rsa** | **ecdsa**}

例：

```
ciscoasa/contexta(config)# show crypto mypubkey key rsa
```

ステップ 4 生成したキー ペアを保存します。

write memory

例：

```
ciscoasa(config)# write memory
```

ステップ 5 必要に応じて、新しいキー ペアを生成できるように既存のキー ペアを削除します。

crypto key zeroize {**rsa** | **ecdsa**}

例：

```
ciscoasa(config)# crypto key zeroize rsa
```

ステップ 6 （オプション）ローカル CA サーバ証明書およびキー ペアをアーカイブします。

copy

例：

```
ciscoasa# copy LOCAL-CA-SERVER_0001.pl2 tftp://10.1.1.22/user6/
```

このコマンドは、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。

（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。

例

次に、キー ペアを削除する例を示します。

```
ciscoasa(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。

手順

ステップ 1 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 trustpoint *trustpoint-name*

例：

```
ciscoasa/contexta(config)# crypto ca trustpoint Main
```

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。

ステップ 2 次のいずれかのオプションを選択します。

- 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 protocol scep *url*

  例：

```
ciscoasa/contexta(config-ca-trustpoint)# enrollment protocol scep url
http://10.29.67.142:80/certsrv/mscep/mscep.dll
```

- CMP 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 protocol cmp*url*

  例

```
ciscoasa/ contexta(config-ca-trustpoint)# enrollment protocol cmp url
http://10.29.67.142:80/certsrv/mscep/mscep.dll
```

- 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。

  enrollment terminal

```
ciscoasa/contexta(config-ca-trustpoint)# enrollment terminal
```

- 自己署名証明書を要求します。

  enrollment self

ステップ 3 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。

revocation-check crl none

例：

```
ciscoasa/contexta(config-ca-trustpoint)# revocation-check crl none
ciscoasa/contexta(config-ca-trustpoint)# revocation-check crl
ciscoasa/contexta(config-ca-trustpoint)# revocation-check none
```

（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 基本制約の拡張および CA フラグを有効または無効にします。

[**no**] **ca-check**

基本制約の拡張によって、証明書のサブジェクトが認証局（CA）かどうかが識別されます。この場合、証明書を使用して他の証明書に署名することができます。CA フラグは、この拡張の一部です。これらの項目が証明書に存在することは、証明書の公開キーを使用して証明書の署名を検証できることを示します。

ca-check コマンドはデフォルトで有効になっているため、このコマンドは、基本制約と CA フラグを無効にする場合にのみ入力する必要があります。

例：

```
ciscoasa/contexta(config-ca-trustpoint)# no ca-check
```

ステップ 5 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。

email *address*

例：

```
ciscoasa/contexta(config-ca-trustpoint)# email example.com
```

ステップ 6 （オプション）再試行間隔を分単位で指定し、SCEP 登録だけに適用します。

enrollment retry period

例：

```
ciscoasa/contexta(config-ca-trustpoint)# enrollment retry period 5
```

ステップ 7 （オプション）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。

enrollment retry count

例：

```
ciscoasa/contexta(config-ca-trustpoint)# enrollment retry period 2
```

ステップ 8 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。

fqdn *fqdn*

例：

```
ciscoasa/contexta(config-ca-trustpoint)# fqdn example.com
```

ステップ 9 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。

ip-address *ip-address*

例：

```
ciscoasa/contexta(config-ca-trustpoint)# ip-address 10.10.100.1
```

ステップ 10 公開キーが認証の対象となるキー ペアを指定します。

keypair *name*

例：

```
ciscoasa/contexta(config-ca-trustpoint)# keypair exchange
```

ステップ 11 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。

match certificate map-name override ocsp

例：

```
ciscoasa/contexta(config-ca-trustpoint)# match certificate examplemap override ocsp
```

ステップ 12 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。

ocsp disable-nonce

例：

```
ciscoasa/contexta(config-ca-trustpoint)# ocsp disable-nonce
```

ステップ 13 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。

ocsp url

例：

```
ciscoasa/contexta(config-ca-trustpoint)# ocsp url
```

ステップ 14 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。

password *string*

例：

```
ciscoasa/contexta(config-ca-trustpoint)# password mypassword
```

ステップ 15 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。

revocation check

例：

```
ciscoasa/contexta(config-ca-trustpoint)# revocation check
```

ステップ 16 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、この値文字列を二重引用符で囲みます（例：O="Company, Inc."）。

subject-name *X.500 name*

例：

```
ciscoasa/contexta(config-ca-trustpoint)# myname X.500 examplename
```

ステップ 17 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。

serial-number

例：

```
ciscoasa/contexta(config-ca-trustpoint)# serial number JMX1213L2A7
```

ステップ 18 実行コンフィギュレーションを保存します。

write memory

例：

```
ciscoasa/contexta(config)# write memory
```

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。

手順

ステップ 1 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。

crypto ca trustpoint *trustpoint-name*

例：

```
ciscoasa (config)# crypto ca trustpoint Main
```

（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 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。

crl configure

例：

```
ciscoasa(config-ca-trustpoint)# crl configure
```

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　取得ポリシーを設定するには、次のいずれかを選択します。

- 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。

  policy cdp

  ```
  ciscoasa(config-ca-crl)# policy cdp
  ```

  （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。

- CRL は、設定した URL だけから取得できます。

  policy static

  ```
  ciscoasa(config-ca-crl)# policy static
  ```

- 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。

  policy both

  ```
  ciscoasa(config-ca-crl)# policy both
  ```

ステップ 4　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* 引数は、URL に割り当てるランクです。

url *n url*

例：

```
ciscoasa (config-ca-crl)# url 2 http://www.example.com
```

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　CRL 取得方式として HTTP、LDAP、または SCEP を指定します。

protocol http | **ldap** | **scep**

例：

```
ciscoasa(config-ca-crl)# protocol http
```

ステップ 6 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* 引数は、CRL を失効と判断するまで ASA が待機する時間（分）です。

cache-time *refresh-time*

例：

```
ciscoasa(config-ca-crl)# cache-time 420
```

ステップ 7 次のいずれかを選択します。

- CRL に NextUpdate フィールドが存在する必要があります。これがデフォルト設定です。

  enforcenextupdate

  ```
  ciscoasa(config-ca-crl)# enforcenextupdate
  ```

- CRL に NextUpdate フィールドが存在しないことを許可します。

  no enforcenextupdate

  ```
  ciscoasa(config-ca-crl)# no enforcenextupdate
  ```

ステップ 8 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。

ldap-defaults *server*

例：

```
ciscoasa (config-ca-crl)# ldap-defaults ldap1
```

（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 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。

ldap-dn *admin-DN password*

例：

```
ciscoasa (config-ca-crl)# ldap-dn cn=admin,ou=devtest,o=engineering c00lRunZ
```

ステップ 10 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。

crypto ca crl request *trustpoint*

例：

```
ciscoasa (config-ca-crl)# crypto ca crl request Main
```

ステップ 11 実行コンフィギュレーションを保存します。

write memory

例：

```
ciscoasa (config)# write memory
```

トラストポイント設定のエクスポートまたはインポート

トラストポイント設定をエクスポート/インポートするには、次の手順を実行します。

手順

ステップ 1 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。

crypto ca export *trustpoint*

例：

```
ciscoasa(config)# crypto ca export Main
```

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。

ステップ 2 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。

crypto ca import *trustpoint* **pkcs12**

例：

```
ciscoasa(config)# crypto ca import Main pkcs12
```

Base-64形式で端末にテキストを貼り付けるようASAによって促されます。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。

（注） 同じ CA を共有するトラストポイントが ASA 内に複数ある場合、CA を共有するトラストポイントのうち 1 つだけを使用してユーザ証明書を検証できます。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、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。

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

Exported pkcs12 follows:

[ PKCS12 data omitted ]

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

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。

```
ciscoasa (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。

```
ciscoasa (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 ルールの設定

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。作成したルールを使用すると、**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。

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。

手順

ステップ 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、ルールのシーケンス番号を指定します。

crypto ca certificate map [*map_name*]*sequence-number*

例：

```
ciscoasa(config)# crypto ca certificate map test-map 10
```

マップ名を指定しない場合、ルールはデフォルト マップ（DefaultCertificateMap）に追加されます。ルール番号ごとに、一致させるフィールドを 1 つ以上指定できます。

ステップ 2 発行元の名前またはサブジェクト名を指定します。

{**issuer-name** | **subject-name**} [**attr** *attribute*] *operator string*

例：

```
ciscoasa(config-ca-cert-map)# issuer-name cn=asa.example.com
ciscoasa(config-ca-cert-map)# subject-name attr cn eq mycert
ciscoasa(config-ca-cert-map)# subject-name attr uid eq jcrichton
```

値全体と一致させることも、一致させる属性を指定することもできます。有効な値は次のとおりです。

- c：国
- cn：共通名
- dc：ドメイン コンポーネント
- dnq：DN 修飾子
- ea：電子メール アドレス
- genq：世代修飾子
- gn：名
- i：イニシャル
- ip：IP アドレス
- l：局所性
- n：名前
- o：組織名
- ou：組織単位
- ser：シリアル番号

- sn：姓
- sp：都道府県
- t：役職
- uid：ユーザ ID
- uname：非構造化名

有効な演算子は次のとおりです。

- eq：フィールドまたは属性が所定の値と一致する。
- ne：フィールドまたは属性が所定の値と一致しない。
- 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。
- 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 3 サブジェクト代替名を指定します。

alt-subject-name *operator string*

例：

```
ciscoasa(config-ca-cert-map)# alt-subject-name eq happydays
```

有効な演算子は次のとおりです。

- eq：フィールドが所定の値と一致する。
- ne：フィールドが所定の値と一致しない。
- co：フィールドの一部または全部が所定の値と一致する。
- nc：フィールドの全部が所定の値と一致しない。

ステップ 4 拡張キーの使用法を指定します。

extended-key-usage *operator OID_string*

例：

```
ciscoasa(config-ca-cert-map)# extended-key-usage nc clientauth
```

有効な演算子は次のとおりです。

- co：フィールドの一部または全部が所定の値と一致する。
- nc：フィールドの全部が所定の値と一致しない。

有効な OID 文字列は次のとおりです。

- [string]：ユーザ定義の文字列。

- clientauth：クライアント認証（1.3.6.1.5.5.7.3.2）
- codesigning：コード署名（1.3.6.1.5.5.7.3.3）
- emailprotection：セキュア電子メール保護（1.3.6.1.5.5.7.3.4）
- ocspsigning：OCSP 署名（1.3.6.1.5.5.7.3.9）
- serverauth：サーバ認証（1.3.6.1.5.5.7.3.1）
- timestamping：タイムスタンプ（1.3.6.1.5.5.7.3.8）

参照 ID の設定

ASA が TLS クライアントとして動作する場合、ASA は RFC 6125 で定義されているアプリケーション サーバの ID の検証ルールをサポートします。この RFC では、参照 ID を表現（ASA 上で設定）し、（アプリケーション サーバから送信）提示された ID に対して参照 ID を照合する手順を示しています。提示された ID が設定済みの参照 ID と一致しなければ、接続は確立されず、エラーがログに記録されます。

接続の確立中、サーバは自身の ID を提示するために、1 つ以上の識別子を含めたサーバ証明書を ASA に提示します。ASA で設定される参照 ID は、接続の確立中にサーバ証明書で提示される ID と比較されます。これらの ID は、RFC 6125 で定義されている 4 つの ID タイプの特定のインスタンスです。4 つの ID タイプは次のとおりです。

- CN_ID：証明書のサブジェクト フィールドに設定される、共通名（CN）タイプの 1 つの属性タイプと値のペアだけが含まれる相対識別名（RDN）。この値は、完全な形のドメイン名と一致します。CN 値は自由形式のテキストにすることはできません。CN-ID 参照 ID では、アプリケーション サービスは特定されません。
- DNS-ID：dNSName タイプの subjectAltName エントリ。これは DNS ドメイン名です。DNS-ID 参照 ID では、アプリケーション サービスは特定されません。
- SRV-ID：RFC 4985 に定義されている SRVName 形式の名前をもつ、otherName タイプの subjectAltName エントリ。SRV-ID 識別子には、ドメイン名とアプリケーション サービス タイプの両方を含めることができます。たとえば、「_imaps.example.net」の SRV-ID は、DNS ドメイン名部分の「example.net」と、アプリケーション サービス タイプ部分の「imaps」に分けられます。
- URI-ID：uniformResourceIdentifier タイプの subjectAltName エントリ。この値には、「scheme」コンポーネントと、RFC 3986 に定義されている「reg-name」ルールに一致する「host」コンポーネント（またはこれに相当するコンポーネント）の両方が含まれます。URI-ID 識別子には、IP アドレスではなく、およびホスト名だけではなく、DNS ドメイン名を含める必要があります。たとえば、「sip:voice.example.edu」という URI-ID は、DNS ドメイン名の「voice.example.edu」とアプリケーション サービス タイプの「sip」に分割できます。

参照 ID は、未使用の名前を設定すると作成されます。参照 ID が作成されると、4 つの ID タイプと関連付けられた値を参照 ID に追加、または参照 ID から削除することができます。参照 ID には、DNS ドメイン名を特定する情報が含まれている必要があります。また、アプリケーション サービスを特定する情報も含めることができます。

始める前に

- 参照 ID は、syslog サーバおよびスマート ライセンス サーバへの接続時にのみ使用されます。その他の ASA SSL クライアント モードの接続では、現時点では、参照 ID の設定や使用はサポートされていません。
- 対話式クライアントの固定証明書およびフォールバックを除き、ASA は RFC 6125 で説明されている ID と一致させるためのすべてのルールを実装します。
- 証明書を固定する機能は実装されません。したがって、「`No Match Found, Pinned Certificate`」メッセージが発生することはありません。また、シスコで実装するクライアントは対話式クライアントではないため、一致が見つからない場合にユーザが証明書を固定することもできません。

手順

ステップ 1 ASA を ca-reference-identity モードにするには、グローバル コンフィギュレーション モードで **[no] crypto ca reference-identity** コマンドを入力します。

[no] crypto ca reference-identity *reference-identity-name*

この *reference-identity-name* が使用されている参照 ID が見つからない場合、新しい参照 ID が作成されます。使用中の参照 ID に対してこのコマンドの **no** 形式を発行すると、警告メッセージが表示されて、参照 ID は削除されません。

ステップ 2 ca-reference-identity モードで、参照 ID を入力します。参照 ID には、任意のタイプの複数の参照 ID を追加できます。

- **[no] cn-id** *value*
- **[no] dns-id** *value*
- **[no] srv-id** *value*
- **[no] uri-id** *value*

参照 ID を削除するには、このコマンドの no 形式を使用します。

例

syslog サーバの RFC 6125 サーバ証明書の検証に使用する参照 ID を設定します。

```
ciscoasa(config)# crypto ca reference-identity syslogServer
ciscoasa(config-ca-ref-identity)# dns-id syslog1-bxb.cisco.com
ciscoasa(config-ca-ref-identity)# cn-id syslog1-bxb.cisco.com
```

次のタスク

設定した参照 ID は、syslog および Smart Call Home サーバ接続を設定する際に使用します。

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。

始める前に

トラストポイントで示されている CA から、base-64 encoded CA 証明書を取得しておく必要があります。

手順

ステップ 1 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。

crypto ca authenticate *trustpoint*

例：

```
ciscoasa(config)# crypto ca authenticate Main
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB
[ certificate data omitted ]
/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==
quit

INFO: Certificate has the following attributes:
Fingerprint:     24b81433 409b3fd5 e5431699 8d490d34
Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.

% Certificate successfully imported
```

トラストポイントの証明書を手動で取得する必要があるかどうかは、そのトラストポイントの設定時に **enrollment terminal** コマンドを使用するかどうかによって決まります。

ステップ 2 このトラストポイントを持つ ASA を登録します。

crypto ca enroll *trustpoint*

例：

```
ciscoasa(config)# crypto ca enroll Main
% Start certificate enrollment ..

% The fully-qualified domain name in the certificate will be: securityappliance.example.com
```

```
% Include the device serial number in the subject name? [yes/no]: n

Display Certificate Request to terminal? [yes/no]: y
Certificate Request follows:

MIIBoDCCAQkCAQAwIzEhMB8GCSqGSIb3DQEJAhYSRmVyYWxQaXguY2lzY28uY29t
[ certificate request data omitted ]
jF4waw68eOxQxVmdgMWeQ+RbIOYmvt8g6hnBTrd0GdqjjVLt

---End - This line not part of the certificate request---

Redisplay enrollment request? [yes/no]: n
```

このコマンドは、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。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 2 つの証明書要求（キーごとに 1 つ）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。

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。

ステップ 3 CAから受信する各証明書をインポートして、証明書をbase-64形式で端末に貼り付けていることを確認します。

crypto ca import *trustpoint* **certificate**

例：

```
ciscoasa (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
```

ステップ 4 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。

show crypto ca certificate

例：

```
ciscoasa(config)# show crypto ca  certificate Main
```

ステップ 5 実行コンフィギュレーションを保存します。

write memory

例：

```
ciscoasa(config)# write memory
```

ステップ 6 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

この項では、SCEP を使用して証明書を自動的に取得する方法について説明します。

始める前に

トラストポイントで示されている CA から、base-64 encoded CA 証明書を取得しておく必要があります。

手順

ステップ 1 設定したトラストポイントの CA 証明書を取得します。

crypto ca authenticate *trustpoint*

例：

```
ciscoasa/contexta(config)# crypto ca authenticate Main
```

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。

ステップ 2 このトラストポイントを持つ ASA を登録します。このコマンドは、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。

crypto ca enroll *trustpoint*

例：

```
ciscoasa/contexta(config)# crypto ca enroll Main
```

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。

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。

（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 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。

show crypto ca certificate

例：

```
ciscoasa/contexta(config)# show crypto ca certificate Main
```

ステップ 4 実行コンフィギュレーションを保存します。

write memory

例：

```
ciscoasa/contexta(config)# write memory
```

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。

手順

ステップ 1 トンネル グループ ipsec 属性コンフィギュレーション モードを開始します。

tunnel-group *name* **ipsec-attributes**

例：

```
ciscoasa(config)# tunnel-group remotegrp ipsec-attributes
```

ステップ 2 クライアント サービスをイネーブルにします。

crypto ikev2 enable outside client-services port *portnumber*

例：

```
ciscoasa(config-tunnel-ipsec)# crypto ikev2 enable outside client-services
```

デフォルトのポート番号は 443 です。

（注） このコマンドは、IKEv2 をサポートする場合にのみ必要です。

ステップ 3 トンネル グループ general 属性コンフィギュレーション モードを開始します。

tunnel-group *name* **general-attributes**

例：

```
ciscoasa(config)# tunnel-group 209.165.200.225 general-attributes
```

ステップ 4 トンネル グループの SCEP 登録をイネーブルにします。

scep-enrollment enable

例：

```
ciscoasa(config-tunnel-general)# scep-enrollment enable
INFO: 'authentication aaa certificate' must be configured to complete setup of this
option.
```

ステップ 5 グループ ポリシー属性コンフィギュレーション モードを開始します。

group-policy *name* **attributes**

例：

```
ciscoasa(config)# group-policy FirstGroup attributes
```

ステップ 6 グループ ポリシー用の SCEP CA を登録します。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。

scep-forwarding-url value *URL*

例：

```
ciscoasa(config-group-policy)# scep-forwarding-url value http://ca.example.com:80/
```

URL は CA の SCEP URL です。

ステップ 7 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、共通のセカンデリ パスワードを使用します。

secondary-pre-fill-username clientless hide use-common-password *password*

例：

```
ciscoasa(config)# tunnel-group remotegrp webvpn-attributes
ciscoasa(config-tunnel-webvpn)# secondary-pre-fill-username clientless hide
use-common-password secret
```

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。

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。

ステップ 8 AnyConnect VPN セッションの事前入力されているセカンデリ ユーザ名を非表示にします。

secondary-pre-fill-username ssl-client hide use-common-password *password*

例：

```
ciscoasa(config-tunnel-webvpn)# secondary-pre-fill-username ssl-client hide
use-common-password secret
```

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。

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 9 証明書が使用できないときにはユーザ名を指定します。

secondary-username-from-certificate {**use-entire-name** | **use-script** | {*primary_attr* [*secondary-attr*]}} [**no-certificate-fallback cisco-secure-desktop machine-unique-id**]

例：

```
ciscoasa(config-tunnel-webvpn)# secondary-username-from-certificate CN
no-certificate-fallback cisco-secure-desktop machine-unique-id
```

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、次の手順を実行します。

手順

ステップ 1 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ 2 証明書に含める有効期限を決定します。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。

lifetime ca-certificate *time*

例：

```
ciscoasa(config-ca-server)# lifetime ca-certificate 365
```

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 （オプション）ローカル CA 証明書のライフタイムをデフォルト値の 3 年にリセットします。

no lifetime ca-certificate

例：

```
ciscoasa(config-ca-server)# no lifetime ca-certificate
```

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 pre-expiration syslog メッセージが生成されます。

```
%ASA-1-717049: Local CA Server certificate is due to expire in days and a replacement
 certificate is available for export.
```

（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、次の手順を実行します。

手順

ステップ 1 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ 2 ユーザ証明書の有効期間の時間の長さを設定します。

lifetime certificate *time*

例：

```
ciscoasa(config-ca-server)# lifetime certificate 60
```

（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、次の手順を実行します。

手順

ステップ 1 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ 2 CRL の有効期間の時間の長さを設定します。

lifetime crl *time*

例：

```
ciscoasa(config-ca-server)# lifetime crl 10
```

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 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。

crypto ca server crl issue

例：

```
ciscoasa(config-ca-server)# crypto ca server crl issue

A new CRL has been issued.
```

（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、次の手順を実行します。

手順

ステップ1 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ2 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。

keysize server

例：

```
ciscoasa(config-ca-server)# keysize server 2048
```

キーペア サイズのオプションは512、768、1024、2048、4096ビットで、デフォルト値は1024ビットです。

（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 --->
```

特定の証明書タイプの設定方法

信頼できる証明書を確立すると、アイデンティティ証明書の確立などの基本的なタスクや、ローカル CA 証明書やコード署名証明書の確立などのさらに高度な設定を行なえるようになります。

始める前に

デジタル証明書情報に目を通し、信頼できる証明書を確立します。秘密キーが設定されていない CA 証明書は、すべての VPN プロトコルと webvpn で使用され、トラストポイントで着信クライアント証明書を検証するように設定されています。また、トラストポイントとは、HTTPS サーバにプロキシ接続された接続を検証し、smart-call-home 証明書を検証する、webvpn 機能によって使用される信頼できる証明書の一覧のことです。

手順

ローカル CA を設定すると、VPN クライアントが ASA から証明書を直接登録できるようになります。この高度な設定により、ASA は CA に変換されます。CA を設定するには、CA 証明書 （37 ページ） を参照してください。

次のタスク

証明書の有効期限にアラートを設定するか、デジタル証明書や証明書の管理履歴をモニタします。

CA 証明書

このページで、CA 証明書を管理します。次のトピックでは、実行できることについて説明します。

ローカル CA サーバの設定

ローカル CA サーバを設定にするには、次の手順を実行します。

手順

ステップ 1 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ 2 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ワンタイム パスワード（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 アドレスです。

smtp from-address *e-mail_address*

例：

```
ciscoasa(config-ca-server) # smtp from-address SecurityAdmin@example.com
```

ステップ 3 （オプション）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。

subject-name-default *dn*

例：

```
ciscoasa(config-ca-server)# subject-name-default cn=engineer, o=asc systems, c="US"
```

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。

（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 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。

no shutdown

例：

```
ciscoasa(config-ca-server)# no shutdown
```

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。

（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。

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。

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

CA サーバ管理

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、次の手順を実行します。

手順

次のコマンドの 1 つを入力して、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。

- **no crypto ca server**

  例

  ```
  ciscoasa(config)# no crypto ca server
  ```

- **clear configure crypto ca server**

  例

  ```
  ciscoasa(config)# clear config crypto ca server
  ```

（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。

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。

ユーザ証明書の管理

証明書のステータスを変更するには、次の手順を実行します。

手順

ステップ 1 [Manage User Certificates] ペインで、ユーザ名または証明書のシリアル番号で特定の証明書を選択します。

ステップ 2 次のいずれかのオプションを選択します。

- ユーザ証明書のライフタイム期間が終了した場合、[Revoke] をクリックしてユーザアクセスを削除します。また、ローカル CA により、証明書データベース内にあるその証明書に失効のマークが付けられ、情報が自動的に更新されて、CRL が再発行されます。
- 失効した証明書を選択して [Unrevoke] をクリックすると、その証明書に再びアクセスできるようになります。また、ローカル CA により、証明書データベース内にあるその証明書に失効解除のマークが付けられ、証明書の情報が自動的に更新された後、更新されたCRLが再発行されます。

ステップ 3 完了したら [Apply] をクリックして、変更を保存します。

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には、次の手順を実行します。

始める前に

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。

手順

ステップ 1 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ 2 ローカル CA サーバをイネーブルにします。

no shutdown

例：

```
ciscoasa(config-ca-server)# no shutdown
```

このコマンドは、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 ファイルにアーカイブします。英数字で 8 ～ 65 文字のパスワードを入力する必要があります。初期スタートアップ後、パスワード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。

ステップ 3 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。

write memory

例：

```
ciscoasa(config)# write memory
```

例

次の例では、ローカル CA サーバをイネーブルにします。

```
ciscoasa(config)# crypto ca server
ciscoasa(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:
```

trustpool 証明書の自動インポートの設定

スマートライセンスでは、Smart Call Homeインフラストラクチャが使用されます。ASAはバックグラウンドでSmart Call Home匿名レポートを設定するときに、Call Homeサーバ証明書を発行したCAの証明書を含むトラストポイントを自動的に作成します。ASAは、サーバ証明書の発行階層が変更された場合に証明書の検証をサポートするようになりました。カスタマーが証明書階層の変更を調整する必要はありません。CA サーバの自己署名証明書が変更された場合に、Smart Call Home がアクティブな状態を維持できるように、定期的な trustpool バンドルの更新を自動化できます。この機能はマルチ コンテキスト展開ではサポートされません。

trustpoolの証明書バンドルを自動的にインポートするには、ASAがバンドルのダウンロードとインポートに使用する URL を指定する必要があります。次のコマンドを入力すると、デフォルトの Cisco URL とデフォルトの時間（22 時間）を使用して、毎日一定の間隔でインポートが実行されます。

```
ciscoasa(config-ca-trustpool)# auto-import-url Default
```

また、次のコマンドを使用して、カスタム URL による自動インポートをイネーブルにできます。

```
ciscoasa(config-ca-trustpool)# auto-import url http://www.thawte.com
```

オフピーク時またはその他の都合のよい時間帯に柔軟にダウンロードを設定できるようにするには、次のコマンドを入力して、カスタム時間によるインポートをイネーブルにします。

```
ciscoasa(config-ca-trustpool)# auto-import time 23:23:23
```

カスタム URL とカスタム時間の両方による自動インポートを設定するには、次のコマンドを使用する必要があります。

```
ciscoasa(config-ca-trustpool)# auto-import time 23:23:23 url http://www.thawte.com
```

trustpool ポリシーのステータスの表示

trustpool ポリシーの現在のステータスを表示するには、次のコマンドを使用します。

```
show crypto ca trustpool policy
```

このコマンドは次のような情報を返します。

```
0 trustpool certificates installed
Trustpool auto renewal statistics:
 State: Not in progress
 Last import result: Not attempted N/A
 Current Jitter: 0

Trustpool auto import statistics:
 Last import result: N/A
 Next schedule import at 22:00:00 Tues Jul 21 2015
```

```
Trustpool Policy

 Trustpool revocation checking is disabled.
 CRL cache time: 60 seconds
 CRL next update field: required and enforced
 Auto import of trustpool is enabled
 Automatic import URL: http://www.cisco.com/security/pki/trs/ios_core.p7b
 Download time: 22:00:00

Policy Overrides:
 None configured
```

CA Trustpool のクリア

trustpool ポリシーをデフォルト状態にリセットするには、次のコマンドを使用します。

```
clear configure crypto ca trustpool
```

トラストポイント証明書の自動インポートはデフォルトでオフになるので、次のコマンドを使用して機能をディセーブにします。

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、次の手順を実行します。

手順

ステップ 1 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ 2 デフォルト値のないパラメータを指定します。

issuer-name *DN-string*

例：

```
ciscoasa(config-ca-server)# issuer-name cn=xx5520,cn=30.132.0.25,ou=DevTest,ou=QA,o=ASC
 Systems
```

ステップ 3 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。

smtp from-address *e-mail_address*

例：

```
ciscoasa(config-ca-server)# smtp from-address SecurityAdmin@example.com
```

ステップ 4 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。

smtp subject *subject-line*

例：

```
ciscoasa(config-ca-server)# smtp subject Priority E-Mail: Enclosed Confidential Information
 is Required for Enrollment
```

ステップ 5 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。

subject-name-default *dn*

例：

```
ciscoasa(config-ca-server)# subject-name default cn=engineer, o=ASC Systems, c=US
```

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。

許可される DN 属性キーワードは次のとおりです。

- C = 国
- CN = 通常名
- EA = 電子メール アドレス
- L = 地名
- O = 組織名
- OU = 組織ユニット
- ST = 州/都道府県
- SN = 姓名の姓
- ST = 州/都道府県

（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、次の手順を実行します。

手順

ステップ 1 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ 2 ローカル CA サーバをディセーブルにします。

shutdown

例：

```
ciscoasa(config-ca-server)# shutdown
INFO: Local CA Server has been shutdown.
```

このコマンドは、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、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、パスワード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。

外部ローカル CA ファイル ストレージの設定

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。

手順

ステップ 1 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。

mount *name* **type**

例：

```
ciscoasa(config)# mount mydata type cifs
```

ステップ 2 CIFS ファイル システムをマウントします。

mount *name* **type cifs**

例：

```
ciscoasa(config-mount-cifs)# mount mydata type cifs
server 10.1.1.10 share myshare
domain example.com
username user6
password ********
```

```
status enable
```

（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 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ 4 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。

database path *mount-name directory-path*

例：

```
ciscoasa(config-ca-server)# database path mydata:newuser
```

このコマンドは、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。

（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 実行コンフィギュレーションを保存します。

write memory

例：

```
ciscoasa(config)# write memory
```

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 からマウント済みファイル システムおよびファイル場所 *mydata:newuser* に保存されます。

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。

```
ciscoasa(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 のダウンロードおよび保存

CEM コントローラをダウンロードおよび保存するには、次の手順を実行します。

手順

ステップ 1 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ 2 インターフェイスのポートを開き、CRL をその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。

publish-crl interface *interface* **port** *portnumber*

例：

```
ciscoasa(config-ca-server)# publish-crl outside 70
```

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。

- inside：interface/GigabitEthernet0/1 の名前
- management：interface/Management0/0 の名前
- outside：interface/GigabitEthernet0/0 の名前
- 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。

（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。

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）。

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。

（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。

ステップ 3 対象となるすべての証明書に含まれる 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。

cdp-url *url*

例：

```
ciscoasa(config-ca-server)# cdp-url http://172.16.1.1/pathname/myca.crl
```

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。

このコマンドがローカル CA ASA から CRL を直接処理するように設定されている場合に、そのインターフェイスから CRL にアクセスできるようにインターフェイスのポートを開く手順については、CRL のダウンロードおよび保存を参照してください。

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。

登録とユーザ管理

登録パラメータの設定

登録パラメータを設定するには、次の手順を実行します。

手順

ステップ 1 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ 2 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。

otp expiration *timeout*

例：

```
ciscoasa(config-ca-server)# otp expiration 24
```

（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 登録されたユーザが PKCS12 登録ファイルを取得できる時間数を指定します。

enrollment-retrieval *timeout*

例：

```
ciscoasa(config-ca-server)# enrollment-retrieval 120
```

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。

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、次の手順を実行します。

手順

ステップ 1 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。

crypto ca server user-db add *username* [**dn** *dn*] [**email** *emailaddress*]

例：

```
ciscoasa(config-ca-server)# crypto ca server user-db add user1 dn user1@example.com,
Engineer, Example Company, US, email user1@example.com
```

username 引数は、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。

dn 引数は、識別名で、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）。

e-mail-address 引数は、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 新たに追加したユーザにユーザ特権を付与します。

crypto ca server user-db allow *user*

例：

```
ciscoasa(config-ca-server)# crypto ca server user-db allow user
```

ステップ 3 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。

crypto ca server user-db email-otp *username*

例：

```
ciscoasa(config-ca-server)# crypto ca server user-db email-otp exampleuser1
```

（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。

ステップ 4 対象の OTP を表示します。

crypto ca server user-db show-otp

例：

```
ciscoasa(config-ca-server)# crypto ca server user-db show-otp
```

ステップ 5 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 *timeout*

例：

```
ciscoasa(config-ca-server)# otp expiration 24
```

このコマンドは、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。

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。

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。

（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。

手順

ステップ 1 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ 2 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。

renewal-reminder *time*

例：

```
ciscoasa(config-ca-server)# renewal-reminder 7
```

証明書は、有効期限を過ぎると無効になります。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。

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。

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。

手順

ステップ1 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ2 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。

crypto ca server unrevoke *cert-serial-no*

例：

```
ciscoasa(config-ca-server)# crypto ca server unrevoke 782ea09f
```

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。

手順

ステップ1 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ2 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。

crypto ca server user-db remove *username*

例：

```
ciscoasa(config-ca-server)# crypto ca server user-db remove user1
```

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。

手順

ステップ 1 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。

crypto ca server

例：

```
ciscoasa(config)# crypto ca server
```

ステップ 2 16 進数の形式で証明書のシリアル番号を入力します。

crypto ca server revoke *cert-serial-no*

例：

```
ciscoasa(config-ca-server)# crypto ca server revoke 782ea09f
```

このコマンドは、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。

（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、パスワードを必ず記録し、安全な場所に保管してください。

証明書の有効期限アラートの設定（ID 証明書または CA 証明書用）

ASA は、トラストポイントの CA 証明書および ID 証明書について有効期限を24時間ごとに 1 回チェックします。証明書の有効期限がまもなく終了する場合、syslog がアラートとして発行されます。

リマインダおよび繰り返し間隔を設定するためにCLIが提供されます。デフォルトでは、リマインダは有効期限の 60 日前に開始され、7 日ごとに繰り返されます。次のコマンドを使用して、最初のアラートが送信される有効期限までの日数を設定し、リマインダが送信される間隔を設定します。

```
[no] crypto ca alerts expiration [begin <days before expiration>] [repeat <days>]
```

アラートの設定に関係なく、有効期限の直前の週はリマインダが毎日送信されます。次の**show**コマンドと **clear** コマンドも追加されています。

```
clear conf crypto ca alerts
show run crypto ca alerts
```

更新リマインダに加え、コンフィギュレーションに期限が切れた証明書が見つかった場合、その証明書を更新するか、または削除することで、コンフィギュレーションを修正するために syslog が毎日 1 回生成されます。

たとえば、有効期限アラートが 60 日に開始され、その後 6 日ごとに繰り返すように設定されているとします。ASAが 40 日に再起動されると、アラートはその日に送信され、次のアラートは 36 日目に送信されます。

（注） 有効期限チェックは、トラストプールの証明書では実行されません。ローカルCA トラストポイントは、有効期限チェックの通常のトラストポイントとしても扱われます。

デジタル証明書のモニタリング

デジタル証明書ステータスのモニタリングについては、次のコマンドを参照してください。

- **show crypto ca server**

  このコマンドは、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**

  このコマンドは、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**

  このコマンドは、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**

  このコマンドは、CRL を表示します。

- **show crypto ca server user-db**

このコマンドは、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。

- allowed：現在登録が許可されているユーザだけを表示します。
- 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。
- 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。
- 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**

  このコマンドは、登録できるユーザを表示します。

- **show crypto ca server user-db enrolled**

  このコマンドは、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**

  このコマンドは、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**

  このコマンドは、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。

- **show crypto key** *name of key*

  このコマンドは、生成したキー ペアを表示します。

- **show running-config**

  このコマンドは、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。

```
ciscoasa/contexta(config)# show crypto key mypubkey rsa
Key pair was generated at: 16:39:47 central Feb 10 2010
Key name: <Default-RSA-Key>
 Usage: General Purpose Key
 Modulus Size (bits): 2048
 Storage: config
 Key Data:

  30820122 300d0609 2a864886 f70d0101 01050003 82010f00 3082010a 02820101
  00ea2c38 df9c606e ddb7b08a e8b0a1a8 65592d85 0711cac5 fceddee1 fa494297
  525fffc0 90da8a4c e696e44e 0646c661 48b3602a 960d7a3a 52dae14a 5f983603
  e1f33e40 a6ce04f5 9a812894 b0fe0403 f8d7e05e aea79603 2dcd56cc 01261b3e
  93bff98f df422fb1 2066bfa4 2ff5d2a4 36b3b1db edaebf16 973b2bd7 248e4dd2
  071a978c 6e81f073 0c4cd57b db6d9f40 69dc2149 e755fb0f 590f2da8 b620efe6
  da6e8fa5 411a841f e72bb8ea cf4bdb79 f4e57ff3 a940ce3b 4a2c7052 56c1d17b
  af8fe2e2 e58718c6 ed1da0f0 1c6f36eb 79eb1aeb f098b5c4 79e07658 a52d8c7a
  51ceabfb f8ade096 7217cf2d 3728077e 89441d89 9bf5f875 c8d2db39 c858bb7a
  7d020301 0001
```

次に、ローカル CA CRL を表示する例を示します。

```
ciscoasa(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。

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

証明書管理の履歴

表 *1 :* 証明書管理の履歴

機能名	プラットフォーム リリース	説明
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。 **issuer-name** *DN-string*、**revocation-check crl none**、**revocation-check crl**、**revocation-check none**。 **crl** {**required** | **optional** | **nocheck**} コマンドが非推奨になりました。

機能名	プラットフォーム リリース	説明
証明書管理	8.0(2)	次のコマンドを導入しました。 **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**

機能名	プラットフォーム リリース	説明
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。 次のコマンドを導入しました。 **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**]。
参照 ID	9.6(2)	TLS クライアント処理は、RFC 6125 のセクション 6 に定義されるサーバ ID の検証ルールをサポートするようになりました。ID 確認は syslog サーバとスマート ライセンス サーバへの TLS 接続の PKI 確認中に行われます。提示された ID が設定されたリファレンス ID と一致しない場合、接続を確立できません。 次のコマンドが追加または変更されました。**crypto ca reference-identity**、**logging host**、**call home profile destination address**

機能名	プラットフォーム リリース	説明
ローカル CA サーバ	9.12(1)	ASA の構成済み FQDN を使用する代わりに設定可能な登録用 URL の FQDN を作成するため、新しい CLI オプションが導入されました。この新しいオプションは、ccrypto ca server の smpt モードに追加されます。 ローカル CA サーバは廃止され、以降のリリースで削除されます。ASA がローカル CA サーバとして設定されている場合、デジタル証明書の発行、証明書失効リスト（CRL）の発行、および発行された証明書の安全な取り消しを行うために有効になります。この機能は古くなったため、crypto ca server コマンドは廃止されています。