認証局の相互運用性

この章では、IPSecプロトコルをサポートするために提供される、認証局（CA）の相互運用性を設定する方法について説明します。CA の相互運用性により、Cisco IOS デバイスと CA の通信が可能になり、Cisco IOS デバイスが CA からデジタル証明書を取得して使用できるようになります。IPSec は CA を使用せずにネットワークで実装できますが、CA を使用すると、IPSec の管理性と拡張性が提供されます。

認証局の前提条件

この相互運用性機能の設定を行う前に、ネットワークで認証局（CA）が使用可能になっている必要があります。CA が公開キーインフラストラクチャ（PKI）プロトコルと Simple Certificate Enrollment Protocol（SCEP）プロトコルをサポートしている必要があります。

認証局の制約事項

CA を設定する際には次の制約事項が適用されます。

- この機能を設定する必要があるのは、ネットワークに IPSec およびインターネットキー交換（IKE）を両方とも設定する場合だけです。
- Cisco IOS ソフトウェアでは、長さが 2048 ビットを超える CA サーバ公開キーはサポートされていません。

認証局について

CA でサポートされる規格

認証局（CA）の相互運用性がなければ、Cisco IOS デバイスは IPSec 実装時に CA を使用することができません。CA は、IPSec ネットワークに管理可能なスケーラブル ソリューションを提供します。

シスコでは、この機能で次の規格をサポートしています。

- IPsec：IPsec は、参加しているピア間のデータ機密性、データ整合性、およびデータ認証を提供するオープン スタンダードのフレームワークです。IPSec は、IP レイヤでこれらのセキュリティ サービスを提供し、インターネット キー交換を使用して、ローカルポリシーに基づいたプロトコルとアルゴリズムのネゴシエーションの処理を行い、IPSec で使用する暗号化および認証キーを生成します。IPSec を使用することにより、ホスト ペア間、セキュリティ ゲートウェイ ペア間、またはセキュリティ ゲートウェイとホスト間の 1 つ以上のデータ フローを保護できます。
- インターネット キー交換（IKE）：Oakley キー交換や Skeme キー交換をインターネット セキュリティ アソシエーション キー管理プロトコル（ISAKMP）フレームワーク内部に実装したハイブリッドプロトコルです。IKE は他のプロトコルで使用できますが、その初期実装時は IPSec プロトコルで使用します。IKE は、IPsec ピアの認証、IPsec キーのネゴシエーションを提供し、IPsec セキュリティ アソシエーションのネゴシエーションを実行します。
- Public-Key Cryptography Standard #7（PKCS #7）：証明書登録メッセージの暗号化および署名に使用される RSA Data Security, Inc. の標準。
- Public-Key Cryptography Standard #10（PKCS #10）：証明書要求のための RSA Data Security, Inc. の標準構文。
- RSA キー：RSA は公開キー暗号化システムで、Ron Rivest、Adi Shamir、Leonard Adleman の 3 名によって開発されました。RSA キーは、1 つの公開キーと 1 つの秘密キーのペアになっています。
- X.509v3 証明書：同等のデジタル ID カードを各デバイスに提供することで、IPSec で保護されたネットワークの拡張を可能にする証明書サポート。2 つの装置が通信する際、デジタル証明書を交換することで ID を証明します（これにより、各ピアが公開キーを手動で交換したり、各ピアで共有キーを手動で指定したりする必要がなくなります）。これらの証明書は CA から取得されます。X.509 は、ITU の X.500 標準の一部です。

CA の目的

認証局（CA）は、証明書要求を管理し、関係する IP セキュリティ ネットワーク デバイスへの証明書を発行します。これらのサービスは、参加デバイスのキー管理を一元化して行います。

CA は、IPSec ネットワーク デバイスの管理を簡素化します。CA は、ルータなど、複数の IPSec 対応デバイスを含むネットワークで使用できます。

公開キー暗号化によりイネーブルにされたデジタル署名は、デバイスおよび個人ユーザをデジタル認証します。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。つまり、署名は、データがユーザの秘密キーで暗号化されるときに形成されます。受信者は、送信者の公開キーを使用してメッセージを復号化することで、署名を検証します。送信側の公開キーを使用してメッセージを復号できたという事実から、そのメッセージが秘密キーの所有者つまり送信者によって作成されたことがわかります。このプロセスでは、受信者が送信者の公開キーのコピーを持っていること、およびそのキーが送信者になりすました別人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が重要です。

デジタル証明書はリンクを提供します。デジタル証明書には、名前、シリアル番号、企業、部署または IP アドレスなど、ユーザまたはデバイスを特定する情報を含んでいます。また、エンティティの公開キーのコピーも含まれています。証明書自体は、受信者が身元を証明しデジタル証明書を作成するうえで確実に信頼できるサード パーティである、認証局（CA）により署名されます。

CA の署名を検証するには、受信者が CA の公開キーを認識している必要があります。このプロセスは通常、アウトオブバンド、またはインストール時に実行される操作によって処理されます。たとえば、通常の Web ブラウザでは、デフォルトで、複数の CA の公開キーが設定されています。IPSec の基本コンポーネントであるインターネット キー交換（IKE）は、デジタル署名を使用して、セキュリティ アソシエーションを設定する前にピア デバイスをスケーラブルに認証できます。

デジタル署名がない場合は、IPSec を使用するデバイスの各ペア間で公開キーまたはシークレットを手動で交換して、通信を保護する必要があります。証明書がない場合、ネットワークに新しいデバイスが追加されるたびに、安全に通信を行う他のすべてのデバイスで設定を変更する必要があります。デジタル証明書がある場合、各デバイスは、認証局に登録されます。2 台のデバイスが通信する場合、証明書を交換し、データをデジタル署名して、お互いを認証します。ネットワークに新しいデバイスを追加する場合には、そのデバイスを CA に登録するだけでよく、他のデバイスの設定を変更する必要はありません。新しいデバイスが IPSec 接続を試行すると、証明書が自動的に交換され、デバイスを認証できます。

CA なしでの IPsec の実装

CA を使用せずに、2 つの Cisco デバイス間で IPSec サービス（暗号化など）を有効にする場合、最初に、各デバイスにもう一方のデバイスのキー（RSA 公開キーや共有キー）が存在することを確認する必要があります。つまり、次のいずれかの操作を手動で実行する必要があります。

- 各デバイスで、もう一方のデバイスの RSA 公開キーを入力します。
- 各デバイスで、両方のデバイスに使用される共有キーを指定します。

上の図では、各デバイスが他方のデバイスのキーを使用して、他方のデバイスのアイデンティティを認証します。この認証は、2 台のデバイス間で IPsec トラフィックが交換される場合には必ず実行されます。

複数の Cisco デバイスをメッシュ トポロジで配置し、すべてのデバイス間で IPSec トラフィックを交換させる場合には、最初に、すべてのデバイス間に共有キーまたは RSA 公開キーを設定する必要があります。

IPSec ネットワークに新しいデバイスを追加するごとに、新しいデバイスと既存の各デバイス間にキーを設定する必要があります。（図 34 の場合、このネットワークに 1 台の暗号化デバイスを追加するには、新たに 4 つのスイッチ間キー設定が必要になります）。

したがって、IPSec サービスを必要とするデバイスが増えるほど、キー管理は複雑になります。このアプローチでは、より大型で複雑な暗号化ネットワークには拡張できません。

CA での IPsec の実装

CA では、すべての暗号化デバイス間にキーを設定する必要はありません。代わりに、加入させる各デバイスを CA に個別に登録し、各デバイスの証明書を要求します。この設定が完了していれば、各加入デバイスは、他のすべての加入デバイスをダイナミックに認証できます。このプロセスについて、図で説明します。

ネットワークに新しい IPSec デバイスを追加する場合、新しいデバイスが CA に証明書を要求するように設定するだけでよく、既存の他のすべての IPSec デバイスとの間に複数のキー設定を行う必要はありません。

複数のルート CA での IPsec の実装

複数のルート CA がある場合、証明書をピアに発行した CA にデバイスを登録する必要はありません。その代わり、信頼できる複数の CA にデバイスを設定します。そのため、デバイスは、設定された CA（信頼できるルート）を使用して、デバイス ID で定義されている同じ CA 以外から発行された証明を、ピアが提供したかどうかを検証できます。

複数の CA を設定することにより、IKE を使用して IPSec トンネルを確立する場合に、異なるドメイン（異なる CA）に登録した 2 台以上のデバイス間で相互の ID を確認できます。

Simple Certificate Enrollment Protocol（SCEP）では、各デバイスは、CA（登録 CA）で設定されます。CA は、CA の秘密キーで署名されるデバイスに証明書を発行します。同じドメインのピアの証明書を確認するため、デバイスは、登録 CA のルート証明書でも設定されます。

異なるドメインからのピアの証明書を確認するには、そのピアのドメインの登録 CA のルート証明書をデバイスで安全に設定する必要があります。

インターネット キー交換（IKE）フェーズ I の署名の検証中、発信側は CA 証明書のリストを応答側に送信します。応答側は、リストのいずれかの CA により発行される証明書を送信する必要があります。証明書が検証されると、デバイスは、証明書に含まれる公開キーを公開キーリングに保存します。

複数のルート CA がある場合、VPN ユーザーは、1 つのドメインで信頼を確立して、それを他のドメインに簡単かつ安全に配布できます。そのため、異なるドメインで認証されるエンティティ間の必要なプライベート通信チャネルが発生します。

IPSec デバイスによる CA 証明書の使用方法

IPSec で保護されたトラフィックを 2 台の IPSec デバイス間で交換させるには、最初に相互に認証しあう必要があります。認証されていない場合、IPSec 保護が適用されません。この認証を行うには、IKE を使用します。

CA を使用しない場合、デバイスは、RSA 暗号化ナンスまたは事前共有キーを使用して、リモートデバイスに対して自身を認証します。いずれの方式でも、2 つのデバイス間でキーを事前に設定しておく必要があります。

CA を使用する場合、デバイスはリモート デバイスに証明書を送信し、何らかの公開キー暗号化を実行することによって、リモート デバイスに対して自身を認証します。各デバイスは、CA により発行されて検証された、固有の証明書を送信する必要があります。このプロセスが有効なのは、各デバイスの証明書にデバイスの公開キーがカプセル化され、各証明書が CA によって認証されることにより、すべての加入デバイスが CA を認証局として認識するからです。この機構は、RSA シグニチャを使用する IKE と呼ばれます。

デバイスは、証明書が期限切れになるまで、複数の IPSec ピアに対して、複数の IPSec セッション用に自身の証明書を継続的に送信できます。証明書が期限満了になったときは、デバイス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。

また、CA は、IPSec に参加しなくなったデバイスの証明書を失効できます。失効された証明書は、他の IPSec デバイスから有効とは見なされません。失効された証明書は、証明書失効リスト（CRL）にリストされ、各ピアは相手側ピアの証明書を受け入れる前に、このリストを確認できます。

登録局

一部の CA に、実装の一部として登録局（RA）があります。RA は本質的に CA のプロキシの役割を果たすサーバであるため、CA がオフラインのときも CA 機能は継続しています。

このマニュアルに記載されている設定タスクの一部は、CA での RA のサポートの有無によって、多少の違いがあります。

認証局の設定方法

NVRAM メモリ使用率の管理

CA 証明書が使用されるとき、デバイスは証明書と証明書失効リスト（CRL）を使用します。通常、一部の証明書とすべての CRL は、デバイスの NVRAM にローカルに保存されており、各証明書および CRL は相応な量のメモリを使用します。

通常、デバイスには次の証明書が保存されます。

- デバイスの証明書
- CA の証明書
- CA サーバから取得したルート証明書（デバイスが初期化された後、すべてのルート証明書が RAM に保存されます）
- 2 つの登録局（RA）証明書（CA が RA をサポートしている場合のみ）

CRL は通常、次の条件に従ってデバイスで保存されます。

- CA が RA をサポートしていない場合、デバイスには 1 つの CRL のみ保存されます。
- CA が RA をサポートしている場合、複数の CRL をデバイスに保存できます。

これらの証明書と CRL をローカルに保存することが、何の問題にもならない場合もあります。しかし、メモリの問題が起こる可能性もあります。特に、CA が RA をサポートし、デバイスに多数の CRL は保存しなければならない場合に起こりやすくなります。NVRAM が小さすぎてルート証明書を保存できない場合は、ルート証明書のフィンガープリントのみ保存されます。

NVRAM スペースを節約するには、証明書と CRL をローカルに保存せず、必要に応じて CA から取得するよう指定します。この代替策では、NVRAM スペースを節約できますが、パフォーマンスに多少影響が出る可能性があります。証明書と CRL をデバイスにローカル保存せず必要なときに取得するよう指定するには、クエリ モードを有効にします。

クエリ モードの有効化は、この時点ではなく後で実施することもできます。証明書と CRL がすでにデバイスに保存されている場合でも可能です。このような場合、クエリ モードを有効にすると、設定を保存した後、保存済みの証明書と CRL がデバイスから削除されます（クエリ モードを有効にする前に TFTP サイトに設定をコピーしておくと、保存されていたあらゆる証明書と CRL を TFTP サイトで保管することができます）。

クエリモードを無効にする前に、**copy system:running-config nvram:startup-config** コマンドを実行して、現在の証明書と CRL をすべて NVRAM に保存します。そうしないと、リブート時にこれらが失われることがあります。

証明書と CRL をデバイスにローカル保存せず必要なときに取得するよう指定するには、グローバル コンフィギュレーション モードで次のコマンドを使用して、クエリ モードを有効にします。

（注） クエリ モードは、CA がダウン状態にある場合、可用性に影響を及ぼす可能性があります。

手順の概要

1. **crypto ca certificate query**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**crypto ca certificate query** 例： `Device(config)# crypto ca certificate query`	クエリ モードを有効にします。これにより、証明書と CRL のローカル保存が行われなくなります。

デバイス ホスト名および IP ドメイン名の設定

デバイスのホスト名および IP ドメイン名が未設定の場合には、これを設定する必要があります。これが必要になるのは、IPSec によって使用されるキーおよび証明書にデバイスが完全修飾ドメイン名（FQDN）を割り当てており、デバイスに割り当てられたホスト名および IP ドメイン名に FQDN が基づいているためです。たとえば、「device20.example.com」という名前の証明書は、「device20」というデバイスのホスト名と「example.com」というデバイスの IP ドメイン名に基づいています。

手順の概要

1. **enable**
2. **configure terminal**
3. **hostname** *name*
4. **ip domain-name** *name*
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**hostname** *name* 例： `Device(config)# hostname device1`	デバイスのホスト名を設定します。
ステップ 4	**ip domain-name** *name* 例： `Device(config)# ip domain-name domain.com`	デバイスの IP ドメイン名を設定します。

	コマンドまたはアクション	目的
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーションを終了して、特権 EXEC モードに戻ります。

RSA キー ペアの生成

Rivest、Shamir、Adelman（RSA）キー ペアは IKE キー管理メッセージの署名および暗号化に使用されます。また、デバイスの証明書を取得する前に必要になります。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** [**usage-keys**]
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**crypto key generate rsa** [**usage-keys**] 例： `Device(config)# crypto key generate rsa usage-keys`	RSA キー ペアを生成します。 • **usage-keys** キーワードを使用して、汎用キーではなく特定目的のキーを指定します。
ステップ 4	**end** 例： `Device(config)# end`	グローバル コンフィギュレーションを終了して、特権 EXEC モードに戻ります。

認証局の宣言

デバイスが使用する 1 つの認証局（CA）を宣言する必要があります。

手順の概要

1. **enable**

2. **configure terminal**
3. **crypto ca trustpoint** *name*
4. **enrollment url** *url*
5. **enrollment command**
6. **exit**
7. **crypto pki trustpoint** *name*
8. **crl query ldap**://*url*:[*port*]
9. **enrollment** {**mode ra** | **retry count** *number* | **retry period** *minutes* | **url** *url*}
10. **enrollment** {**mode ra** | **retry count** *number* | **retry period** *minutes* | **url** *url*}
11. **revocation-check** *method1* [*method2 method3*]
12. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**crypto ca trustpoint** *name* 例： `Device(config)# crypto ca trustpoint ka`	デバイスが使用する認証局（CA）を宣言し、CA プロファイル登録コンフィギュレーション モードを開始します。
ステップ 4	**enrollment url** *url* 例： `Device(ca-profile-enroll)# enrollment url http://entrust:81`	登録要求の送信先とする CA サーバの URL を指定します。
ステップ 5	**enrollment command** 例： `Device(ca-profile-enroll)# enrollment command`	登録のため CA に送信される HTTP コマンドを指定します。
ステップ 6	**exit** 例： `Device(ca-profile-enroll)# exit`	CA プロファイル登録コンフィギュレーション モードを終了してグローバル コンフィギュレーション モードに戻ります。
ステップ 7	**crypto pki trustpoint** *name* 例： `Device(config)# crypto pki trustpoint ka`	デバイスで使用するトラストポイントを宣言し、CA トラストポイント コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 8	**crl query ldap**://*url*:[*port*] 例： `Device(ca-trustpoint)# crl query ldap://bar.cisco.com:3899`	証明書失効リスト（CRL）を照会し、ピアの証明書が失効していないことを確認します。
ステップ 9	**enrollment** {**mode ra** | **retry count** *number* | **retry period** *minutes* | **url** *url*} 例： `Device(ca-trustpoint)# enrollment retry period 2`	証明書要求を再試行するまでの登録待機時間を指定します。
ステップ 10	**enrollment** {**mode ra** | **retry count** *number* | **retry period** *minutes* | **url** *url*} 例： `Device(ca-trustpoint)# enrollment retry count 8`	以前の要求への応答が得られない場合にデバイスが証明書要求を再送信する回数を指定します。
ステップ 11	**revocation-check** *method1* [*method2 method3*] 例： `Device(ca-trustpoint)# revocation-check crl ocsp`	証明書の失効ステータスをチェックします。
ステップ 12	**end** 例： `Device(ca-trustpoint)# end`	CA トラストポイントコンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。

ルート CA（信頼できるルート）の設定

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto ca trustpoint** *name*
4. **revocation-check** *method1* [*method2 method3*]
5. **root tftp** *server-hostname filename*
6. **enrollment http-proxy** *hostname port-number*
7. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**crypto ca trustpoint** *name* 例： `Device(config)# crypto ca trustpoint ka`	デバイスで使用するトラストポイントを宣言し、CA トラストポイント コンフィギュレーション モードを開始します。
ステップ 4	**revocation-check** *method1* [*method2 method3*] 例： `Device(ca-trustpoint)# revocation-check ocsp`	証明書の失効ステータスをチェックします。
ステップ 5	**root tftp** *server-hostname filename* 例： `Device(ca-trustpoint)# root tftp server1 file1`	TFTP 経由で認証局（CA）の証明書を取得します。
ステップ 6	**enrollment http-proxy** *hostname port-number* 例： `Device(ca-trustpoint)# enrollment http-proxy host2 8080`	HTTP を使用して、プロキシ サーバ経由で認証局（CA）にアクセスします。
ステップ 7	**end** 例： `Device(ca-trustpoint)# end`	CA トラストポイント コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

CA の認証

デバイスは認証局（CA）を認証する必要があります。CA を認証するには、CA の公開キーが含まれている CA の自己署名証明書を取得します。この CA の証明書は自己署名（CA が自身の証明書に署名したもの）であるため、CA の公開キーは、この手順実行時に、CA の管理者に連絡して CA 証明書のフィンガープリントを比較することにより、手動で認証する必要があります。

CA の公開キーを取得するには次のタスクを実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto pki authenticate***name*
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**crypto pki authenticate***name* 例： `Device(config)# crypto pki authenticate myca`	CA の証明書を取得することにより CA を認証します。
ステップ 4	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

署名証明書の要求

デバイスのRSAキーペアごとに、認証局（CA）から署名証明書を取得する必要があります。汎用 RSA キーを生成した場合、デバイスは 1 組の RSA キー ペアだけを持ち、1 個の証明書だけが必要です。特定目的の RSA キーを以前に生成している場合、デバイスは 2 組の RSA キーペアを持ち、2 個の証明書が必要です。

CA から署名証明書を要求するには、次の作業を実行します。

（注） **crypto pki enroll** コマンドを発行した後、証明書を受信する前にデバイスがリブートされた場合は、コマンドを再発行して CA の管理者に連絡する必要があります。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto pki enroll** *number*
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**crypto pki enroll** *number* 例： `Device(config)# crypto pki enroll myca`	CA からデバイスの証明書を取得します。
ステップ 4	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

次のタスク

設定の保存

設定の変更を行った場合は、必ず作業結果を保存するようにしてください。

copy system:running-config nvram:startup-config コマンドを使用して、設定を保存します。このコマンドには、RSA キーをプライベート NVRAM に保存する命令が含まれています。**copy system:running-config rcp:** または copy **system:running-config tftp:** コマンドを使用すると、RSA キーは設定に保存されません 。

認証局のモニタリングと維持

証明書失効リストの要求

証明書失効リスト（CRL）の要求は、認証局（CA）が登録局（RA）をサポートしていないときのみ実施可能です。次のタスクは、CA が RA をサポートしていないときのみ適用されます。

デバイスがピアから証明書を受信すると、デバイスは CA から CRL をダウンロードします。次に、デバイスは CRL をチェックして、ピアから送信された証明書が無効になっていないことを確認します（証明書が CRL に表示されている場合、デバイスは証明書を受け付けず、ピアを認証しません）。

クエリ モードがオフの場合は、CRL の期限が切れるまで CRL を後続の証明書に再使用することができます。該当する CRL の期限が切れた後でデバイスがピアの証明書を受信すると、デバイスは新しい CRL をダウンロードします。

デバイスにある CRL は有効期限内だがそのコンテンツが古くなっていることが疑われる場合は、古い CRL と置き換える最新の CRL をすぐにダウンロードするよう要求することができます。

-

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto pki crl request** *name*
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**crypto pki crl request** *name* 例： `Device(config)# crypto pki crl request myca`	CA から新しい証明書失効リスト（CRL）をただちに取得するよう要求します。
ステップ 4	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

証明書失効リストの照会

証明書失効リスト（CRL）の照会は、信頼できるルートでデバイスを設定するときのみ実行可能です。デバイスが別のドメイン（異なる CA）のピアから証明書を受信した場合、デバイスの CA からダウンロードした CRL には、そのピアの証明書情報が含まれません。そのため、LDAP URL で設定したルートにより発行された CRL をチェックして、ピアの証明書が失効していないことを確認する必要があります。

デバイス再起動時にルート証明書の CRL を照会したい場合は、**crl query** コマンドを入力する必要があります。

LDAP URL で設定されたルートにより発行された CRL を照会するには、次の作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **crl query ldap** ://*url* : [*port*]
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**crypto pki trustpoint** *name* 例： `Device(ca-trustpoint)# crypto pki trustpoint mytp`	デバイスで使用するトラストポイントを宣言し、CA トラストポイント コンフィギュレーション モードを開始します。
ステップ 4	**crl query ldap** ://*url* : [*port*] 例： `Device(ca-trustpoint)# crl query ldap://url:[port]`	CRL を照会し、ピアの証明書が失効していないことを確認します。
ステップ 5	**end** 例： `Device(ca-trustpoint)# end`	CA トラストポイント コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

デバイスからの RSA キーの削除

特定の状況下では、デバイスから RSA キーを削除することが必要になる場合があります。たとえば、何らかの原因で RSA キーペアの信用性が失われ、使用しなくなった場合、そのキーペアを削除します。

]

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto key zeroize rsa** [*key-pair-label*]
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**crypto key zeroize rsa** [*key-pair-label*] 例： `Device(config)# crypto key zeroize rsa`	すべての Rivest、Shamir、Adelman（RSA）キーをデバイスから削除します。
ステップ 4	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

次のタスク

デバイスから RSA キーを削除した後、次の 2 つの追加作業も完了する必要があります。

- CA の管理者に、CA でデバイスの証明書を無効にするよう依頼します。このとき、**crypto pki enroll** コマンドを使用して初めてデバイスの証明書を取得した際に作成したチャレンジパスワードを、提供する必要があります。
- デバイスの設定からデバイスの証明書を手動で削除します。

ピアの公開キーの削除

特定の状況下では、デバイスの設定からピア デバイスの RSA 公開キーを削除することが必要になる場合があります。たとえば、ピアの公開キーの整合性が信頼できなくなった場合、キーを削除する必要があります。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto key pubkey-chain rsa**
4. **no named key** *key-name* [**encryption** | **signature**]
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**crypto key pubkey-chain rsa** 例： `Device(config)# crypto key pubkey-chain rsa`	他のデバイスのRSA公開キーを手動で指定できるようにするため、公開キーチェーン コンフィギュレーション モードを開始します。
ステップ 4	**no named key** *key-name* [**encryption** | **signature**] 例： `Device(config-pubkey-c)# no named-key otherpeer.example.com`	リモートピアのRSA公開キーを削除して、公開キーコンフィギュレーション モードを開始します。
ステップ 5	**end** 例： `Device(config-pubkey)# end`	公開キー コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

設定からの証明書の削除

必要に応じて、デバイスに保存された証明書を削除することができます。デバイスには、自身の証明書、CA の証明書、任意の RA 証明書が保存されています。

CA の証明書を削除するには、CA のアイデンティティ全体を削除する必要があります。これにより、CA に関連付けられたすべての証明書（ルータの証明書、CA 証明書、任意の RA 証明書）も削除されます。

手順の概要

1. **enable**
2. **show crypto pki certificates**
3. **configure terminal**
4. **crypto pki certificate chain** *name*
5. **no certificate** *certificate-serial-number*
6. **exit**
7. **no crypto pki import** *name* **certificate**
8. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**show crypto pki certificates** 例： `Device# show crypto pki certificates`	デバイスの証明書、認証局（CA）証明書、および任意の登録局（RA）証明書に関する情報を表示します。
ステップ 3	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 4	**crypto pki certificate chain** *name* 例： `Device(config)# crypto pki certificate chain myca`	証明書チェーン コンフィギュレーション モードを開始します。
ステップ 5	**no certificate** *certificate-serial-number* 例： `Device(config-cert-chain)# no certificate 0123456789ABCDEF0123456789ABCDEF`	証明書を削除します。
ステップ 6	**exit** 例： `Device(config-cert-chain)# exit`	証明書チェーン コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**no crypto pki import** *name* **certificate** 例： `Device(config)# no crypto pki import MS certificate`	証明書を手動で削除します。
ステップ 8	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

キーと証明書の表示

キーと証明書を表示するには次の作業を実行します。

手順の概要

1. **enable**
2. **show crypto key mypubkey rsa** [*keyname*]

3. **show crypto key pubkey-chain rsa**
4. **show crypto key pubkey-chain rsa** [**name** *key-name* | **address** *key-address*]
5. **show crypto pki certificates**
6. **show crypto pki trustpoints**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**show crypto key mypubkey rsa** [*keyname*] 例： `Device# show crypto key mypubkey rsa [keyname]`	デバイスで設定されている RSA 公開キーを表示します。
ステップ 3	**show crypto key pubkey-chain rsa** 例： `Device# show crypto key pubkey-chain rsa`	デバイスに保存されている、ピアの RSA 公開キーを表示します。
ステップ 4	**show crypto key pubkey-chain rsa** [**name** *key-name* | **address** *key-address*] 例： `Device# show crypto key pubkey-chain rsa address 209.165.202.129`	特定のキーのアドレスを表示します。
ステップ 5	**show crypto pki certificates** 例： `Device# show crypto pki certificates`	デバイスの証明書、認証局（CA）証明書、および任意の登録局（RA）証明書に関する情報を表示します。
ステップ 6	**show crypto pki trustpoints** 例： `Device# show crypto pki certificates`	デバイスで設定されているトラストポイントを表示します。

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。