CHAPTER 39

証明書の設定

この章では、証明書の設定方法について説明します。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、ユーザまたはデバイスを識別する情報が含まれています。これらの情報には、名前、シリアル番号、社名、部署、IP アドレスなどがあります。デジタル証明書には、ユーザまたは装置の公開キーのコピーが含まれています。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。

この章は、次の項で構成されています。

公開キー暗号化

この項では、次のトピックについて取り上げます。

公開キー暗号化について

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 暗号化システムをはじめとする公開キー暗号化では、各ユーザが公開キーと秘密キーの両方を含むキー ペアを持ちます。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、アウトオブバンドで取得するか、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は、通信相手のピアごとに各 IPSec ピアを手動で設定する必要があります。また、ネットワークに新しいピアを追加するたびに、セキュアに通信する必要のあるピアごとの設定変更が必要になります。

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。各ピアは、CA によって発行および検証された一意の証明書を送ります。このプロセスが機能を果たすのは、各証明書が関連付けられているピアの公開キーをカプセル化し、各証明書が CA によって認証され、参加しているピアすべてが CA を認証権限者として認識しているためです。このプロセスは、RSA シグニチャ付きの 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 ペアについて

キー ペアは RSA キーです。

- RSA キーは SSH や SSL に使用できます。
- SCEP 登録は、RSA キーの証明書をサポートしています。
- キー生成用の RSA キーの最大係数は 2048 です。デフォルト サイズは 1024 です。
- 署名操作でサポートされているキーの最大サイズは 4096 ビットです。
- 署名にも暗号化にも使用できる汎用 RSA キー ペアを生成することも、署名用と暗号化用に別々の RSA キー ペアを生成することもできます。

  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。これは、SSL は署名用ではなく暗号化用にキーを使用しますが、IKE は暗号化用ではなく署名用にキーを使用するためです。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。

トラストポイントについて

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）　セキュリティ アプライアンスに同じ 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。**support-user-cert-validation** コマンドで、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。これは、異なるセキュリティ アプライアンス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。

失効チェックについて

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。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、セキュリティ アプライアンスがチェックします。

（注）　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。

失効チェックをイネーブルにすると、PKI 証明書の検証プロセス中に、セキュリティ アプライアンスは証明書の失効ステータスを確認します。この場合、CRL チェックか Online Certificate Status Protocol（OCSP）のいずれか、またはその両方を使用できます。両方を使用する場合、2 番目の方法は、1 番目の方法でエラーが返された場合（サーバが使用できない場合など）だけ有効になります。

CRL チェックでは、セキュリティ アプライアンスは失効した証明書の詳細なリストである証明書失効リストを取得、解析、およびキャッシュします。OCSP は失効ステータスを確認する拡張性の高い方法であり、検証局で証明書ステータスをローカライズします。この検証局が特定の証明書のステータスを問い合わせます。

CRL について

証明書失効リストは、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セキュリティ アプライアンスが判断するための 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 CRL チェックを行うように、セキュリティ アプライアンスを設定できます（**revocation-check crl** コマンド）。また、**none** 引数（**revocation-check crl none** コマンド）を追加することで、CRL チェックをオプションにすることもできます。こうすると、更新された CRL データが CA から提供されない場合でも、証明書認証は成功します。

セキュリティ アプライアンスは 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長さの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セキュリティ アプライアンスにキャッシュされている CRL がある場合、セキュリティ アプライアンスはその 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。セキュリティ アプライアンスは、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。

セキュリティ アプライアンスが CRL をキャッシュしておく時間の長さは、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。セキュリティ アプライアンスが 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。

セキュリティ アプライアンスは、これら 2 つの要素を次のように使用します。

- 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、セキュリティ アプライアンスは 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、**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、セキュリティ アプライアンスは 70 分後に CRL に失効のマークを付けます。

セキュリティ アプライアンスがメモリ不足で、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。

トラストポイント用の CRL 動作を設定する方法の詳細については、「トラストポイントの CRL の設定」（P.39-13）を参照してください。

OCSP について

Online Certificate Status Protocol（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セキュリティ アプライアンスが判断するための 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 は、証明書のステータスをチェックする範囲を検証局（OCSP サーバ、*応答側*とも呼ばれます）に限定し、セキュリティ アプライアンスは検証局に特定の証明書のステータスを問い合わせます。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、セキュアなネットワークを拡大できます。

証明書を認証するときに必ず OCSP チェックを行うように、セキュリティ アプライアンスを設定できます（**revocation-check ocsp** コマンド）。また、**none** 引数（**revocation-check ocsp none** コマンド）を追加することで、OCSP チェックをオプションにすることもできます。こうすると、更新された OCSP データが検証局から提供されない場合でも、証明書認証は成功します。

OCSP を導入すると、OCSP サーバの URL を 3 つの方法で定義できます。セキュリティ アプライアンスは、これらのサーバを次の順に使用します。

1. 証明書の照合の上書き規則で定義されている OCSP サーバの URL（**match certificate** コマンド）
2. **ocsp url** コマンドで設定されている 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、同じように設定します。

OCSP サーバ（応答側）証明書は一般に、OCSP 応答に署名します。セキュリティ アプライアンス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。CA は通常、自身の OCSP 応答側証明書のライフタイムを比較的短い期間に設定して、証明書が侵害される可能性を最小限に抑えます。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。しかし、この拡張が含まれていない場合、セキュリティ アプライアンスはトラストポイントに指定されているものと同じ方法で自身の失効ステータスをチェックしようとします。応答側証明書が検証可能でない場合、失効チェックは失敗します。このような失敗を回避するには、トラストポイントを検証する応答側証明書には **revocation-check none** を設定し、クライアント証明書には **revocation-check ocsp** を設定します。

サポート対象の CA サーバ

セキュリティ アプライアンスは次の CA サーバをサポートしています。

- Cisco IOS CS
- Baltimore Technologies
- Entrust
- Microsoft Certificate Services
- Netscape CMS
- RSA Keon
- VeriSign

証明書の設定

この項では、セキュリティ アプライアンスにおける証明書の設定方法と、証明書の使用と管理に関するその他の手順について説明します。

この項では、次のトピックについて取り上げます。

証明書の準備

セキュリティ アプライアンスの証明書を設定する前に、セキュリティ アプライアンスが証明書をサポートするように正しく設定されていることを確認してください。セキュリティ アプライアンスが正しく設定されていないと、登録が失敗したり、不正確な情報が含まれる証明書を登録時に要求したりする可能性があります。

セキュリティ アプライアンスの証明書を準備するには、次の手順を実行します。

ステップ 1　セキュリティ アプライアンス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。**show running-config** コマンドを使用すると、現在設定されているホスト名とドメイン名を表示できます。

ホスト名を設定する方法の詳細については、「ホスト名の設定」（P.8-2）を参照してください。

ドメイン名を設定する方法の詳細については、「ドメイン名の設定」（P.8-2）を参照してください。

ステップ 2　CA を設定する前に、セキュリティ アプライアンス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。セキュリティ アプライアンスが CA に登録して証明書を取得するとき、セキュリティ アプライアンスは現在の時刻が証明書の有効期間の範囲内であるかどうかをチェックします。現在の時刻が有効期間の範囲外の場合、登録は失敗します。

クロックを設定する方法の詳細については、「日付と時刻の設定」（P.8-2）を参照してください。

キー ペアの設定

この項では、次のトピックについて取り上げます。

- 「キー ペアの生成」（P.39-6）
- 「キー ペアの削除」（P.39-7）

キー ペアの生成

「キー ペアについて」（P.39-2）で説明したように、キー ペアとは RSA キーのことです。使用する証明書のタイプに応じてキー ペアを生成する必要があります。

キー ペアを生成するには、次の手順を実行します。

ステップ 1　PKI 実装に必要なタイプのキー ペアを生成します。これには、必要に応じて次の手順を実行します。

a.　RSA キー ペアを生成する場合は、**crypto key generate rsa** コマンドを使用します。

```
hostname/contexta(config)# crypto key generate rsa
```

追加のキーワードを使用しない場合、このコマンドは汎用 RSA キー ペアを 1 つ生成します。キー係数が指定されないため、デフォルトのキー係数である 1024 が使用されます。その他の係数サイズを指定するには、**modulus** キーワードを使用します。**label** キーワードを使用すると、各キー ペアにラベルを割り当てることもできます。このラベルは、キー ペアを使用するトラストポイントによって参照されます。ラベルを割り当てなかった場合、キー ペアには <Default-RSA-Key> というラベルが自動的に付けられます。

```
hostname/contexta(config)# crypto key generate rsa label key-pair-label
```

ステップ 2　（任意）**show crypto key mypubkey** コマンドを使用して、キー ペアを表示します。次の例では、汎用 RSA キーを表示します。

```
hostname/contexta(config)# show crypto key mypubkey
Key pair was generated at: 16:39:47 central Feb 10 2005
Key name: <Default-RSA-Key>
 Usage: General Purpose Key
 Modulus Size (bits): 1024
 Key Data:

  30819f30 0d06092a 864886f7 0d010101 05000381 8d003081 89028181 00ea51b7
  0781848f 78bccac2 4a1b5b8d 2f3e30b4 4cae9f86 f4485207 159108c9 f5e49103
  9eeb0f5d 45fd1811 3b4aafce 292b3b64 b4124a6f 7a777b08 75b88df1 8092a9f8
  5508e9e5 2c271245 7fd1c0c3 3aaf1e04 c7c4efa4 600f4c4a 6afe56ad c1d2c01c
  e08407dd 45d9e36e 8cc0bfef 14f9e6ac eca141e4 276d7358 f7f50d13 79020301 0001
Key pair was generated at: 16:34:54 central Feb 10 2005
```

ステップ 3　生成したキー ペアを保存します。これを行うには、**write memory** コマンドを入力して、実行コンフィギュレーションを保存します。

キー ペアの削除

キー ペアを削除するには、グローバル コンフィギュレーション モードで **crypto key zeroize** コマンドを使用します。

次の例では、RSA キー ペアを削除します。

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

Do you really want to remove these keys? [yes/no] y
hostname(config)#
```

トラストポイントの設定

トラストポイントについては、「トラストポイントについて」（P.39-3）を参照してください。

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。

ステップ 1　セキュリティ アプライアンス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。

```
hostname/contexta(config)# crypto ca trustpoint trustpoint
```

たとえば、Main という名前のトラストポイントを宣言するには、次のようにします。

```
hostname/contexta(config)# crypto ca trustpoint Main
hostname/contexta(config-ca-trustpoint)#
```

このコマンドを入力すると、Crypto ca トラストポイント コンフィギュレーション モードに入ります。

ステップ 2　このトラストポイントで使用する登録方式を指定します。

登録方式を指定するには、次のいずれかを実行します。

- SCEP 登録を指定するには、**enrollment url** コマンドを使用して、宣言したトラストポイントで SCEP 登録に使用する URL を設定します。たとえば、セキュリティ アプライアンスが URL http://10.29.67.142:80/certsrv/mscep/mscep.dll を使用してトラストポイント Main に証明書を要求する場合、コマンドは次のようになります。

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

- 手動登録を指定するには、**enrollment terminal** コマンドを使用して、CA から取得した証明書を端末に貼り付けることを指定します。

ステップ 3　必要に応じて、別の特性をトラストポイントに指定できます。定義する必要のある特性は、CA とそのコンフィギュレーションによって異なります。トラストポイントの特性を指定するには、次のコマンドを入力します。これらのコマンドの完全な説明と使用上のガイドラインについては、『*Cisco Security Appliance Command Reference*』を参照してください。

- **accept-subordinates**：このトラストポイントに関連付けられている CA よりも下位の CA 証明書が、事前にデバイスにインストールされていない状態でフェーズ 1 の IKE 交換時に配信された場合に、受け入れられるかどうかを示します。
- **crl required | optional | nocheck**：CRL コンフィギュレーション オプションを指定します。コマンド文内に含まれる **optional** キーワードを使用して **crl** コマンドを入力すると、セキュリティ アプライアンスが CRL にアクセスできない場合でも、セキュリティ アプライアンスはピアからの証明書を受け入れることができます。

（注）　必須またはオプションの CRL チェックをイネーブルにする場合、必ず CRL 管理用のトラストポイントを設定してください。これは、証明書を取得した後に実行する必要があります。トラストポイントの CRL 管理を設定する方法の詳細については、「トラストポイントの CRL の設定」（P.39-13）を参照してください。

- **crl configure**：CRL コンフィギュレーション モードに入ります。
- **default enrollment**：すべての登録パラメータをシステム デフォルト値に戻します。このコマンドの呼び出しは、アクティブなコンフィギュレーションには含まれません。
- **email** *address*：登録中に、指定した電子メール アドレスを証明書のサブジェクト代替名の拡張に含めるかどうかを CA に確認します。
- **enrollment retry period**：（任意）リトライ間隔を分単位で指定します。この特性は、SCEP 登録を使用している場合にだけ適用されます。
- **enrollment retry count**：（任意）許可されるリトライの最大回数を指定します。この特性は、SCEP 登録を使用している場合にだけ適用されます。
- **enrollment terminal**：このトラストポイントへのカット アンド ペースト登録を指定します。
- **enrollment url** *URL*：このトラストポイントで使用する自動登録（SCEP）を指定し、登録 URL を設定します。
- **fqdn** *fqdn*：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。
- **id-cert-issuer**：このトラストポイントに関連付けられている CA が発行したピア証明書を、システムが受け入れるかどうかを示します。
- **ip-address** *ip-address*：登録時に、セキュリティ アプライアンスの IP アドレスを証明書に含めるように CA に要求します。
- **keypair** *name*：公開キーが証明対象となるキー ペアを指定します。

- **match certificate map**：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。
- **ocsp disable-nonce**：OCSP 要求の nonce 拡張をディセーブルにします。nonce 拡張は、リプレイ攻撃を防ぐために、要求と応答を暗号化してバインドします。
- **ocsp url**：セキュリティ アプライアンスで、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。
- **password** *string*：登録中に CA に登録されるチャレンジ フレーズを指定します。通常、CA は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。
- **revocation-check**：失効チェックの方法（CRL、OCSP、および none）を 1 つ以上設定します。
- **subject-name** *X.500 name*：登録中に、指定したサブジェクト DN を証明書に含めるかどうかを CA に確認します。
- **serial-number**：登録時に、セキュリティ アプライアンス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。
- **support-user-cert-validation**：イネーブルになっている場合、リモート ユーザ証明書を検証するコンフィギュレーション設定が、このトラストポイントから取得されます。ただし、このトラストポイントが、そのリモート ユーザ証明書を発行した CA に認証されることが前提です。
- **exit**：このモードを終了します。

ステップ 4　トラストポイント コンフィギュレーションを保存します。これを行うには、**write memory** コマンドを入力して、実行コンフィギュレーションを保存します。

証明書の取得

セキュリティ アプライアンスは、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、セキュリティ アプライアンス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。

セキュリティ アプライアンスは、SCEP を使用した登録と、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、各セキュリティ アプライアンス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、各セキュリティ アプライアンスと各リモート アクセス VPN クライアントを登録する必要があります。

この項では、次のトピックについて取り上げます。

- 「SCEP を使用した証明書の取得」（P.39-9）
- 「手動での証明書の取得」（P.39-11）

SCEP を使用した証明書の取得

ここでは、SCEP を使用して証明書を設定するための手順について説明します。自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。この手順を完了したときに、セキュリティ アプライアンスはトラストポイントの CA 証明書と、署名と暗号化の目的で 1 つまたは 2 つの証明書を受信しています。汎用 RSA キーを使用する場合は、署名用と暗号化用の証明書を受信します。署名と暗号化に別々の RSA キーを使用する場合は、セキュリティ アプライアンスは署名用と暗号化用の別々の証明書を受信します。

（注） トラストポイントが証明書の取得に SCEP を使用するかどうかは、そのトラストポイントを設定するときに **enrollment url** コマンドを使用するかどうかによって決まります（「トラストポイントの設定」（P.39-7）を参照）。

SCEP を使用して証明書を取得するには、次の手順を実行します。

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。

```
hostname/contexta(config)# crypto ca authenticate trustpoint
```

たとえば、下位 CA を表す Main という名前のトラストポイントを使用する場合は、次のようになります。

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

INFO: Certificate has the following attributes:
Fingerprint:     3736ffc2 243ecf05 0c40f2fa 26820675
Do you accept this certificate? [yes/no]: y

Trustpoint 'Main' is a subordinate CA and holds a non self signed cert.
Trustpoint CA certificate accepted.
```

ステップ 2　このトラストポイントを持つセキュリティ アプライアンスを登録します。このプロセスでは、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。

ステップ 3　登録を行うには、**crypto ca enroll** コマンドを使用します。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。

```
hostname(config)# crypto ca enroll trustpoint
```

セキュリティ アプライアンス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、証明書要求を再送信します。セキュリティ アプライアンスは、証明書を受信するまで、1 分ごとに証明書要求を送信し続けます。

（注） トラストポイントの完全修飾ドメイン名がセキュリティ アプライアンス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。必要に応じて、登録プロセスを終了し、必要な修正を行ってから、再度 **crypto ca enroll** コマンドを入力します。

次の例では、Main という名前のトラストポイントで登録を実行しています。

```
hostname(config)# crypto ca enroll Main
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
% password to the CA Administrator in order to revoke your certificate.
% For security reasons your password will not be saved in the configuration.
% Please make a note of it.
Password: 2b0rn0t2b
Re-enter password: 2b0rn0t2b
% The subject name in the certificate will be: securityappliance.example.com
% The fully-qualified domain name in the certificate will be:
securityappliance.example.com
% Include the device serial number in the subject name? [yes/no]: no
Request certificate from CA [yes/no]: yes
% Certificate request sent to Certificate authority.
```

（注）　セキュリティ アプライアンスの証明書を無効にする場合はパスワードが必要になるため、このパスワードを覚えておくことが不可欠です。パスワードは書き留めて安全な場所に保管してください。

セキュリティ アプライアンスが登録する必要のあるトラストポイントごとに、**crypto ca enroll** コマンドを入力する必要があります。

（注）　**crypto ca enroll** コマンドを発行した後、証明書を受信する前にセキュリティ アプライアンスがリブートされた場合は、**crypto ca enroll** コマンドを再発行して、CA 管理者に連絡してください。

ステップ 4　**show crypto ca certificate** コマンドを使用して、登録プロセスが成功したことを確認します。たとえば、トラストポイント Main から受信した証明書を表示するには、次のように入力します。

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

このコマンドの出力で、セキュリティ アプライアンスに対して発行された証明書とトラストポイントの CA 証明書の詳細が表示されます。

ステップ 5　**write memory** コマンドを使用して、コンフィギュレーションを保存します。

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

手動での証明書の取得

ここでは、手動による証明書要求を設定するための手順について説明します。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。この手順を完了したときに、セキュリティ アプライアンスはトラストポイントの CA 証明書と、署名と暗号化の目的で 1 つまたは 2 つの証明書を受信しています。汎用 RSA キーを使用する場合は、署名用と暗号化用の証明書を受信します。署名と暗号化に別々の RSA キーを使用する場合、受信した証明書はそれぞれ署名専用と暗号化専用に使用されます。

（注）　トラストポイントが手動で証明書を取得する必要があるかどうかは、そのトラストポイントを設定するときに **enrollment terminal** コマンドを使用するかどうかによって決まります（「トラストポイントの設定」（P.39-7）を参照）。

証明書を手動で取得するには、次の手順を実行します。

ステップ 1　トラストポイントで示されている CA から、base-64 encoded CA 証明書を取得します。

ステップ 2　CA 証明書をインポートします。これを行うには、**crypto ca authenticate** コマンドを使用します。次の例は、トラストポイント Main の CA 証明書要求を示しています。

```
hostname (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
hostname (config)#
```

ステップ 3　証明書要求を生成します。これを行うには、**crypto ca enroll** コマンドを使用します。次の例は、トラストポイント Main の証明書と暗号キーの要求を示しています。トラストポイント Main は、手動登録を使用し、署名用と暗号化用の汎用 RSA キーを使用するように設定されています。

```
hostname (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
hostname (config)#
```

（注）　署名と暗号化に別々の RSA キーを使用する場合、**crypto ca enroll** コマンドは各キーに対する 2 つの証明書要求を表示します。登録を完了するには、**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。

ステップ 4　**crypto ca enroll** コマンドで生成される要求ごとに、該当するトラストポイントで示される CA から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 5　CA から受信した証明書ごとに、**crypto ca import certificate** コマンドを使用します。セキュリティ アプライアンスは、証明書を base-64 形式で端末に貼り付けることを求めるプロンプトを表示します。

（注）　署名と暗号化に別々の RSA キー ペアを使用する場合は、証明書ごとにこのステップを実行してください。セキュリティ アプライアンスは、証明書が署名キー ペア用か暗号キー ペア用かを自動的に判断します。2 つの証明書をインポートする順序は、どちらが先でも関係ありません。

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

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

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

```
hostname (config)#
```

ステップ 6　**show crypto ca certificate** コマンドを使用して、登録プロセスが成功したことを確認します。たとえば、トラストポイント Main から受信した証明書を表示するには、次のように入力します。

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

このコマンドの出力で、セキュリティ アプライアンスに対して発行された証明書とトラストポイントの CA 証明書の詳細が表示されます。

ステップ 7　**write memory** コマンドを使用して、コンフィギュレーションを保存します。

```
hostname/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 チェックを行う場合は、トラストポイントごとに CRL を設定する必要があります。CRL の詳細については、「CRL について」（P.39-3）を参照してください。

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。

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 トラストポイント コンフィギュレーション モードに入ります。これを行うには、**crypto ca trustpoint** コマンドを入力します。

ステップ 2　まだ CRL をイネーブルにしていない場合は、**crl** コマンドに **required** キーワードまたは **optional** キーワードを指定して、イネーブルにします。**required** キーワードを指定すると、CRL が使用できない場合、このトラストポイントの証明書認証は失敗します。

ステップ 3　**crl configure** コマンドを入力します。

```
hostname/contexta(config-ca-trustpoint)# crl configure
hostname/contexta(config-ca-crl)#
```

このコマンドを入力すると、現在のトラストポイントに対して crl コンフィギュレーション モードに入ります。

ヒント　すべての CRL コンフィギュレーションのオプション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 の設定中に、やり直す必要がある場合は、いつでもこのコマンドを入力して、この手順を再度開始できます。

ステップ 4　**policy** コマンドで、取得ポリシーを設定します。このコマンドの次のキーワードによって、取得ポリシーが決まります。

- **cdp**：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。

  （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。

- **static**：CRL は、設定した URL だけから取得できます。
- **both**：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。

ステップ 5　CRL ポリシーを設定したときに static キーワードまたは both キーワードを使用した場合は、**url** コマンドを使用して、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。

```
hostname/contexta(config-ca-crl)# url n URL
```

n は、URL に割り当てられるランクです。URL を削除するには、**no url** *n* コマンドを使用します。

ステップ 6　**protocol** コマンドで、取得方式を設定します。このコマンドの次のキーワードによって、取得方式が決まります。

- **http**：CRL 取得方式として HTTP を指定します。
- **ldap**：CRL 取得方式として LDAP を指定します。
- **scep**：CRL 取得方式として SCEP を指定します。

ステップ 7　セキュリティ アプライアンス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。セキュリティ アプライアンスが CRL を失効と判断するまで待機する時間を分数で指定するには、次のコマンドを入力します。

```
hostname/contexta(config-ca-crl)# cache-time n
```

n は分数です。たとえば、CRL を 7 時間キャッシュする必要がある場合は、次のコマンドを入力します。

```
hostname/contexta(config-ca-crl)# cache-time 420
```

ステップ 8　セキュリティ アプライアンスに CRL の NextUpdate フィールドが必要かどうかを設定します。セキュリティ アプライアンスが [NextUpdate] フィールドをどのように使用するかについては、「CRL について」（P.39-3）を参照してください。

次のいずれかを実行します。

- [NextUpdate] フィールドが必要な場合は、**enforcenextupdate** コマンドを入力します。これがデフォルト設定です。
- CRL の [NextUpdate] フィールドが不要な場合は、**no enforcenextupdate** コマンドを入力します。

ステップ 9　取得プロトコルとして LDAP を指定した場合は、次の手順を実行します。

a.　次のコマンドを入力して、セキュリティ アプライアンスが LDAP サーバを認識するようにします。

```
hostname/contexta(config-ca-crl)# ldap-defaults server
```

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。たとえば、次のコマンドは、ホスト名が ldap1 の LDAP サーバから CRL を取得するようにセキュリティ アプライアンスを設定します。

```
hostname/contexta(config-ca-crl)# ldap-defaults ldap1
```

（注）　LDAP サーバの指定に IP アドレスではなくホスト名を使用する場合は、DNS を使用するようにセキュリティ アプライアンスを設定していることを確認してください。DNS を設定する方法については、『*Cisco Security Appliance Command Reference*』の **dns** コマンドを参照してください。

b.　LDAP サーバに CRL 取得を許可するクレデンシャルが必要な場合は、次のコマンドを入力します。

```
hostname/contexta(config-ca-crl)# ldap-dn admin-DN password
```

次に例を示します。

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

ステップ 10　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するには、**crypto ca crl request** コマンドを使用します。このコマンドは、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 を取得します。

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** コマンドを入力します。

トラストポイントのエクスポートとインポート

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は、エクスポートおよびインポートできます。セキュリティ アプライアンスは、トラストポイントのエクスポートとインポートにおいて、PKCS12 形式をサポートしています。

この項では、次のトピックについて取り上げます。

- 「トラストポイント コンフィギュレーションのエクスポート」（P.39-15）
- 「トラストポイント コンフィギュレーションのインポート」（P.39-15）

トラストポイント コンフィギュレーションのエクスポート

関連付けられているすべてのキーや証明書とともに、トラストポイント コンフィギュレーションを PKCS12 形式でエクスポートするには、**crypto ca export** コマンドを使用します。セキュリティ アプライアンスは 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。

次の例では、パスフレーズとして Wh0zits を使用して、PKCS12 データをトラストポイント Main にエクスポートしています。

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

Exported pkcs12 follows:

[ PKCS12 data omitted ]

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

hostname (config)#
```

トラストポイント コンフィギュレーションのインポート

トラストポイント コンフィギュレーションに関連付けられているキー ペアと発行済み証明書をインポートするには、グローバル コンフィギュレーション モードで **crypto ca import pkcs12** コマンドを使用します。セキュリティ アプライアンスは、テキストを base-64 形式で端末に貼り付けることを求めるプロンプトを表示します。

トラストポイントとともにインポートされたキー 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。たとえば、エクスポートされたトラストポイントが <Default-RSA-Key> というラベルの付いた RSA キーを使用していた場合、PKCS12 をインポートして Main という名前のトラストポイントを作成すると、作成されるキー ペアの名前は <Default-RSA-Key> ではなく Main になります。

（注）　セキュリティ アプライアンスに同じ CA を共有するトラストポイントが複数ある場合、CA を共有するトラストポイントでユーザ証明書の検証に使用できるのは 1 つだけです。**crypto ca import pkcs12** コマンドを使用すると、このような状況を構成できます。**support-user-cert-validation** コマンドで、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します。

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

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

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

CA 証明書マップ規則の設定

証明書の [Issuer] フィールドと [Subject] フィールドに基づいて、ルールを設定できます。作成した規則を使用すると、**tunnel-group-map** コマンドによって、IPSec ピアの証明書をトンネル グループにマッピングできます。セキュリティ アプライアンスは CA 証明書マップを 1 つサポートしています。これには、複数の規則を設定できます。トンネル グループで CA 証明書マップ規則を使用する方法の詳細については、「証明書グループ照合のルールとポリシーの作成」（P.27-10）を参照してください。

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。

ステップ 1　設定する規則に対して CA 証明書マップ コンフィギュレーション モードに入ります。これには、**crypto ca certificate map** コマンドを入力し、規則インデックス番号を指定します。次の例では、インデックス番号 1 の規則に対して CA 証明書マップ モードに入っています。

```
hostname(config)# crypto ca certificate map 1
hostname(config-ca-cert-map)#
```

ステップ 2　規則を設定するには、**issuer-name** コマンドと **subject-name** コマンドを使用します。これらのコマンドは、セキュリティ アプライアンスが証明書の [Issuer] フィールドまたは [Subject] フィールドの値に適用できるテストを指定します。これらのテストは、特定の属性に適用することや、[Issuer] フィールドまたは [Subject] フィールド全体に適用することができます。ルールごとに複数のテストを設定できますが、これらのコマンドで指定するすべてのテストは、証明書と一致するルールで真である必要があります。**issuer-name** コマンドと **subject-name** コマンドの有効な演算子は、次のとおりです。

演算子	意味
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。

issuer-name コマンドと **subject-name** コマンドの詳細については、『*Cisco Security Appliance Command Reference*』を参照してください。

次の例では、Issuer フィールド内の属性に文字列 cisco が含まれるように指定しています。

```
hostname(config-ca-cert-map)# issuer-name co cisco
hostname(config-ca-cert-map)#
```

次の例では、Subject フィールド内の Organizational Unit 属性が文字列 Engineering と完全に一致しなければならないことを指定しています。

```
hostname(config-ca-cert-map)# subject-name attr ou eq Engineering
hostname(config-ca-cert-map)#
```

マップ規則は、**show running-config** コマンドの出力に表示されます。

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

ステップ 3　マップ規則の設定が完了したら、作業内容を保存します。**write memory** コマンドを入力します。