認証局相互運用性の実装

CA の相互運用性により、デバイスと CA は通信でき、デバイスがデジタル証明書を CA から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。

（注） IPSec は将来のリリースでサポートされる予定です。

認証局相互運用性の実装

CA の相互運用性により、デバイスと CA は通信でき、デバイスがデジタル証明書を CA から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。

（注） IPSec は将来のリリースでサポートされる予定です。

認証局の実装に関する前提条件

CA 相互運用性を実装するには、次の前提条件を満たす必要があります。

- 適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。
- セキュリティ ソフトウェアのパッケージインストレーションエンベロープ（PIE）をインストールしてアクティブにする必要があります。

  オプションの PIE インストールの詳細については、『*System Management Guide*』を参照してください。

Cisco IOS XR ソフトウェア Release 7.0.1 以降では、PIE はベースイメージ自体で使用できるため、この機能をインストールする必要はありません。

- この相互運用性機能を設定する前に、ネットワークで CA を使用可能にする必要があります。CA は、Cisco Systems PKI プロトコル、Simple Certificate Enrollment Protocol（SCEP）（以前の Certificate Enrollment Protocol（CEP））をサポートする必要があります。

認証局の実装に関する制約事項

ルータのホスト名および IP ドメイン名の設定

この作業では、ルータのホスト名および IP ドメイン名を設定します。

ルータのホスト名および IP ドメイン名が未設定の場合には、これらを設定する必要があります。ホスト名および IP ドメイン名が必要なのは、ルータが完全修飾ドメイン名（FQDN）を IPSec により使用されるキーおよび証明書に割り当て、ルータ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、router20.example.com という名前の証明書は、router20 というルータのホスト名と example.com というルータの IP ドメイン名に基づいています。

手順

ステップ 1 **configure**

ステップ 2 **hostname** *name*

例：

```
RP/0/RP0/cpu 0: router(config)# hostname myhost
```

ルータのホスト名を設定します。

ステップ 3 **domain name domain-name**

例：

```
RP/0/RP0/cpu 0: router(config)# domain name mydomain.com
```

ルータの IP ドメイン名を設定します。

ステップ 4 **commit**

RSA キー ペアの生成

RSA キー ペアを生成します。

Cisco IOS XR ソフトウェア Release 7.0.1 以降では、暗号キーはルータの起動時に自動生成されます。したがって、一部のシナリオでルータにRSAホストキーペアが存在しない場合にのみ、ステップ 1 を設定する必要があります。

RSAキーペアはIKEキー交換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、ルータの証明書を取得する際に必要です。

手順

ステップ 1 **crypto key generate rsa** [**usage keys** | **general-keys**] [**keypair-label**]

例：

```
RP/0/RP0/cpu 0: router# crypto key generate rsa general-keys
```

RSA キー ペアを生成します。

- 特殊用途キーを指定するには、**usage keys** キーワードを使用します。汎用RSAキーを指定するには、**general-keys** キーワードを使用します。
- *keypair-label* 引数は、RSA キー ペアを指定する RSA キー ペア ラベルです。

ステップ 2 **crypto key zeroize rsa** [**keypair-label**]

例：

```
RP/0/RP0/cpu 0: router# crypto key zeroize rsa key1
```

（任意）ルータからすべての RSA を削除します。

- 場合によっては、すべての RSA キーをルータから削除します。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。
- 特定の RSA キー ペアを削除するには、*keypair-label* 引数を使用します。

ステップ 3 show crypto key mypubkey rsa

例：

```
RP/0/RP0/cpu 0: router# show crypto key mypubkey rsa
```

（任意）ルータの RSA 公開キーを表示します。

公開キーのルータへのインポート

公開キーをルータにインポートします。

公開キーがルータにインポートされ、ユーザが認証されます。

手順

ステップ 1 **crypto key import authentication rsa** [**usage keys** | **general-keys**] [**keypair-label**]

例：

```
RP/0/RP0/cpu 0: router# crypto key import authentication rsa general-keys
```

RSA キー ペアを生成します。

- 特殊用途キーを指定するには、**usage keys** キーワードを使用します。汎用 RSA キーを指定するには、**general-keys** キーワードを使用します。
- *keypair-label* 引数は、RSA キー ペアを指定する RSA キー ペア ラベルです。

ステップ 2 show crypto key mypubkey rsa

例：

```
RP/0/RP0/cpu 0: router# show crypto key mypubkey rsa
```

（任意）ルータの RSA 公開キーを表示します。

認証局の宣言と信頼できるポイントの設定

CA を宣言し、信頼できるポイントを設定します。

手順

ステップ 1 **configure**

ステップ 2 **crypto ca trustpoint ca-name**

例：

```
RP/0/RP0/cpu 0: router(config)# crypto ca trustpoint myca
```

CA を宣言します。

- ルータがピアに対して発行された証明書を確認できるように、選択した名前で信頼できるポイントを設定します。
- 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 3 **enrollment url CA-URL**

例：

```
RP/0/RP0/cpu 0: router(config-trustp)# enrollment url
http://ca.domain.com/certsrv/mscep/mscep.dll
```

CA の URL を指定します。

- URL には、非標準 cgi-bin スクリプトの場所が含まれている必要があります。

ステップ 4 **query url LDAP-URL**

例：

```
RP/0/RP0/cpu 0: router(config-trustp)# query url ldap://my-ldap.domain.com
```

（任意）CA システムにより LDAP プロトコルがサポートされている場合、LDAP サーバの位置を指定します。

ステップ 5 **enrollment retry period minutes**

例：

```
RP/0/RP0/cpu 0: router(config-trustp)# enrollment retry period 2
```

（任意）再試行期間を指定します。

- 証明書の要求後、ルータは CA からの証明書の受け取りを待機します。ルータが期間（再試行期間）内に証明書を受け取らない場合、ルータは、別の証明書要求を送信します。
- 範囲は 1 ～ 60 分です。デフォルトは 1 分です。

ステップ 6 **enrollment retry count number**

例：

```
RP/0/RP0/cpu 0: router(config-trustp)# enrollment retry count 10
```

（任意）失敗した証明書要求送信を続行する回数を指定します。

- 範囲は 1 ～ 100 です。

ステップ 7 **rsakeypair keypair-label**

例：

```
RP/0/RP0/cpu 0: router(config-trustp)# rsakeypair mykey
```

（任意）このトラストポイントに **crypto key generate rsa** コマンドを使用して生成した名前付き RSA キー ペアを指定します。

- このキーペアを設定しない場合、トラストポイントは現在の設定のデフォルトの RSA キーを使用します。

ステップ 8 **commit**

CA の認証

ここでは、ルータへの CA を認証します。

ルータは 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て、CA を認証する必要があります。CA の証明書は自己署名（CA が自身の証明書に署名する）であるため、CA の公開キーは、CA 管理者に連絡し、CA 証明書のフィンガープリントを比較して手動で認証します。

手順

ステップ 1 **crypto ca authenticate ca-name**

例：

```
RP/0/RP0/cpu 0: router# crypto ca authenticate myca
```

CA の公開キーを含む CA 証明書を取得することで、ルータに対して CA を認証します。

ステップ 2 show crypto ca certificates

例：

```
RP/0/RP0/cpu 0: router# show crypto ca certificates
```

（任意）CA 証明書に関する情報を表示します。

自身の証明書の要求

CA からの証明書を要求します。

ルータの RSA キー ペアごとに、CA からの署名付き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、ルータは 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。前に特別な用途の RSA キーを生成した場合、ルータは 2 組の RSA キー ペアを持ち、2 個の証明書が必要です。

手順

ステップ 1 **crypto ca enroll ca-name**

例：

```
RP/0/RP0/cpu 0: router# crypto ca enroll myca
```

すべての RSA キー ペアの証明書を要求します。

- このコマンドでは、ルータは存在する RSA キー ペアと同数の証明書を要求するため、特定目的の RSA キー ペアがある場合にも、このコマンドは 1 回しか実行する必要はありません。
- このコマンドでは、設定に保存されないチャレンジ パスワードを作成する必要があります。証明書を失効させる必要が生じた場合、このパスワードが要求されるので、このパスワードを覚えておく必要があります。
- 証明書はすぐに発行できます。または、登録リトライ時間に達し、タイムアウトが発生するまで、ルータが証明書要求を毎分送信します。タイムアウトが発生した場合、システム管理者に要求承認を依頼して、このコマンドを再入力します。

ステップ 2 **show crypto ca certificates**

例：

```
RP/0/RP0/cpu 0: router# show crypto ca certificates
```

（任意）CA 証明書に関する情報を表示します。

カットアンドペーストによる証明書登録の設定

ルータが使用するトラストポイント認証局（CA）を宣言して、このトラストポイント CA をカットアンドペーストによる手動登録に設定します。

手順

ステップ 1 **configure**

ステップ 2 **crypto ca trustpoint** *ca-name*

例：

```
 RP/0/RP0/cpu 0: router(config)# crypto ca trustpoint myca
RP/0//CPU0:router(config-trustp)#
```

ルータが使用する CA を宣言し、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。

- *ca-name* 引数を使用して、CA の名前を指定します。

ステップ 3 enrollment terminal

例：

```
 RP/0/RP0/cpu 0: router(config-trustp)# enrollment terminal
```

カットアンドペーストによる手動での証明書登録を指定します。

ステップ 4 **commit**

ステップ 5 **crypto ca authenticate** *ca-name*

例：

```
RP/0/RP0/cpu 0: router# crypto ca authenticate myca
```

CA の証明書を取得することにより、CA を認証します。

- *ca-name* 引数を使用して、CA の名前を指定します。ステップ 2 で入力したのと同じ名前を使用します。

ステップ 6 **crypto ca enroll** *ca-name*

例：

```
RP/0/RP0/cpu 0: router# crypto ca enroll myca
```

CA からルータの証明書を取得します。

- *ca-name* 引数を使用して、CA の名前を指定します。ステップ 2 で入力したのと同じ名前を使用します。

ステップ 7 **crypto ca import** ca- *name* **certificate**

例：

```
RP/0/RP0/cpu 0: router# crypto ca import myca certificate
```

端末で証明書を手動でインポートします。

- *ca-name* 引数を使用して、CA の名前を指定します。ステップ 2 で入力したのと同じ名前を使用します。

（注） 用途キー（署名キーおよび暗号キー）を使用する場合は、**crypto ca import** コマンドを 2 回入力する必要があります。このコマンドを最初に入力した場合は、認証の 1 つがルータにペーストされます。2 回目に入力した場合は、他の認証がルータにペーストされます（どの証明書が最初にペーストされるかは重要ではありません）。

ステップ 8 show crypto ca certificates

例：

```
RP/0/RP0/cpu 0: router# show crypto ca certificates
```

証明書と CA 証明書に関する情報を表示します。

次に、CA 相互運用性を設定する例を示します。

さまざまなコマンドを説明するコメントが設定に含まれます。

```
configure
hostname myrouter
domain name mydomain.com
end
```

```
Uncommitted changes found, commit them? [yes]:yes

crypto key generate rsa mykey

The name for the keys will be:mykey
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose
 Keypair
Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus [1024]:
Generating RSA keys ...
Done w/ crypto generate keypair
[OK]

show crypto key mypubkey rsa

Key label:mykey
Type     :RSA General purpose
Size     :1024
Created  :17:33:23 UTC Thu Sep 18 2003
Data     :
 30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00CB8D86
 BF6707AA FD7E4F08 A1F70080 B9E6016B 8128004C B477817B BCF35106 BC60B06E
 07A417FD 7979D262 B35465A6 1D3B70D1 36ACAFBD 7F91D5A0 CFB0EE91 B9D52C69
 7CAF89ED F66A6A58 89EEF776 A03916CB 3663FB17 B7DBEBF8 1C54AF7F 293F3004
 C15B08A8 C6965F1E 289DD724 BD40AF59 E90E44D5 7D590000 5C4BEA9D B5020301
 0001

! The following commands declare a CA and configure a trusted point.

configure
crypto ca trustpoint myca
enrollment url http://xyz-ultra5
enrollment retry count 25
enrollment retry period 2
rsakeypair mykey
end

Uncommitted changes found, commit them? [yes]:yes

! The following command authenticates the CA to your router.

crypto ca authenticate myca

Serial Number  :01
Subject Name   :
cn=Root coax-u10 Certificate Manager,ou=HFR,o=Cisco Systems,l=San Jose,st=CA,c=US
Issued By      :
cn=Root coax-u10 Certificate Manager,ou=HFR,o=Cisco Systems,l=San Jose,st=CA,c=US
Validity Start :07:00:00 UTC Tue Aug 19 2003
Validity End   :07:00:00 UTC Wed Aug 19 2020
Fingerprint:58 71 FB 94 55 65 D4 64 38 91 2B 00 61 E9 F8 05
Do you accept this certificate?? [yes/no]:yes

! The following command requests certificates for all of your RSA key pairs.

crypto ca enroll myca

% 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:
```

```
Re-enter Password:
    Fingerprint: 17D8B38D ED2BDF2E DF8ADBF7 A7DBE35A

! The following command displays information about your certificate and the CA certificate.

show crypto ca certificates

Trustpoint       :myca
========================================================
CA certificate
  Serial Number  :01
  Subject Name   :
       cn=Root coax-u10 Certificate Manager,ou=HFR,o=Cisco Systems,l=San Jose,st=CA,c=US

  Issued By      :
       cn=Root coax-u10 Certificate Manager,ou=HFR,o=Cisco Systems,l=San Jose,st=CA,c=US

  Validity Start :07:00:00 UTC Tue Aug 19 2003
  Validity End   :07:00:00 UTC Wed Aug 19 2020
Router certificate
  Key usage      :General Purpose
  Status         :Available
  Serial Number  :6E
  Subject Name   :
        unstructuredName=myrouter.mydomain.com,o=Cisco Systems
  Issued By      :
       cn=Root coax-u10 Certificate Manager,ou=HFR,o=Cisco Systems,l=San Jose,st=CA,c=US

  Validity Start :21:43:14 UTC Mon Sep 22 2003
  Validity End   :21:43:14 UTC Mon Sep 29 2003
  CRL Distribution Point
        ldap://coax-u10.cisco.com/CN=Root coax-u10 Certificate Manager,O=Cisco Systems
```

認証局のトラスト プール管理

トラスト プール機能を使用すると、認証局（CA）と呼ばれる一般的に認識された信頼できるエージェントを使用して、デバイス間で発生する HTTPS などのセッションを認証できます。この機能はデフォルトでソフトウェアで有効になっており、セッションのセキュリティ保護のためにブラウザが提供するサービスと同じ方法で、既知の CA の証明書のプールのプロビジョニング、保管、管理を行うスキーマを作成できます。トラストプールと呼ばれる特別な信頼できるポイントが指定され、シスコから、および場合によっては他のベンダーからの複数の既知の CA 証明書が含まれています。トラスト プールは、組み込みの CA 証明書とダウンロードされた CA 証明書の両方で構成されます。

「認証局相互運用性の実装」では、認証局と信頼できるポイントの詳細について説明します。

トラスト プールでの CA 証明書のバンドル

ルータは、asr9k-k9sec PIE にパッケージ化された組み込みの CA 証明書バンドルを使用します。このバンドルは、シスコによって自動的に更新される、CA トラスト プールと呼ばれる特別な証明書ストアに含まれています。このトラストプールは、シスコおよび他のベンダーにも知られています。CA 証明書バンドルは次の形式で提供されます。

- 公開キー暗号メッセージ構文規格 7（pkcs7）内に含まれる識別符号化規則（DER）バイナリ形式の特権管理インフラストラクチャ（PMI）証明書。

- PEMヘッダー付きプライバシー強化メール（PEM）形式の連結型X.509証明書を含むファイル。

CA トラストプールの更新

次の条件が発生した場合は、CA トラストプールを更新する必要があります。

- トラストプールの証明書が期限切れまたは再発行されている。
- 公開された CA 証明書のバンドルに、特定のアプリケーションで必要な追加の信頼できる証明書が含まれている。
- 設定が破損している。

CA トラストプールは単一のエンティティと見なされます。したがって、実行する更新によってトラストプール全体が置き換えられます。

（注） トラストプールに組み込まれた証明書は物理的に置き換えることができません。ただし、組み込まれた証明書の X.509 所有者名属性が CA 証明書バンドル内の証明書と一致する場合、組み込まれた証明書は無効と表示されます。

以下は、トラストプール内の証明書を更新するために使用できる方法です。

- **自動更新**：最も早い有効期限を持つCA証明書と一致するトラストプールにタイマーが確立されます。タイマーが作動しても、バンドルのロケーションが設定されておらず、明示的に無効になっていない場合、syslog 警告が適切な間隔で発行され、このトラストプールポリシーオプションが設定されていないことが管理者に警告されます。トラストプールの自動更新では設定済みURLを使用します。CA トラストプールが失効すると、ポリシーが読み込まれ、バンドルがロードされ、PKI トラストプールが置き換えられます。CA トラストプールの自動更新の開始時に問題が発生した場合は、ダウンロードが成功するまで、次のスケジュールで更新が開始されます。20 日、15 日、10 日、5 日、4 日、3 日、2 日、1 日、最後に 1 時間ごとです。
- **手動更新**：「トラスト プール内の証明書の手動更新 （11 ページ）」に詳細を示します。

トラスト プール内の証明書の手動更新

CA トラスト プール機能はデフォルトで有効で、トラスト プールに組み込まれた CA 証明書バンドルを使用し、シスコから自動更新を受信します。トラストプール内の証明書が最新のものではない、破損している、または特定の証明書を更新する必要がある場合は、次の作業を実行して証明書を手動で更新します。

手順

	コマンドまたはアクション	目的
ステップ 1	**crypto ca trustpool import url clean** 例： `RP/0/RP0/CPU0:IMC0#crypto ca trustpool import url clean`	（任意）ダウンロードしたすべてのCA証明書を手動で削除します。このコマンドはEXECモードで実行されます。
ステップ 2	**crypto ca trustpool import url** `url` 例： `RP/0/RP0/CPU0:IMC0#crypto ca trustpool import url http://www.cisco.com/security/pki/trs/ios.p7b`	CAトラストプール証明書バンドルのダウンロード元となるURLを指定します。CA証明書バンドルをCAトラストプールに手動でインポート（ダウンロード）したり、既存のCA証明書バンドルを交換したりします。

ステップ 3	**show crypto ca trustpool policy**
	例：
	`RP/0/RP0/CPU0:IMC0#show crypto ca trustpool`
	
	`Trustpool: Built-In`
	`========`
	`CA certificate`
	`  Serial Number  : 5F:F8:7B:28:2B:54:DC:8D:42:A3:15:B5:68:C9:AD:FF`
	
	`  Subject:`
	`  CN=Cisco Root CA 2048,O=Cisco Systems`
	`Issued By  :`
	`  CN=Cisco Root CA 2048,O=Cisco Systems`
	`  Validity Start : 20:17:12 UTC Fri May 14 2004`
	`  Validity End  : 20:25:42 UTC Mon May 14 2029`
	`  SHA1 Fingerprint:`
	
	`DE990CED99E0431F60EDC3937E7CD5BF0ED9E5FA`
	
	`Trustpool: Built-In`
	`========`
	`CA certificate`
	`  Serial Number  : 2E:D2:0E:73:47:D3:33:83:4B:4F:DD:0D:D7:B6:96:7E`
	`Subject:`
	`  CN=Cisco Root CA M1,O=Cisco`
	`  Issued By  :`
	`  CN=Cisco Root CA M1,O=Cisco`
	`  Validity Start : 20:50:24 UTC Tue Nov 18 2008`
	`  Validity End  : 21:59:46 UTC Fri Nov 18 2033`
	`  SHA1 Fingerprint:`
	
	`45AD6BB499011BB4E84E84316A81C27D89EE5CE7`
		冗長形式でルータのCAトラストプール証明書を表示します。

オプションのトラストプール ポリシー パラメータの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure**	

ステップ 2	**crypto ca trustpool policy**
	例：
	`RP/0/RP0/CPU0:IMC0(config)#crypto ca trustpool policy`
	`RP/0/RSP0/CPU0:IMC0(config-trustpool)#`
		CA トラストプール ポリシー パラメータを設定するコマンドにアクセスできる、ca-trustpool 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	**cabundle url** *URL* 例： `RP/0/RP0/CPU0:IMC0(config-trustpool)#cabundle url http://www.cisco.com/security/pki/crl/crca2048.crl`	CA トラストプール証明書バンドルのダウンロード元となる URL を指定します。
ステップ 4	**crl optional** 例： `RP/0/RP0/CPU0:IMC0(config-trustpool)#crl optional`	トラストプール ポリシー使用時の失効確認を無効にします。デフォルトでは、ルータは証明書失効リスト（CRL）を照会することにより、証明書の失効ステータスのチェックを強制します。
ステップ 5	**description** *LINE* 例： `RP/0/RP0/CPU0:IMC0(config-trustpool)#description Trustpool for Test.`	

トラスト プールとトラスト ポイントの両方に表示される CA 証明書の処理

トラスト プールとトラスト ポイントの両方に CA が格納されている場合があります。たとえば、トラスト ポイントで CA を使用し、CA バンドルが同じ CA 内で後からダウンロードされたりします。このシナリオでは、トラストプール機能がルータに実装されても、現在の動作が変更されないようにするため、トラスト ポイント内の CA とそのポリシーは、トラスト プールまたはトラスト プール ポリシー内の CA より前に検討されます。

このポリシーは、セキュリティ アプライアンスが CA 証明書と CA によって発行されたユーザ証明書の認証ポリシーをどのように取得するかを示します。

PKI 証明書の期限の通知

この項では、公開キーインフラストラクチャ（PKI）の証明書が有効期限に近づいている場合の SNMP トラップと syslog メッセージを使用した通知メカニズムについて説明します。

PKI アラート通知について

Cisco IOS 認証局（CA）サーバを使用すると、証明書が失効する前にその証明書が自動的に登録されます。これにより、認証時にアプリケーションの証明書が使用できるようになります。ただし、ネットワークの停止、クロック更新の問題、およびCAの過負荷は、証明書の更新を妨げる可能性があります。それにより、認証に有効な証明書を使用できないため、サブシステムがオフラインになります。この機能は、証明書の失効が近づくと、CAクライアントがsyslogサーバに通知を送信するためのメカニズムを提供します。

通知は次の間隔で送信されます。

- 最初の通知：この通知は証明書が失効する 60 日前に送信されます。
- 通知の繰り返し：最初の通知の後、証明書が失効する1週間前まで後続の通知が毎週送信されます。最後の週には、証明書の失効日まで通知が毎日送信されます。

証明書の有効期限が 1 週間以上ある場合、通知は [warning] モードで送信されます。証明書の有効期限が1週間未満の場合、通知はアラートモードで送信されます。通知には次の情報が含まれます。

- 証明書シリアル番号
- 証明書の発行元名
- トラストポイント名
- 証明書タイプ
- 証明書が失効するまでの残り日数
- 証明書の件名

アラート通知は syslog サーバまたは Simple Network Management Protocol（SNMP）のトラップを介して送信されます。トラストポイントが自動登録で設定され、対応するシャドウまたはロールオーバー証明書が存在する場合は、通知が停止します。証明書のシャドウまたはロールオーバーの開始時間は、証明書の終了時刻と同じか、またはそれよりも前になります。

この機能は無効にできず、設定作業を追加する必要はありません。

次に、デバイスに表示される syslog メッセージを示します。

```
%SECURITY-CEPKI-1-CERT_EXPIRING_ALERT : Certificate expiring WITHIN A WEEK.
Trustpoint Name= check, Certificate Type= ID, Serial Number= 02:EC,
Issuer Name= CN=cacert,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN, Subject name= CN=cisco.com,
Time Left= 1 days, 23 hours, 59 minutes, 41 seconds
```

PKI クレデンシャル失効アラートの制約事項

アラートは、次の証明書には送信されません。

- 永続的または一時的な自己署名証明書
- セキュアな固有デバイス識別子（SUDI）証明書

- トラストプールに属する証明書。トラストプールには独自の失効アラートメカニズムがあります
- トラストポイントのクローン

PKI トラップ

PKI トラップでは、ネットワーク内のデバイスの証明書情報を取得するため、PKI 展開の監視と運用が簡単になります。ルート デバイスは、デバイスに設定されたしきい値に基づいて、ネットワーク管理システム（NMS）に SNMP トラップを定期的に送信します。トラップは次のシナリオで送信されます。

- 新しい証明書がインストールされる場合。SNMP トラップ（新しい証明書通知）は、証明書のシリアル番号、証明書の発行者名、証明書の所有者名、トラストポイント名、証明書タイプ、証明書の開始日と終了日などの情報を含む SNMP サーバに送信されます。
- 証明書が失効間近の場合：SNMP トラップ（証明書失効通知）は、証明書の終了日の 60 日から 1 週間前まで SNMP サーバに定期的に送信されます。証明書が失効する週には、トラップが毎日送信されます。トラップには、証明書のシリアル番号、証明書の発行者名、トラストポイント名、証明書タイプ、証明書の寿命などの証明書情報が含まれます。

PKI トラップを有効にするには、**snmp-server traps pki** コマンドを使用します。SNMP が設定されている場合、SNMP トラップは同じ PKI 有効期限タイマーで設定されます。

実行コンフィギュレーション

次に、sanitized キーワードを指定して実行コンフィギュレーションのサニタイズバージョンを表示する、show running-config コマンドからの出力例を示します。

```
Router# show running-config
Tue Jan 21 12:51:09.861 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Tue Jan 21 12:51:04 2020 by UNKNOWN
!
interface GigabitEthernet0/0/0/0
shutdown
!
interface GigabitEthernet0/0/0/1
shutdown
!
interface GigabitEthernet0/0/0/2
shutdown
!
interface GigabitEthernet0/0/0/3
shutdown
!
interface GigabitEthernet0/0/0/4
shutdown
!
snmp-server traps pki
end
```

シャドウまたは証明書のロールオーバー開始時間が証明書の終了時間よりも遅い場合、シャドウ証明書が有効でないことを示すトラップが送信されます。ただし、同じトラストポイントで利用可能なシャドウ証明書とシャドウ証明書が有効な場合には、トラップは送信されません。

認証局の実装について

認証局相互運用性のサポートされている標準

シスコでは次の標準をサポートしています。

- 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 アソシエーションおよび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 ピアの認証を提供し、IPSec キーを交渉し、IPSec セキュリティ アソシエーション（SA）を交渉します。
- 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 Adelman 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。
- SSL：Secure Socket Layer プロトコル。
- 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 台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、各ピアで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。

認証局

CA の目的

CA は、証明書要求を管理し、参加する IPSec ネットワーク デバイスへの証明書の発行します。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。

Public Key Cryptography 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、シグニチャは、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。受信側は、送信側の公開キーを使用してメッセージを復号化することで、シグニチャを検証します。メッセージは、送信側の公開キーを使用して復号化できるため、秘密キーの所有者、つまり送信者がメッセージを作成することになります。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て、これ

が本当に送信者のものであり、送信者を騙る他人のものではないことを高い確実性を持って知っていることを基盤としています。

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んでいます。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、CA により署名されます。

CA のシグニチャを検証するには、受信者は、CA の公開キーを認識している必要があります。通常、このプロセスは、アウトオブバンドで、またはインストールで行われる操作により処理されます。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。IKE は、IPSec の必須要素で、デジタル証明書を使用して、SA を設定する前にピア デバイスの拡張性を認証します。

デジタル シグニチャがない場合、ユーザは、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、CA 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。新しいデバイスがネットワークに追加されると、ユーザは、そのデバイスを CA に登録します。他のデバイスでは変更の必要はありません。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 登録局