キーチェーン管理の実装

このモジュールでは、キーチェーン管理の実装方法について説明します。キーチェーン管理は、相互に信頼を確立する前にキーなどの秘密を交換するすべてのエンティティに共有秘密を設定する認証の一般的な方式です。ピアとの通信中に、Cisco IOS XR ソフトウェアのルーティングプロトコルおよびネットワーク管理アプリケーションは、多くの場合、セキュリティを強化するために認証を使用します。

- キーチェーン管理の実装 （1 ページ）

キーチェーン管理の実装

このモジュールでは、キーチェーン管理の実装方法について説明します。キーチェーン管理は、相互に信頼を確立する前にキーなどの秘密を交換するすべてのエンティティに共有秘密を設定する認証の一般的な方式です。ピアとの通信中に、Cisco IOS XR ソフトウェアのルーティングプロトコルおよびネットワーク管理アプリケーションは、多くの場合、セキュリティを強化するために認証を使用します。

キーチェーン管理の実装に関する制約事項

システムクロックを変更すると、現在のコンフィギュレーションのキーの有効性に影響を与えることに注意する必要があります。

キーチェーンの設定

この作業では、キーチェーンの名前を設定します。

キーチェーンの名前を作成または変更できます。

手順

ステップ 1 **configure**

ステップ 2 **key chain** *key-chain-name*

例：

```
RP/0/RP0/cpu 0: router(config)# key chain isis-keys
RP/0/RP0/cpu 0: router(config-isis-keys)#
```

キーチェーンの名前を作成します。

（注） キーの ID を設定せずにキーチェーン名のみを設定しても、操作は無効と見なされます。設定を終了しても、キー ID と 1 つ以上の モードの属性または keychain-key コンフィギュレーションモードの属性（ライフタイムやキー文字列など）を設定するまでは、変更のコミットは要求されません。

ステップ 3 **commit**

ステップ 4 **show key chain** *key-chain-name*

例：

```
RP/0/RP0/cpu 0: router# show key chain isis-keys
```

（任意）キーチェーン名を表示します。

（注） *key-chain-name* 引数の指定は任意です。*key-chain-name* 引数で名前を指定しない場合は、すべてのキーチェーンが表示されます。

例

次に、キーチェーン管理を設定する例を示します。

```
configure
key chain isis-keys
accept-tolerance infinite
key 8
key-string mykey91abcd
cryptographic-algorithm MD5
send-lifetime 1:00:00 june 29 2006 infinite
accept-lifetime 1:00:00 june 29 2006 infinite
end

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

show key chain isis-keys

Key-chain: isis-keys/ -

accept-tolerance -- infinite
Key 8 -- text "1104000E120B520005282820"
  cryptographic-algorithm -- MD5
  Send lifetime:   01:00:00, 29 Jun 2006 - Always valid  [Valid now]
  Accept lifetime: 01:00:00, 29 Jun 2006 - Always valid [Valid now]
```

キーを受け付ける許容値の設定

このタスクでは、ルーティングおよび管理プロトコルなどのアプリケーションのヒットレスキー ロール オーバーを容易にするために、キーチェーンのキーを受け付ける許容値を設定します。

手順

ステップ 1 **configure**

ステップ 2 **key chain** *key-chain-name*

例：

```
RP/0//CPU0:router(config)# key chain isis-keys
```

キーチェーンの名前を作成します。

ステップ 3 **accept-tolerance** *value* [**infinite**]

例：

```
RP/0//CPU0:router(config-isis-keys)# accept-tolerance infinite
```

キーチェーンのキーを受け入れる際の許容値を設定します。

- 許容値を秒単位で設定するには、*value* 引数を使用します。範囲は、1 ～ 8640000 です。
- 許容範囲が無限であることを指定するには、**infinite** キーワードを使用します。

ステップ 4 **commit**

キーチェーンのキー ID の設定

この作業では、キーチェーンのキー ID を設定します。

キーチェーンのキーを作成または変更できます。

手順

ステップ 1 **configure**

ステップ 2 **key chain** *key-chain-name*

例：

```
RP/0//CPU0:router(config)# key chain isis-keys
```

キーチェーンの名前を作成します。

ステップ 3 **key** *key-id*

例：

```
RP/0//CPU0:router(config-isis-keys)# key 8
```

キーチェーンのキーを作成します。キー ID 番号は 10 進数から 16 進数に変換され、コマンドモード サブプロンプトが作成されます。

- *key-id* 引数は 48 ビット整数型として使用します。

ステップ 4 **commit**

キー文字列のテキストの設定

この作業では、キー文字列のテキストを設定します。

手順

ステップ 1 **configure**

ステップ 2 **key chain** *key-chain-name*

例：

```
RP/0//CPU0:router(config)# key chain isis-keys
```

キーチェーンの名前を作成します。

ステップ 3 **key** *key-id*

例：

```
RP/0//CPU0:router(config-isis-keys)# key 8
RP/0//CPU0:router(config-isis-keys-0x8)#
```

キーチェーンのキーを作成します。

ステップ 4 **key-string** [**clear** | **password**] *key-string-text*

例：

```
RP/0//CPU0:router(config-isis-keys-0x8)# key-string password 8
```

キーのテキスト文字列を指定します。

- クリア テキスト形式でキー文字列を指定するには **clear** キーワードを使用します。暗号化形式でキーを指定するには **password** キーワードを使用します。

ステップ 5 **commit**

有効なキーの判断

このタスクでは、リモート ピアを認証するローカル アプリケーションごとに、有効なキーを判断します。

手順

ステップ 1 **configure**

ステップ 2 **key chain** *key-chain-name*

例：

```
RP/0/RP0/cpu 0: router(config)# key chain isis-keys
```

キーチェーンの名前を作成します。

ステップ 3 **key** *key-id*

例：

```
RP/0/RP0/cpu 0: router(config-isis-keys)# key 8
RP/0/RP0/cpu 0: router(config-isis-keys-0x8)#
```

キーチェーンのキーを作成します。

ステップ 4 **accept-lifetime** *start-time* [**duration** *duration-value* | **infinite** | *end-time*]

例：

```
RP/0/RP0/cpu 0: router(config-isis-keys)# key 8
RP/0/(config-isis-keys-0x8)# accept-lifetime 1:00:00 october 24 2005 infinite
```

（任意）時間の観点から、キーのライフタイムの有効性を指定します。

ステップ 5 **commit**

アウトバウンド アプリケーション トラフィックの認証ダイジェストを生成するキーの設定

アウトバウンド アプリケーション トラフィックの認証ダイジェストを生成するためのキーを設定します。

手順

ステップ 1 **configure**

ステップ 2 **key chain** *key-chain-name*

例：

```
RP/0/RP0/cpu 0: router(config)# key chain isis-keys
```

キーチェーンの名前を作成します。

ステップ 3 **key** *key-id*

例：

```
RP/0/RP0/cpu 0: router(config-isis-keys)# key 8
RP/0/RP0/cpu 0: router(config-isis-keys-0x8)#
```

キーチェーンのキーを作成します。

ステップ 4 **send-lifetime** *start-time* [**duration** *duration-value* | **infinite** | *end-time*]

例：

```
RP/0/RP0/cpu 0: router(config-isis-keys)#key 8
RP/0/(config-isis-keys-0x8)# send-lifetime 1:00:00 october 24 2005 infinite
```

（任意）キーチェーンの認証キーが有効に送信される設定期間を指定します。時間の観点から、キーのライフタイムの有効性を指定できます。

また、start-time の値と次のいずれかの値を指定できます。

- **duration** キーワード（秒）
- **infinite** キーワード
- *end-time* 引数

キーのライフタイムを設定する場合は、ネットワーク タイム プロトコル（NTP）またはその他の同期方式を推奨します。

ステップ 5 **commit**

暗号化アルゴリズムの設定

暗号化アルゴリズムの選択を受け入れるためのキーチェーンを設定できるようにします。

手順

ステップ 1 **configure**

ステップ 2 **key chain** *key-chain-name*

例：

```
RP/0/RP0/cpu 0: router(config)# key chain isis-keys
RP/0/RP0/cpu 0: router(config-isis-keys)#
```

キーチェーンの名前を作成します。

ステップ 3 **key** *key-id*

例：

```
RP/0/RP0/cpu 0: router(config-isis-keys)# key 8
RP/0/RP0/cpu 0: router(config-isis-keys-0x8)#
```

キーチェーンのキーを作成します。

ステップ 4 **cryptographic-algorithm** [**HMAC-MD5** | **HMAC-SHA1-12** | **HMAC-SHA1-20** | **MD5** | **SHA-1** | **AES-128-CMAC-96** | **HMAC-SHA-256** | **HMAC-SHA1-96**]

例：

```
RP/0/RP0/cpu 0: router(config-isis-keys-0x8)# cryptographic-algorithm MD5
```

暗号化アルゴリズムを選択します。次のアルゴリズムから選択できます。

- HMAC-MD5
- HMAC-SHA1-12
- HMAC-SHA1-20
- MD5
- SHA-1
- HMAC-SHA-256
- HMAC-SHA1-96
- AES-128-CMAC-96

各ルーティングプロトコルは、次のように異なる暗号化アルゴリズムのセットをサポートします。

- Border Gateway Protocol（BGP）は、HMAC-MD5 と HMAC-SHA1-12 だけをサポートします。
- Intermediate System-to-Intermediate System（IS-IS）は、HMAC-MD5、SHA-1、MD5、AES-128-CMAC-96、HMAC-SHA-256、HMAC-SHA1-12、HMAC-SHA1-20、および HMAC-SHA1-96 をサポートします。
- Open Shortest Path First（OSPF）は、MD5、HMAC-MD5、HMAC-SHA-256、HMAC-SHA1-12、HMAC-SHA1-20、および HMAC-SHA1-96 をサポートします。

ステップ 5 **commit**

キーのライフタイム

セキュリティ方式としてキーを使用する場合は、キーのライフタイムを指定して、期限が切れた際には定期的にキーを変更する必要があります。安定性を維持するには、各パーティがアプリケーションのキーを複数保存して同時に使用できるようにする必要があります。キーチェーンは、同じピア、ピアのグループ、またはその両方を認証するために一括管理されている一連のキーです。

キーチェーン管理では、一連のキーをキーチェーンの下にまとめてグループ化し、キーチェーン内の各キーをライフタイムに関連付けます。

（注） ライフタイムが設定されていないキーはすべて無効と見なされるため、キーは設定中に拒否されます。

キーのライフタイムは、次のオプションによって定義されます。

- Start-time：絶対時間を指定します。
- End-time：開始時間に対応する絶対時間を指定するか、無期限を指定します。

キーチェーン内のそれぞれのキーの定義では、キーが有効な期間（ライフタイムなど）を指定する必要があります。指定したキーのライフタイム期間中は、この有効なキーとともにルーティング更新パケットが送信されます。キーが有効ではない期間はキーを使用できません。このため、指定したキーチェーンでは、キーの有効期間を重複させて、有効なキーの不在期間をなくすことを推奨します。有効なキーの不在期間が発生した場合、ネイバー認証は行われず、ルーティング更新は失敗します。

複数のキーチェーンを指定できます。