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、ユーザーまたはデバイスを識別する情報が含まれます。CA は、証明書要求の管理とデジタル証明書の発行を行います。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ーのアイデンティティを保証する、信頼できる機関です。

デジタル証明書には、ユーザー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。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。

ルートCA証明書と2つの中間CA証明書を持つ階層では、ID証明書が一方の中間CAによって署名されているが、CRLがもう一方の中間CAによって署名されている場合、9.13以降を実行しているヘッドエンドでリモートアクセスのCRL検証が失敗します。障害は、ヘッドエンドが両方の中間を信頼し、両方が同じルートによって署名されている場合でも発生します。

したがって、各署名者は独自のCRLを維持する必要があります。各署名者は、署名する各証明書のURLリスト内でCRLの場所を指定します。または、正しいCRLの場所を指している各署名者のトラストポイントで、URLオーバーライドを構成することもできます。

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ー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。

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ー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、各ユーザーは、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ーであるため、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 と証明書の管理およびトラックを行えます。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、ユーザー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 が発行したユーザー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。

Cisco ASA トラストポイントに自動証明書管理環境（ACME）プロトコルを設定して、TLS デバイス証明書を管理できます。ACME 対応トラストポイントは、手動の証明書登録もサポート

しています。ACME サーバーは、ASA ポート 80 を使用してドメイン所有権を検証します。ACME は、非クラスタ化シングルコンテキスト ASA 展開でのみサポートされます。

SCEP 要求のプロキシ

ASA は、セキュアクライアント 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、ユーザーが 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、セキュアクライアント 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）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 セキュアクライアント）はサポートしています。

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、サーバー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 サーバー

ASA は次の CA サーバー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。

（注） 9.13(1) で削除された **revocation-check crl none** が復元されました。

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。

（注） CRL サーバは HTTP フラグ「Connection: Keep-alive」で応答して永続的な接続を示しますが、ASA は永続的な接続のサポートを要求しません。リストの送信時に「Connection: Close」と応答するように、CRL サーバの設定を変更します。

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 16 MB のサイズ制限を超えると、ユーザー接続/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。

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。大規模な CRL では、解析に大量の計算オーバーヘッドが必要です。したがって、パフォーマンスを向上させるには、少数の大規模な CRL ではなく、小さいサイズの CRL を多数使用するか、または OCSP を使用することを推奨します。

キャッシュサイズは次のとおりです。

- シングルコンテキストモード：128 MB
- マルチコンテキストモード：コンテキストあたり 16 MB

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 サーバー、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。

（注） 9.13(1) で削除された **revocation-check ocsp none** が復元されました。

OCSP を利用すると、OCSP サーバーの URL を 3 つの方法で定義できます。ASA は、これらのサーバー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 サーバーの URL
2. **ocsp url** コマンドを使用して設定されている OCSP サーバーの 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 サーバー（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。

証明書とユーザー 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ー ログイン クレデンシャル（ユーザー名とパスワード）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、セキュアクライアント、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、すべてのユーザーの共通パスワードまたはユーザー名のいずれかを、パスワードとして使用します。

ユーザー 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、ユーザー 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）の認証サーバー グループ設定によりイネーブルにされます。
  - ユーザー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）の認可サーバー グループ設定によりイネーブルにされます。
  - ユーザー名をクレデンシャルとして使用します。

証明書

ユーザー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 は認証用のユーザー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、ユーザーの認証と認可の両方にユーザー ログイン クレデンシャルが使用されます。

- 認証
  - 認証サーバー グループ設定によってイネーブルにされます。
  - ユーザー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
  - 認可サーバー グループ設定によってイネーブルにされます。
  - ユーザー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
  - 認証サーバー 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
  - 認可サーバー 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ー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ー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ー名は 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。
- ステートフル フェールオーバーを設定すると、証明書は自動的にスタンバイユニットにコピーされます。証明書がない場合は、アクティブユニットで **write standby** コマンドを使用します。

IPv6 のガイドライン

IPv6 OCSP および CRL URL をサポートします。IPv6 アドレスは角カッコで囲む必要があります（例：*http://[0:0:0:0:0.18:0a01:7c16]*）。

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 サーバーを使用して同期されていることを確認します。
- セキュアクライアント 3.0 以降がエンドポイントで実行されている必要があります。
- 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。

その他のガイドライン

- 使用できる証明書のタイプは、証明書を使用するアプリケーションでサポートされている証明書タイプによって制約されます。RSA証明書は通常、証明書を使用するすべてのアプリケーションでサポートされます。ただし、EDDSA証明書は、ワークステーションのオペレーティングシステム、ブラウザ、ASDM、または セキュアクライアント ではサポートされない場合があります。たとえば、リモートアクセスVPNのIDおよび認証にはRSA証明書を使用する必要があります。ASAが証明書を使用するアプリケーションであるサイト間 VPN の場合は、EDDSA がサポートされます。
- ASA が CA サーバー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。

- ASA は、次の認定条件のいずれかが満たされている場合にのみ LDAP/SSL 接続を確立します。
  - LDAP サーバー証明書が信頼されていて（トラストポイントまたは ASA トラストプールに存在する）、有効であること。
  - チェーンを発行しているサーバーからの CA 証明書が信頼されていて（トラストポイントまたは ASA トラストプールに存在する）、チェーン内のすべての下位 CA 証明書が完全かつ有効であること。

- 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。

- 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 と セキュアクライアント で検証できるのは、[X520Serialnumber] フィールド（[サブジェクト名（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。ASA では、これらの証明書が復号化されて内部データ構造に組み込まれます。空白のフィールドがある証明書は、復号化標準に準拠していないと解釈されるため、インストールの検証は失敗します。ただし、バージョン 9.16 以降、オプションフィールドの空白値は、復号化およびインストールの検証基準に影響しません。

- 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 サーバー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
```

- ACME を正常に登録できるようにデバイスで DNS サーバーを設定します。DNS サーバーの設定手順については、DNS サーバーの設定を参照してください。

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。

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

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

エドワーズ曲線署名アルゴリズム（EdDSA）キーも必要な場合は、Default-EdDSA-Key を生成できます。デフォルトの長さは 256 ビットです。

（注）
タイプ EdDSA（Ed25519）のキーペアを使用した ASA での EST 登録はサポートされていません。EST 登録では、RSA または ECDSA キーのみを使用できます。

crypto key generate eddsa edward-curve Ed25519

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。

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 サーバー証明書およびキー 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 サーバー証明書とキー 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。

ステップ 7 自動登録をサポートする登録方式の場合、キーペアは自動的に生成され、登録時にフラッシュに保存されます。このサポートを有効にするには、トラストポイントで次の主要なパラメータを指定します。

次の例は、ECDSA キーペアの設定を示しています。

```
ciscoasa(config-ca-trustpoint)# keypair ?

crypto-ca-trustpoint mode commands/options:
  WORD < 129 char  Name of key pair
  ecdsa            Generate ECDSA keys
  eddsa            Generate EDDSA keys
  rsa              Generate RSA keys
ciscoasa(config-ca-trustpoint)# keypair ecdsa elliptic-curve ?

crypto-ca-trustpoint mode commands/options:
  256  256 bits
  384  384 bits
  521  521 bits
ciscoasa(config-ca-trustpoint)# keypair ecdsa elliptic-curve 521
ciscoasa(config-ca-trustpoint)#
```

次の例は、RSA キーペアの設定を示しています。

```
ciscoasa(config-ca-trustpoint)# keypair ?

crypto-ca-trustpoint mode commands/options:
  WORD < 129 char  Name of key pair
  ecdsa            Generate ECDSA keys
  eddsa            Generate EDDSA keys
  rsa              Generate RSA keys
ciscoasa(config-ca-trustpoint)# keypair ecdsa elliptic-curve ?

ciscoasa(config-ca-trustpoint)# keypair rsa modulus ?

crypto-ca-trustpoint mode commands/options:
  2048  2048 bits
  3072  3072 bits
  4096  4096 bits
ciscoasa(config-ca-trustpoint)# keypair rsa modulus 2048
ciscoasa(config-ca-trustpoint)#
```

（注）
EDDSA キーは ACME に対してサポートされていません。ACME 登録プロトコルがトラストポイントに追加され、ACME 用に設定されたトラストポイントに EDDSA キーペアを追加しようとしたときに、EDDSA キーペアの選択がすでに存在している場合、CLI エラーが発生します。

ステップ 8 登録要求で現在使用されているキーペアと同じキーペアを使用するか、新しいキーペアを生成するかを指定するには、次のコマンドを実行します。

crypto ca enroll my_acme_tp regenerate

次に、現在使用中の同じキーペアを保持するトラストポイントの登録の設定例を示します。

例：

```
ciscoasa(config)# crypto ca enroll my_acme_tp ?
configure mode commands/options:
  noconfirm   Specify this keyword to suppress all interactive prompting.
  regenerate  Regenerate the key pair to be used for enrollment
  <cr>
ciscoasa(config)# crypto ca enroll my_acme_tp noconfirm
```

ステップ 9 次のコマンドを実行して、キーペアを削除できます。

crypto key zeroize rsa

次に、キー ペアを削除する例を示します。

例：

```
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 CA 証明書の送信元インターフェイスを指定します。

enrollment interface *interface-name*

```
ciscoasa(config-ca-trustpoint)#  enrollment interface mgmt
```

ステップ 3 次のいずれかのオプションを選択します。

- ACME 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ます。
  1. 次のコマンドを実行して、登録 URL を設定します。

     enrollment protocol acme *url*

     例：

```
ciscoasa(config-ca-trustpoint)# enrollment protocol acme ?
crypto-ca-trustpoint mode commands/options:
  authentication  ACME authentication method
  url             CA server enrollment URL
ciscoasa(config-ca-trustpoint)# enrollment protocol acme url ?
crypto-ca-trustpoint mode commands/options:
  LINE < 477 char  URL
  LetsEncrypt      Use the Let's Encrypt CA
```

```
ciscoasa(config-ca-trustpoint)#  enrollment protocol acme url
https://mytest.com:8443/acme/acme/directory
```

2. 次のコマンドを実行して、証明書を要求する認証方式とインターフェイスを設定します。

   enrollment protocol acme authentication http01 *inf-name*

```
ciscoasa(config-ca-trustpoint)# enrollment protocol acme ?
crypto-ca-trustpoint mode commands/options:
  authentication  ACME authentication method
  url             CA server enrollment URL

ciscoasa(config-ca-trustpoint)# enrollment protocol acme authentication ?
crypto-ca-trustpoint mode commands/options:
  http01  Use the HTTP-01 method, which opens port 80 on the specified interface

ciscoasa(config-ca-trustpoint)#  enrollment protocol acme authentication http01
 mgmt
```

- 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、次のコマンドを実行することにより、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、次のコマンドを実行することにより、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
```

  （注）
  CMP 登録を有効にするには、キャリアライセンスが含まれている必要があります。

- 次のコマンドを実行して、CA から取得した証明書を端末に貼り付け、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
```

- 次のコマンドを実行して自己署名済み証明書を要求します。

  enrollment self

- EST 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、次のコマンドを実行することにより、登録用 URL を設定します。

enrollment protocol est *url*

例

```
asa(config-ca-trustpoint)# enrollment protocol est ?

crypto-ca-trustpoint mode commands/options:
  url CA server enrollment URL
asa(config-ca-trustpoint)# enrollment protocol est url ?
 crypto-ca-trustpoint mode commands/options:
  LINE < 477 char  URL
asa(config-ca-trustpoint)# enrollment protocol est url https://xyz.com/est
```

ステップ 4 上記のステップでCMPまたはACMEを使用するようにトラストポイントを設定した場合、オプションで自動的に証明書を要求する機能を有効にすることができます。証明書の絶対ライフタイムのうち、自動登録が必要になるまでの期間をパーセンテージで入力し、証明書の再生成中に新しいキーを生成するかどうかを指定します。

```
[no] auto-enroll [<percent>] [regenerate]
```

ステップ 5 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

（注）
9.13(1) で削除された **revocation-check crl none** が復元されました。

例：

```
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 6 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 7 （ACMEに適用不可）登録時に、指定された電子メールアドレスを、証明書のサブジェクト代替名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 8 （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 9 （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 10 登録時に、指定された完全修飾ドメイン名を証明書のサブジェクト代替名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 11 登録時に、複数の完全修飾ドメイン名の値を証明書（または手動、SCEP、CMP、EST、ACME、および自己署名証明書の要求）のサブジェクト代替名子に含めるように CA に要求します。FQDN 値の最大長は 128 文字です。

alt-fqdn *fqdn*

例：

```
ciscoasa/contexta(config-ca-trustpoint)# fqdn primary example.com
ciscoasa/contexta(config-ca-trustpoint)# alt-fqdn example1.com
ciscoasa/contexta(config-ca-trustpoint)# alt-fqdn example2.com
ciscoasa/contexta(config-ca-trustpoint)# alt-fqdn example3.com
```

ステップ 12 （ACME では推奨されない）登録時に、Cisco 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 13 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 14 CMP または ACME にトラストポイントを設定した場合、任意の手動および自動登録に EDDSA キー、EDCSA キーまたは RSA キーを生成するかどうかを決定します。

```
no keypair name | [rsa modulus 2048|4096] | [edcsa elliptic-curve 256|384|521] | [ eddsa
 edwards-curve Ed25519 ]
```

（注）
タイプ EDDSA（Ed25519）のキーペアを使用した ASA での EST 登録はサポートされていません。EST 登録では、RSA キーと ECDSA キーのみを使用できます。キーペアタイプ EDDSA での ACME 登録はサポートされていません。

（注）
ECDHE_ECDSA 暗号グループを使用する場合は、ECDSA 対応キーを含む証明書を使用してトラストポイントを設定します。RSA キーを含む証明書は、ECDSA 暗号と互換性がありません。

ステップ 15 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 16 OCSP に到達するように ASA の送信元インターフェイスを設定します。

interface *nameif*

例：

```
ciscoasa(config)# crypto ca trustpoint TP
ciscoasa(config-ca-trustpoint)# ocsp ?

crypto-ca-trustpoint mode commands/options:
  disable-nonce  Disable OCSP Nonce Extension
  interface      Configure Source interface
  url            OCSP server URL
ciscoasa(config-ca-trustpoint)# ocsp  interface
ciscoasa(config-ca-trustpoint)# ocsp interface ?

crypto-ca-trustpoint mode commands/options:
Current available interface(s):
  inside   Name of interface GigabitEthernet0/0.100
  inside1  Name of interface GigabitEthernet0/0.41
  mgmt     Name of interface Management0/0
  outside  Name of interface GigabitEthernet0/0.51
ciscoasa(config-ca-trustpoint)# ocsp interface mgmt
```

ステップ 17 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 18 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 サーバーを設定します。クライアント証明書の AIA 拡張で指定されているサーバーは使用しません。

ocsp url

例：

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

ASA は、IPv4 と IPv6 両方の OCSP URL をサポートします。IPv6 アドレスは角カッコで囲みます（例：*http://[0:0:0:0:0.18:0a01:7c16/]*）。

ステップ 19 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 20 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。

（注）
失効チェックに OCSP URL を割り当てる場合、OCSP が到達可能なインターフェイス（管理インターフェイスを含む）を指定できます。このインターフェイス値によってルーティングの判断が決まります。

revocation check

例：

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

ステップ 21 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 22 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 23 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。

（注）
ASA は、IPv4 または IPv6 ベースの CDP およびスタティック URL をサポートします。IPv6 アドレスは角カッコで囲みます（例：*http://[0:0:0:0:0.18:0a01:7c16/]*）。

- 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（CDP）URL 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、設定した証明書マップ一致ルール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、設定した証明書マップ一致ルール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 取得用の証明書マップ一致ルールを設定する必要があります。1つのマップに複数のスタティック CDP を設定できるようになりました。

enrollment terminal

特定のインスタンスを削除するには、コマンドの **no** 形式でシーケンス番号または URL を含めます。指定した値が設定値と一致することを確認してください。マップのすべてのエントリを削除するには、**no** コマンドを使用します。

例：

```
ciscoasa(crypto ca trustpoint)#enrollment terminal
```

```
ciscoasa(crypto ca trustpoint)#match certificate Main override cdp  10 url
http://192.0.2.10
ciscoasa(crypto ca trustpoint)#match certificate Main override cdp  20 url
http://192.0.2.12
ciscoasa(crypto ca trustpoint)#match certificate Main override cdp  30 url
http://192.0.2.13
```

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 サーバーを指定します。LDAP サーバーは、DNS ホスト名または IP アドレスで指定できます。LDAP サーバーがデフォルトの 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 サーバーを指定するために、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 サーバー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***passphrase*

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ー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 が発行したユーザー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
- emailAddress：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

  （注）
  subject-name でシリアル番号属性を指定していることを確認します。証明書マップは、subject-name で指定されたシリアル番号属性とのみ一致します。

- sn：姓
- sp：都道府県
- t：役職
- uid：ユーザー 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]：ユーザー定義の文字列。
- 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：サーバー認証（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 サーバーの 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 サーバーから送信）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、サーバーは自身の ID を提示するために、1 つ以上の識別子を含めたサーバー証明書を 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ー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 サーバーおよびスマート ライセンス サーバー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ー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 サーバーの RFC 6125 サーバー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 サーバー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 トラストポイントが CMP 用に設定されている場合、共有秘密値（ir）またはリクエストに署名する証明書を含むトラストポイントの名前（cr）のどちらかを指定できますが、両方を指定することはできません。ASA と交換されるメッセージの信頼性と整合性を確認するための CA からのアウトオブバンド値を指定するか、あるいは CMP 登録要求の署名用に以前に発行されたデバイス証明書をトラストポイントの名前に指定します。共有秘密または署名証明書のキーワードは、トラストポイント登録プロトコルが CMP に設定されている場合にのみ使用できます。

```
crypto ca enroll trustpoint [regenerate] [shared-secret <value> | signing-certificate
<value>
```

ステップ 4 登録要求を作成する前に、新しい鍵ペアを生成すべきかどうかを判断します。

```
crypto ca enroll trustpoint [regenerate] [shared-secret <value> | signing-certificate
<value>
```

ステップ 5 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 6 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 7 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 8 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 グループの 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 tunnel-group general-attributes 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、セキュアクライアント 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 セキュアクライアント VPN セッションの事前入力されているセカンダリユーザー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 を使用するセキュアクライアント 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ー名を指定します。

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 の設定

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 サーバー証明書を発行した 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、サーバー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 サーバー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。

手順

	コマンドまたはアクション	目的
ステップ 1	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`
ステップ 2	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`
ステップ 3	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`
ステップ 4	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。

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ー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、ユーザーとユーザー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ーだけを表示します。
  - enrolled：登録され、有効な証明書を持つユーザーだけを表示します。
  - expired：期間満了になった証明書を持つユーザーだけを表示します。
  - 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ーだけを表示します。

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

  このコマンドは、登録できるユーザーを表示します。

- **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ーを表示します。

- **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ーを表示します。

- **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ーを表示します。

- **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ー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、ユーザー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**
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**]。

機能名	プラットフォームリリース	説明
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ー 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 サーバーとスマートライセンス サーバーへの 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**
ローカル CA サーバー	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、crypto ca server の smpt モードに追加されます。 We deprecated Local CA Server and will be removing in a later release—When ASA is configured as local CA server, it is enabled to issue digital certificates, publish Certificate Revocation Lists（CRLs), and securely revoke issued certificates. 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。
ローカル CA サーバー	9.13(1)	ローカル CA サーバーのサポートが削除されました。したがって、**crypto ca server** コマンドとそのサブコマンドは削除されています。 **crypto ca server** コマンドとそのすべてのサブコマンドが削除されました。
CRL 分散ポイント コマンドの変更	9.13(1)	スタティック CDP URL コンフィギュレーション コマンドが削除され、match certificate コマンドに移行しました。 新規/変更されたコマンド：**crypto-ca-trustpoint crl** と **crl url** はその他の関連ロジックで削除され、**match-certificate override-cdp** が導入されました。

機能名	プラットフォームリリース	説明
CRL キャッシュサイズの拡張	9.13(1)	大規模な CRL ダウンロードの失敗を防ぐため、キャッシュサイズを拡張し、また、個別の CRL 内のエントリ数の制限を取り除きました。 • マルチ コンテキスト モードの場合、コンテキストごとの合計 CRL キャッシュサイズが 16 MB に増加しました。 • シングル コンテキスト モードの場合、合計 CRL キャッシュサイズが 128 MB に増加しました。
証明書有効性チェックをバイパスするオプションの復元	9.15(1)	CRL または OCSP サーバーとの接続問題に起因する失効チェックをバイパスする 9.13(1) で削除されたオプションが復元されました。 新規/変更されたコマンド：**revocation-check crl none**、**revocation-check ocsp none**、**revocation-check crl ocsp none**、**revocation-check ocsp crl none** が復元されました。
スタティック CRL 分散ポイント URL をサポートするための match certificate コマンドの変更	9.15(1)	スタティック CDP URL コンフィギュレーション コマンドでは、スタティック CDP を検証中のチェーン内の各証明書に一意にマッピングできます。ただし、このようなマッピングは各証明書で 1 つだけサポートされていました。今回の変更で、静的に設定された CDP を認証用の証明書チェーンにマッピングできるようになりました。 新規/変更されたコマンド：**match certificate** *map* **override cdp** *seq* **url** *url* and **no match certificate** *map* **override cdp** *seq* **url** *url*
トラストポイントキーペアおよび暗号キー生成コマンドの変更	9.16(1)	2048 より小さいキーサイズの証明書のサポートが削除されました。512、768、または 1024 ビットのオプションを使用する設定は、必要性の通知とともに 2048 に移行されます。 認証に SHA1 ハッシュアルゴリズムを使用するサポートが削除されました。 （注） これらの制限を上書きする **crypto ca permit-weak-crypto** コマンドが導入されました。 新しいキーオプション EDDSA が、既存の RSA および ECDSA オプションに追加されました。

機能名	プラットフォームリリース	説明
OCSP および CRL IPv6 URL のサポート	9.20(1)	IPv6 OCSP および CRL URL を使用するためのサポートが追加されました。IPv6アドレスは角カッコで囲む必要があります。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。