



Cisco NCS 560 シリーズルータ（IOS XR リリース 6.6.x）システムセキュリティコンフィギュレーションガイド

初版：2019年5月30日

シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先：シスコ コンタクトセンター

0120-092-255（フリーコール、携帯・PHS含む）

電話受付時間：平日 10:00～12:00、13:00～17:00

<http://www.cisco.com/jp/go/contactcenter/>

【注意】 シスコ製品をご使用になる前に、安全上の注意（www.cisco.com/jp/go/safety_warning/）をご確認ください。本書は、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。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com go trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2019 Cisco Systems, Inc. All rights reserved.



目次

第 1 章

AAA サービスの設定 1

AAA サービスの設定 1

AAA サービスの設定に関する前提条件 2

AAA サービスの設定に関する制約事項 2

タスク グループの設定 2

ユーザ グループの設定 4

ユーザの設定 5

RADIUS サーバ通信用のルータの設定 6

RADIUS Dead サーバ検出の設定 10

TACACS+ サーバの設定 11

RADIUS サーバ グループの設定 13

TACACS+ サーバ グループの設定 15

一連の認証方式の作成 16

一連の許可方式の作成 18

一連のアカウントिंग方式の作成 20

中間アカウントング レコードの生成 21

方式リストの適用 22

アカウントング サービスのイネーブル化 23

ログイン パラメータの設定 24

タスク マップ 25

AAA サービスに関する情報 27

第 2 章

認証局相互運用性の実装 41

認証局相互運用性の実装 41

認証局の実装に関する前提条件	41
認証局の実装に関する制約事項	42
ルータのホスト名および IP ドメイン名の設定	42
RSA キー ペアの生成	42
公開キーのルータへのインポート	43
認証局の宣言と信頼できるポイントの設定	44
CA の認証	45
自身の証明書の要求	46
カットアンドペーストによる証明書登録の設定	47
認証局のトラスト プール管理	50
トラスト プールでの CA 証明書のバンドル	50
CA トラストプールの更新	50
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	52
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	53
認証局の実装について	53
認証局相互運用性のサポートされている標準	53
認証局	54

第 3 章**キーチェーン管理の実装 57**

キーチェーン管理の実装	57
キーチェーン管理の実装に関する制約事項	57
キーチェーンの設定	57
キーを受け付ける許容値の設定	59
キーチェーンのキー ID の設定	59
キー文字列のテキストの設定	60
有効なキーの判断	61
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	61
暗号化アルゴリズムの設定	62
キーのライフタイム	64

第 4 章	URPF の実装 65
	URPF について 65
	URPF ルーズ モードの設定 65

第 5 章	管理プレーン保護の実装 69
	管理プレーン保護の利点 69
	管理プレーン保護の実装に関する制約事項 70
	インバンドインターフェイスの管理プレーン保護のデバイスの設定 70
	アウトオブバンドインターフェイスの管理プレーン保護のデバイスの設定 73
	管理プレーン保護の実装について 77
	インターフェイス上のピア フィルタリング 77
	コントロールプレーン保護 77
	管理プレーン 77

第 6 章	サードパーティ製アプリケーションのためのトラフィック保護 79
	gRPC プロトコル 79
	サードパーティ製アプリケーションのためのトラフィック保護の制限事項 80
	gRPC を介したサードパーティ製アプリケーションのためのトラフィック保護の前提条件 80
	サードパーティ製アプリケーションのための MPP の設定 81
	サードパーティ製アプリケーションのためのトラフィック保護のトラブルシューティング 81

第 7 章	セキュア シェルの実装 83
	セキュア シェルの実装 83
	セキュア シェルの実装に関する前提条件 83
	セキュア シェルの実装に関する制約事項 84
	SSH の設定 85
	SSH クライアントの設定 88
	セキュア シェルの実装について 90
	SSH サーバ 90

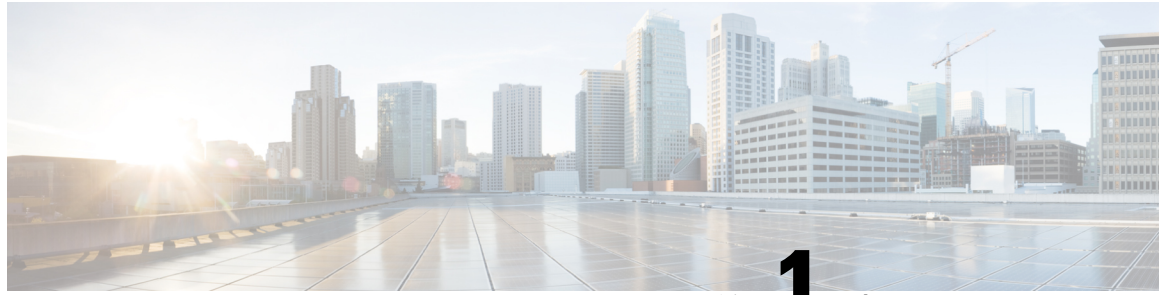
SSH クライアント 90

SFTP 機能の概要 92

RSA ベースのホスト認証 93

RSA ベースのユーザ認証 93

SSHv2 クライアント キーボードインタラクティブ認証 94



第 1 章

AAA サービスの設定

このモジュールでは、ソフトウェアシステムでユーザアクセスの制御に使用されるタスクベース認可の管理モデルの実装について説明します。タスクベース認可の実装では、主にユーザグループおよびタスクグループを設定する必要があります。

ユーザグループとタスクグループは、認証サービス、認可サービス、アカウントिंग (AAA) サービスで使用されるソフトウェアコマンドセットを介して設定されます。認証コマンドは、ユーザまたはプリンシパルの ID の検証に使用されます。認可コマンドは、認証済みユーザ（またはプリンシパル）に特定のタスクを実行するための権限が付与されていることを確認するために使用します。アカウントングコマンドは、セッションのログイン、および特定のユーザまたはシステムにより生成されるアクションを記録することで監査証跡を作成するときに使用されます。

AAA はソフトウェア ベース パッケージの一部であり、デフォルトで使用可能です。

- [AAA サービスの設定 \(1 ページ\)](#)

AAA サービスの設定

このモジュールでは、ソフトウェアシステムでユーザアクセスの制御に使用されるタスクベース認可の管理モデルの実装について説明します。タスクベース認可の実装では、主にユーザグループおよびタスクグループを設定する必要があります。

ユーザグループとタスクグループは、認証サービス、認可サービス、アカウントング (AAA) サービスで使用されるソフトウェアコマンドセットを介して設定されます。認証コマンドは、ユーザまたはプリンシパルの ID の検証に使用されます。認可コマンドは、認証済みユーザ（またはプリンシパル）に特定のタスクを実行するための権限が付与されていることを確認するために使用します。アカウントングコマンドは、セッションのログイン、および特定のユーザまたはシステムにより生成されるアクションを記録することで監査証跡を作成するときに使用されます。

AAA はソフトウェア ベース パッケージの一部であり、デフォルトで使用可能です。

AAA サービスの設定に関する前提条件

次に、AAA サービスの設定に関する前提条件を示します。

- 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。
- 初期設定ダイアログを使用してルート システム ユーザを確立します。管理者は、特定の AAA 設定なしでいくつかのローカル ユーザを設定できます。外部セキュリティ サーバは、ユーザアカウントが管理ドメイン内の複数のルータで共有される場合に必要になります。一般的な設定では、外部サーバが到達不能になった場合のバックアップとしてローカルデータベースオプションを使用して、外部 AAA セキュリティサーバおよびデータベースを使用します。

AAA サービスの設定に関する制約事項

ここでは、AAA サービスを設定する場合の制限を示します。

互換性

互換性は Cisco のフリーウェア TACACS+ サーバおよび FreeRADIUS だけで検証されます。

相互運用性

ルータの管理者は、ルータと現在 Cisco ソフトウェアを実行していない他のシスコ製機器に対して、同じ AAA サーバのソフトウェアとデータベース (CiscoSecure ACS など) を使用できます。タスク ID をサポートしない外部 TACACS+ サーバとルータとの間の相互運用性をサポートするには、「[TACACS+ および RADIUS 認証ユーザのタスク ID \(37 ページ\)](#)」の項を参照してください。

タスク グループの設定

タスクベースの認可では、その基本要素としてタスク ID の概念が使用されます。タスク ID は、ユーザの操作実行許可を定義します。各ユーザは、タスク ID で識別される許可されたルータ操作タスクのセットに関連付けられます。ユーザは、ユーザグループに関連付けられることで許可が付与されます。ユーザ グループには、タスク グループが関連付けられます。各タスク グループは 1 つ以上のタスク ID に関連付けられます。認可スキームを設定する場合、最初にタスク グループを設定します。次に、タスク グループ、個々のユーザの順に設定します。

no プレフィックスを指定して **task** コマンドを使用すると、特定のタスク ID をタスク グループから削除できます。

タスク グループ自体は削除できます。ドキュメント名のあるタスク グループを削除すると、エラーが発生します。

始める前に

タスク グループを作成して、タスク ID を関連付ける前に、タスク ID のルータ リストおよび各タスク ID の目的について理解しておく必要があります。タスク ID の完全なリストを表示するには、**show aaa task supported** コマンドを使用します。



(注) AAA タスク ID の write 許可を持っているユーザだけタスク グループを設定できます。

手順

ステップ 1 **configure**

ステップ 2 **taskgroup** *taskgroup-name*

例 :

```
RP/0/RP0/CPU0:router(config)# taskgroup beta
```

特定のタスク グループの名前を作成し、タスク グループ コンフィギュレーションサブモードを開始します。

- **taskgroup** コマンドの **no** 形式を指定すると、特定のタスク グループをシステムから削除できます。

ステップ 3 **description** *string*

例 :

```
RP/0/RP0/CPU0:router(config-tg)# description this is a sample task group description
```

(任意) ステップ 2 で指定したタスク グループの説明を作成します。

ステップ 4 **task** {**read** | **write** | **execute** | **debug**} *taskid-name*

例 :

```
RP/0/RP0/CPU0:router(config-tg)# task read bgp
```

ステップ 2 で指定したタスク グループに関連付けるタスク ID を指定します。

- そのタスク ID が関連付けられ、タスク グループのメンバにより実行される任意の CLI または API 呼び出しに **read** 許可を割り当てます。
- **no** プレフィックスを指定して **task** コマンドを使用すると、特定のタスク ID をタスク グループから削除できます。

ステップ 5 ステップ 2 で指定したタスク グループに関連付ける各タスク ID に対して手順を繰り返します。

ステップ 6 commit

次のタスク

タスク グループのすべてのセットの設定が完了したら、ユーザグループのフルセットを設定します（「ユーザグループの設定」の項を参照）。

ユーザグループの設定

ユーザグループは、タスクグループなど一連のユーザに対するコマンドパラメータによって設定されます。**usergroup** コマンドを入力すると、ユーザグループコンフィギュレーションサブモードが開始されます。**usergroup** コマンドの **no** 形式を使用すると、特定のユーザグループを削除できます。システムで参照されているユーザグループを削除すると、警告が表示されます。

始める前に



- (注) WRITE:AAA タスク ID が関連付けられているユーザだけ、ユーザグループを設定できます。ユーザグループは、事前定義されたグループのプロパティ（owner-sdr など）を継承できません。

手順

ステップ 1 configure

ステップ 2 **usergroup** *usergroup-name*

例：

```
RP/0/RP0/CPU0:router(config)# usergroup beta
```

特定のユーザグループの名前を作成し、ユーザグループコンフィギュレーションサブモードを開始します。

- **usergroup** コマンドの **no** 形式を指定すると、特定のユーザグループをシステムから削除できます。

ステップ 3 **description** *string*

例：

```
RP/0/RP0/CPU0:router(config-ug)#
description this is a sample user group description
```

(任意) ステップ 2 で指定したユーザグループの説明を作成します。

ステップ 4 **inherit usergroup** *usergroup-name*

例：

```
RP/0/RP0/CPU0:router(config-ug)#  
inherit usergroup sales
```

- ユーザグループの権限を明示的に定義します。

ステップ5 **taskgroup** *taskgroup-name*

例：

```
RP/0/RP0/CPU0:router(config-ug)# taskgroup beta
```

ステップ2で指定したユーザグループをこのステップで指定したタスクグループに関連付けます。

- ユーザグループは、入力したタスクグループに対してすでに定義されている設定属性（タスクIDリストと権限）を取ります。

ステップ6 ステップ2で指定したユーザグループを関連付ける各タスクグループに対して手順を繰り返します。

—

ステップ7 **commit**

ユーザの設定

ユーザを設定するには、次のタスクを実行します。

各ユーザは、管理ドメイン内で一意のユーザ名によって識別されます。各ユーザは、少なくとも1つのユーザグループのメンバーであることが必要です。ユーザグループを削除すると、そのグループに関連付けられたユーザが孤立する場合があります。AAAサーバでは孤立したユーザも認証されますが、ほとんどのコマンドは許可されません。

手順

ステップ1 **configure**

ステップ2 **username** *user-name*

例：

```
RP/0/RP0/CPU0:router(config)# username user1
```

新しいユーザの名前を作成（または現在のユーザを識別）して、ユーザ名コンフィギュレーションサブモードを開始します。

- *user-name* 引数には1つの単語だけ使用できます。スペースと引用符は使用できません。

ステップ3 次のいずれかを実行します。

- **password** {0 | 7} *password*
- **secret** {0 | 5} *secret*

例：

```
RP/0/RP0/CPU0:router(config-un)# password 0 pwd1
```

または

```
RP/0/RP0/CPU0:router(config-un)# secret 0 sec1
```

ステップ 2 で指定したユーザのパスワードを指定します。

- **secret** コマンドを使用して、ステップ 2 で指定したユーザ名用の安全なログインパスワードを作成します。
- **password** コマンドの後に **0** を入力すると、暗号化されていない（クリアテキスト）パスワードが続くことが指定されます。**password** コマンドの後に **7** を入力すると、暗号化されたパスワードが続くことが指定されます。
- **secret** コマンドの後に **0** を入力すると、セキュアな暗号化されていない（クリアテキスト）パスワードが続くことが指定されます。**secret** コマンドの後に **5** を入力すると、セキュアな暗号化されたパスワードが続くことが指定されます。
- タイプ **0** が、**password** コマンドおよび **secret** コマンドのデフォルトです。

ステップ 4 **group group-name**

例：

```
RP/0/RP0/CPU0:router(config-un)# group sysadmin
```

ステップ 2 で指定したユーザを **usergroup** コマンドで定義したユーザ グループに割り当てます。

- ユーザは、ユーザ グループのさまざまなタスク グループへの割り当てによって定義された内容に従って、ユーザ グループのすべての属性を受け取ります。
- 各ユーザは、少なくとも 1 つのユーザグループに割り当てする必要があります。ユーザは複数のユーザグループに属することがあります。

ステップ 5 ステップ 2 で指定したユーザに関連付けるユーザグループごとに、ステップ 4 を繰り返します。

—

ステップ 6 **commit**

RADIUS サーバ通信用のルータの設定

ルータと RADIUS サーバの通信を設定します。通常、RADIUS ホストは、シスコ（CiscoSecure ACS）、Livingston、Merit、Microsoft、または他のソフトウェアプロバイダーの RADIUS サーバソフトウェアを実行するマルチユーザシステムです。RADIUS サーバとの通信のためにルータを設定するには、次のような要素があります。

- ホスト名または IP アドレス
- 認証の宛先ポート

- アカウンティングの宛先ポート
- 再送信回数
- タイムアウト時間
- キー文字列

RADIUS セキュリティ サーバは、ホスト名または IP アドレス、ホスト名と特定のユーザ データグラム プロトコル (UDP) ポート番号、または IP アドレスおよび特定の UDP ポート番号により識別されます。固有の識別情報は、IP アドレスと UDP ポート番号の組み合わせで構成されます。これにより、RADIUS ホストとして定義されているさまざまなポートが、固有の AAA サービスを提供できるようになります。つまり、この固有識別情報を使用して、ある IP アドレスに位置する 1 台のサーバ上に複数の UDP ポートが存在する場合、それぞれの UDP ポートに対して RADIUS 要求を送信できます。同じ RADIUS サーバ上の異なる 2 つのホスト エントリに同じサービス (たとえばアカウンティング) を設定した場合、2 番めに設定したホスト エントリは、最初に設定したホスト エントリの自動スイッチオーバー バックアップとして動作します。この場合、最初のホスト エントリがアカウンティング サービスを提供できなかった場合、ネットワーク アクセス サーバは同じ装置上でアカウンティング サービス用に設定されている 2 番めのホスト エントリを試行します (試行される RADIUS ホスト エントリの順番は、設定されている順序に従います)。

RADIUS サーバと Cisco ルータは、共有秘密テキスト スtring を使用してパスワードを暗号化し、応答を交換します。RADIUS を設定して AAA セキュリティ コマンドを使用するには、RADIUS サーバデーモンを実行するホストと、ルータと共有する秘密テキスト (キー) スtring を指定する必要があります。

タイムアウト値、再送信値、および暗号キー値には、すべての RADIUS サーバを対象にしたグローバル設定、サーバ別設定、またはグローバル設定とサーバ別設定の組み合わせを使用できます。すべての RADIUS サーバとルータとの通信にこのようなグローバル設定を適用するには、**radius-server timeout**、**radius-server retransmit**、および **radius-server key** という 3 つの固有なグローバル コンフィギュレーション コマンドを使用します。特定の RADIUS サーバにこれらの値を適用するには、**radius-server host** コマンドをグローバル コンフィギュレーション モードで使用します。



(注) 同じシスコ製ネットワーク アクセス サーバで、タイムアウト、再送信、およびキー値のコマンドを同時に設定 (グローバル設定およびサーバ別設定) できます。ルータにグローバル機能とサーバ別機能の両方を設定する場合、サーバ別のタイマー、再送信、およびキー値のコマンドの方が、グローバルのタイマー、再送信、およびキー値のコマンドよりも優先されます。

手順

ステップ 1 configure

ステップ 2 **radius-server host** {hostname | ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]

例：

```
RP/0/RP0/CPU0:router(config)# radius-server host host1
```

リモート RADIUS サーバホストのホスト名または IP アドレスを指定します。

- **auth-port** *port-number* オプションを使用して、認証専用の RADIUS サーバに固有の UDP ポートを設定します。
- **acct-port** *port-number* オプションを使用して、アカウント専用 RADIUS サーバに固有の UDP ポートを設定します。
- ネットワーク アクセス サーバが単一の IP アドレスと関連付けられた複数のホストエントリを認識するように設定するには、このコマンドを必要な回数だけ繰り返します。その際、各 UDP ポート番号が異なっていることを確認してください。各 RADIUS ホストで使用するタイムアウト、再送信回数、および暗号キーの値をそれぞれ設定してください。
- タイムアウトを設定しない場合、グローバル値が使用されます。設定する場合、値の範囲は 1 ～ 1000 です。再送信値を設定しない場合、グローバル値が使用されます。設定する場合、値の範囲は 1 ～ 100 です。キー文字列を指定しない場合、グローバル値が使用されます。

(注) キーは、RADIUS サーバで使用する暗号化キーに一致するテキストストリングでなければなりません。キーの先頭にあるスペースは無視されますが、キー内のスペースとキー末尾のスペースは使用されるため、キーは常に **radius-server host** コマンド構文の最後のアイテムとして設定してください。キーにスペースを使用する場合、引用符をキーに含める場合を除き、引用符でキーを囲まないでください。

ステップ 3 radius-server retransmit *retries*

例：

```
RP/0/RP0/CPU0:router(config)# radius-server retransmit 5
```

ソフトウェアが RADIUS サーバホストのリストを検索する回数の最大値を指定します。

- この例では、再送信の試行回数は 5 に設定されます。

ステップ 4 radius-server timeout *seconds*

例：

```
RP/0/RP0/CPU0:router(config)# radius-server timeout 10
```

タイムアウトになるまでルータがサーバホストの応答を待機する秒数を設定します。

- 次に、インターバルタイマーが 10 秒に設定されている例を示します。

ステップ 5 radius-server key {0 *clear-text-key* | 7 *encrypted-key* | *clear-text-key*}

例：

```
RP/0/RP0/CPU0:router(config)# radius-server key 0 samplekey
```

ルータおよび RADIUS デーモン間のすべての RADIUS コミュニケーションの認証キーおよび暗号キーを指定します。

ステップ 6 `radius source-interface type instance [vrf vrf-id]`

例：

```
RP/0/RP0/CPU0:router(config)# radius source-interface 0/3/0/1
```

(任意) RADIUS で、すべての発信 RADIUS パケットに指定のインターフェイスまたはサブインターフェイスの IP アドレスが使用されるようにします。

- 指定されたインターフェイスまたはサブインターフェイスには、IP アドレスが関連付けられている必要があります。指定のインターフェイスまたはサブインターフェイスに IP アドレスが設定されていないか、そのインターフェイスがダウン状態にある場合、RADIUS はデフォルトに戻ります。これを回避するには、インターフェイスまたはサブインターフェイスに IP アドレスを追加するか、そのインターフェイスをアップ状態にします。

`vrf` キーワードは、VRF 単位で仕様を有効にします。

ステップ 7 設定する各外部サーバに対して ステップ 2 ~ 6 を繰り返します。

—

ステップ 8 `commit`

ステップ 9 `show radius`

例：

```
RP/0/RP0/CPU0:router# show radius
```

(任意) システムに設定されている RADIUS サーバの情報を表示します。

Radius の要約の例

```
radius source-interface Mgm0/rp0/cpu0/0 vrf default
radius-server timeout 10
radius-server retransmit 2
!
! OOB RADIUS
radius-server host 123.100.100.186 auth-port 1812 acct-port 1813
key cisco123
timeout 10
retransmit 2
!
radius-server host 123.100.100.187 auth-port 1812 acct-port 1813
key cisco123
timeout 10
retransmit 2
!
aaa group server radius radgrp
server 123.100.100.186 auth-port 1812 acct-port 1813
server 123.100.100.187 auth-port 1812 acct-port 1813
!
aaa authorization exec radauthen group radgrp local
aaa authentication login radlogin group radgrp local
!
line template vty
authorization exec radauthen
```

```
login authentication radlogin
timestamp disable
exec-timeout 0 0
!
vty-pool default 0 99 line-template vty
```

RADIUS Dead サーバ検出の設定

RADIUS Dead-Server Detection 機能を使用すると、RADIUS サーバをデッド状態と指定するために使用する条件を設定および決定できます。条件が明示的に設定されていない場合は、条件は未処理のトランザクションの数に基づいて動的に計算されます。RADIUS Dead-Server Detection を設定すると、応答を停止している RADIUS サーバが即時検出されます。この未応答 RADIUS サーバの即時検出、動きが鈍いサーバの誤検出の回避、デッド状態とライブ状態を繰り返す現象の回避が有効になると、デッドタイムが短くなり、パケット処理が高速になります。

ルータが RADIUS サーバから有効なパケットを最後に受け取ってから RADIUS サーバがデッド状態と指定されるまでに経過する必要がある最低時間を秒単位で設定することができます。ルータが起動してからパケットの受信がなく、タイムアウトになると、時間基準は満たされたものとして処理されます。

さらに、RADIUS サーバがデッド状態と指定されるまでにルータで発生する必要がある連続タイムアウト回数を設定することもできます。サーバが認証とアカウントングの両方を実行する場合、両方の種類のパケットがこの回数に含まれます。正しく作成されていないパケットは、タイムアウトになっているものとしてカウントされます。カウントされるのは再転送だけで、最初の転送はカウントされません。たとえば、各タイムアウトで1回の再送信が送信されます。



(注) 時間の条件と試行回数の条件の両方を満たしていないと、サーバはデッド状態と指定されません。

radius-server deadtime コマンドは、サーバがデッド状態とマークされて、デッド状態に留まる時間を分単位で指定します。この時間が経過すると、サーバからの応答が受信されない場合でも、サーバは稼働中とマークされます。デッド条件を設定しても、**radius-server deadtime** コマンドを設定しない限り、サーバはモニタされません。

手順

ステップ 1 configure

ステップ 2 radius-server deadtime minutes

例：

```
RP/0/RP0/CPU0:router(config)# radius-server deadtime 5
```

いくつかのサーバが使用不能になったときの RADIUS サーバの応答時間を短くし、使用不能になったサーバがすぐにスキップされるようにします。

ステップ 3 radius-server dead-criteria time seconds

例 :

```
RP/0/RP0/CPU0:router(config)# radius-server dead-criteria time 5
```

RADIUS サーバがデッド状態として指定されるデッド条件の時間を確立します。

ステップ 4 radius-server dead-criteria tries tries

例 :

```
RP/0/RP0/CPU0:router(config)# radius-server dead-criteria tries 4
```

RADIUS サーバがデッド状態として指定されるデッド条件の試行回数を確立します。

ステップ 5 commit**ステップ 6 show radius dead-criteria host ip-addr [auth-port auth-port] [acct-port acct-port]**

例 :

```
RP/0/RP0/CPU0:router# show radius dead-criteria host 172.19.192.80
```

(任意) 指定 IP アドレスで RADIUS サーバに要求された `dead-server-detection` 情報を表示します。

TACACS+ サーバの設定

TACACS+ サーバを設定します。

ポートが指定されていない場合、標準ポート番号 **49** がデフォルトで使用されます。 **timeout** および **key** パラメータは、すべての TACACS+ サーバに対してグローバルで指定できます。 **timeout** パラメータは、AAA サーバが TACACS+ サーバから応答を受信するまでの時間を指定します。 **key** パラメータは、AAA サーバと TACACS+ サーバ間で共有される認証および暗号キーを指定します。

手順

ステップ 1 configure**ステップ 2 tacacs-server host host-name port port-number**

例 :

```
RP/0/RP0/CPU0:router(config)# tacacs-server host 209.165.200.226 port 51
RP/0/RP0/CPU0:router(config-tacacs-host)#
```

TACACS+ ホストサーバを指定し、オプションでサーバポート番号を指定します。

- このオプションによって、デフォルトのポート **49** は上書きされます。有効なポート番号の範囲は **1 ~ 65535** です。

ステップ 3 tacacs-server host *host-name* timeout *seconds*

例 :

```
RP/0/RP0/CPU0:router(config-tacacs-host)# tacacs-server host 209.165.200.226 timeout 30
```

```
RP/0/RP0/CPU0:router(config)#
```

TACACS+ ホスト サーバを指定し、オプションで、AAA サーバが TACACS+ サーバからの応答を待機する時間の長さを設定するタイムアウト値を指定します。

- このオプションを指定すると、このサーバに限り、**tacacs-server timeout** コマンドで設定されたグローバルタイムアウト値が上書きされます。タイムアウト値は、タイムアウト間隔を指定する整数として表されます。範囲は 1 ~ 1000 です。

ステップ 4 tacacs-server host *host-name* key [0 | 7] *auth-key*

例 :

```
RP/0/RP0/CPU0:router(config)# tacacs-server host 209.165.200.226 key 0 a_secret
```

TACACS+ ホスト サーバを指定し、オプションで、AAA サーバと TACACS+ サーバ間で共有される認証および暗号キーを指定します。

- TACACS+ パケットは、このキーを使って暗号化されます。このキーは TACACS+ デーモンで使用されるキーと一致する必要があります。このキーを指定すると、このサーバに限り、**tacacs-server key** コマンドで設定されたグローバル キーが上書きされます。
- (任意) **0** を入力することにより、暗号化されていない (クリアテキスト) キーが続くことを指定します。
- (任意) **7** を入力することにより、暗号化された (クリアテキスト) キーが続くことを指定します。
- *auth-key* 引数は、AAA サーバと TACACS+ サーバ間で共有される暗号化されたまたは暗号化されていないキーを指定します。

ステップ 5 tacacs-server host *host-name* single-connection

例 :

```
RP/0/RP0/CPU0:router(config)# tacacs-server host 209.165.200.226 single-connection
```

単一 TCP 接続を介してすべての TACACS+ 要求をこのサーバに多重化するようにルータを設定します。デフォルトでは、セッションごとに別個の接続が使用されます。

ステップ 6 tacacs source-interface *type instance*

例 :

```
RP/0/RP0/CPU0:router(config)# tacacs source-interface 0/4/0/0
```

(任意) すべての発信 TACACS+ パケットに対して、選択したインターフェイスの発信元 IP アドレスを指定します。

- 指定されたインターフェイスまたはサブインターフェイスには、IP アドレスが関連付けられている必要があります。指定のインターフェイスまたはサブインターフェイスに IP ア

ドレスが設定されていないか、そのインターフェイスがダウン状態にある場合、TACACS+ はデフォルトインターフェイスに戻ります。これを回避するには、インターフェイスまたはサブインターフェイスに IP アドレスを追加するか、そのインターフェイスをアップ状態にします。

- **vrf** オプションは、AAA TACACS+ サーバグループのバーチャルプライベートネットワーク (VPN) ルーティングおよび転送 (VRF) 参照を指定します。

ステップ 7 設定する各外部サーバに対して ステップ 2 ~ 5 を繰り返します。

ステップ 8 **commit**

ステップ 9 **show tacacs**

例：

```
RP/0/RP0/CPU0:router# show tacacs
```

(任意) システムに設定されている TACACS+ サーバの情報を表示します。

Tacacs の要約の例：

```
! OOB TAC
tacacs-server host 123.100.100.186 port 49
key lm51
!
tacacs-server host 123.100.100.187 port 49
key lm51
!
aaa group server tacacs+ tacgrp
server 123.100.100.186
server 123.100.100.187
!
aaa group server tacacs+ eem
server 123.100.100.186
server 123.100.100.187
!
aaa authorization exec tacauthen group tacgrp local
aaa authentication login taclogin group tacgrp local
!
line console
authorization exec tacauthen
login authentication taclogin
timeout login response 30
timestamp
exec-timeout 0 0
session-timeout 15
!
vty-pool default 0 99 line-template console
```

RADIUS サーバグループの設定

この作業では、RADIUS サーバグループを設定します。

1つ以上の **server** コマンドを入力できます。 **server** コマンドは、外部 RADIUS サーバのホスト名または IP アドレスをポート番号とともに指定します。設定されている場合、このサーバグループは、AAA 方式リスト（認証、認可またはアカウントिंगの設定に使用されます）から参照できます。

始める前に

正常に設定を行うため、外部サーバが設定時にアクセスできる必要があります。

手順

ステップ 1 **configure**

ステップ 2 **aaa group server radius group-name**

例：

```
RP/0/RP0/CPU0:router(config)# aaa group server radius radgroup1
```

各種サーバホストを別個のリストにグループ化し、サーバグループ 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 **server {hostname | ip-address} [auth-port port-number] [acct-port port-number]**

例：

```
RP/0/RP0/CPU0:router(config-sg-radius)# server 192.168.20.0
```

外部 RADIUS サーバのホスト名または IP アドレスを指定します。

- サーバグループは、設定されると、AAA 方式リスト（認証、認可またはアカウントिंगの設定に使用されます）から参照できます。

ステップ 4 ステップ 3 で指定したサーバグループに追加する各外部サーバに対し、ステップ 4 を繰り返します。

—

ステップ 5 **deadtime minutes**

例：

```
RP/0/RP0/CPU0:router(config-sg-radius)# deadtime 1
```

RADIUS サーバグループ レベルでデッドタイム値を設定します。

- *minutes* 引数には、トランザクション要求によって RADIUS サーバをスキップする時間を分単位で指定します。最長 1440 分（24 時間）まで指定できます。指定できる範囲は 1 ～ 1440 です。

RADIUS サーバグループ **radgroup1** が認証要求への応答に失敗したときの **deadtime** コマンドに対して、1 分のデッドタイムを指定する例を示します。

（注） グループの作成後にグループレベルのデッドタイムを設定できます。

ステップ 6 commit**ステップ 7 show radius server-groups [group-name [detail]]**

例：

```
RP/0/RP0/CPU0:router# show radius server-groups
```

(任意) システムで設定されている各 RADIUS サーバグループの情報を表示します。

次のタスク

RADIUS サーバグループを設定したら、認証、認可およびアカウントिंगを設定して方式リストを定義します

TACACS+ サーバグループの設定

TACACS+ サーバグループを設定します。

1 つ以上の **server** コマンドを入力できます。 **server** コマンドは、外部 TACACS+ サーバのホスト名または IP アドレスを指定します。設定後は、このサーバグループは、AAA 方式リスト (認証、認可またはアカウントिंगの設定に使用されます) から参照できます

始める前に

正常に設定を行うため、外部サーバが設定時にアクセスできる必要があります。グローバルおよび VRF 設定の両方に同じ IP アドレスを設定するときは、**server-private** パラメータが必要です。

手順

ステップ 1 configure**ステップ 2 aaa group server tacacs+ group-name**

例：

```
RP/0/RP0/CPU0:router(config)# aaa group server tacacs+ tacgroup1
```

各種サーバホストを別個のリストにグループ化し、サーバグループ 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 server {hostname | ip-address}

例：

```
RP/0/RP0/CPU0:router(config-sg-tacacs+)# server 192.168.100.0
```

外部 TACACS+ サーバのホスト名または IP アドレスを指定します。

- 設定されている場合、このグループは、AAA 方式リスト (認証、認可またはアカウントिंगの設定に使用されます) から参照できます

ステップ 4 (任意) `vrf vrf-id`

例 :

```
RP/0/RP0/CPU0:router(config-sg-tacacs)# vrf vrf-id
```

`vrf` オプションは、AAA TACACS+ サーバグループのバーチャルプライベート ネットワーク (VPN) ルーティングおよび転送 (VRF) 参照を指定します。

ステップ 5 ステップ 2 で指定したサーバグループに追加する各外部サーバに対し、ステップ 3 を繰り返します。

—

ステップ 6 (任意) `vrf vrf-id`

`vrf` オプションは、AAA TACACS+ サーバグループのバーチャルプライベート ネットワーク (VPN) ルーティングおよび転送 (VRF) 参照を指定します。

ステップ 7 `commit`**ステップ 8** `show tacacs server-groups`

例 :

```
RP/0/RP0/CPU0:router# show tacacs server-groups
```

(任意) システムで設定されている各 TACACS+ サーバグループの情報を表示します。

一連の認証方式の作成

認証は、ユーザ（またはプリンシパル）が検証されるプロセスです。認証設定は、方式リストを使用して、さまざまなデータソースに保存されている、AAA データソースの優先順位を定義します。認証を設定して、複数の方式リストを定義できます。アプリケーションは（ログインなど）、これらのいずれかを選択できます。たとえば、コンソールポートと VTY ポートとで異なる方式リストを使用できます。方式リストが指定されていない場合、アプリケーションは、デフォルトの方式リストを使用します。



(注) 方式リストが有効になるようにするには、アプリケーションは明示的に定義済みの方式リストを示す必要があります。

認証は、**login authentication** ラインコンフィギュレーションサブモードコマンドを使用して、TTY 回線に適用できます。方式が、サーバグループではなく、RADIUS または TACACS+ サーバの場合、RADIUS または TACACS+ サーバは、設定されている RADIUS および TACACS+ サーバのグローバルプールから、設定順に選択されます。このグローバルプールから選択されるサーバは、サーバグループに追加できるサーバです。

後続の認証方式は、初期方式がエラーを返すか、要求が拒否された場合だけ使用されます。

始める前に



- (注) デフォルトの方式リストは、デフォルト以外の名前付き方式リストが明示的に設定されている場合（この場合は名前付き方式リストが適用される）を除き、認証のすべてのインターフェイスに適用されます。

aaa authentication コマンドの **group radius**, **group tacacs+** および **group group-name** 形式は、以前に定義した一連の RADIUS サーバまたは TACACS+ サーバを参照します。ホストサーバを設定するには、**radius server-host or tacacs-server host** コマンドを使用します。特定のサーバグループを作成するには、**aaa group server radius or aaa group server tacacs+** コマンドを使用します。

手順

ステップ 1 configure

ステップ 2 **aaa authentication {login} {default | list-name} method-list**

例：

```
RP/0/RP0/CPU0:router(config)# aaa authentication login default group tacacs+
```

一連の認証方式、つまり方式リストを作成します。

- **login** キーワードを使用すると、ログインの認証が設定されます。 **ppp** キーワードを使用すると、ポイントツーポイントプロトコルの認証が設定されます。
- **default** キーワードを入力すると、このキーワードの後ろにリストされている認証方式が、認証のデフォルトの方式リストになります。
- **list-name** 文字列を入力すると、認証方式リストが識別されます。
- 方式リストのタイプの後ろに **method-list** 引数を入力します。方式リストタイプは、目的の順序で入力します。リストされる方式タイプは、次のいずれかのオプションです。
 - **group tacacs+** : サーバグループまたは TACACS+ サーバを認証に使用します
 - **group radius** : サーバグループまたは RADIUS サーバを認証に使用します
 - **groupnamed-group** : TACACS+ サーバまたは RADIUS サーバの名前付きサブセットを認証に使用します。
 - **local** : ユーザ名またはパスワードのローカルデータベースを認証に使用します
 - **line** : 回線パスワードまたはユーザグループを認証に使用します
- この例では、**default**方式リストが認証に使用されます。

ステップ 3 commit

ステップ4 設定されるすべての認証方式リストに対して、ステップ1～3を繰り返します。

—

一連の許可方式の作成

許可方式リストによって、許可の実行方法とこれらの方式の実行順序が定義されます。方式リストは、シーケンスで照会される認可方式（TACACS+ など）を説明する単なる名前付きリストです。方式リストを使用すると、認可に使用するセキュリティプロトコルを1つ以上指定できるため、最初の方式が失敗した場合のバックアップシステムを確保できます。ソフトウェアは方式リストの最初の方式を使用して、特定のネットワークサービスに対してユーザを許可します。その方式が応答に失敗すると、方式リスト内の次の方式が選択されます。このプロセスは、リスト内の許可方式との通信に成功するまで、または定義されている方式を使い果たすまで続行されます。



(注) ソフトウェアは、前の方式から応答がない場合またはエラー（障害ではない）応答が返された場合のみ、次に指定されている方式を使って許可を試みます。このサイクルの任意の時点で認可が失敗した場合（つまり、セキュリティサーバまたはローカルユーザ名データベースからユーザサービスの拒否応答が返される場合）、認可プロセスは停止し、その他の認可方式は試行されません。

名前付き方式リストを作成すると、指定した許可タイプに対して特定の許可方式リストが定義されます。方式リストを定義した場合、定義した方式のいずれかを実行するには、まず特定の回線またはインターフェイスに方式リストを適用する必要があります。新しいメソッドリストを作成するときに、TACACS+などの方式名を使用しないでください。

ラインテンプレートにコマンド認可方式リストを適用した結果の「コマンド」の許可とは別個であり、ルータで自動的に実行される「タスクベース」の許可に加えて実行される方式です。コマンドの許可のデフォルト動作はnoneです。デフォルトの方式リストが設定されていても、その方式リストは、使用するためのラインテンプレートに追加する必要があります。

aaa authorization commands コマンドにより、許可プロセスの一環として、一連の属性値（AV）ペアを含む要求パケットがTACACS+デーモンに送信されます。デーモンは、次のいずれかのアクションを実行できます。

- 要求をそのまま受け入れます。
- 許可を拒否します。

aaa authorization コマンドを使用して、認可パラメータを設定し、各回線またはインターフェイスで使用できる特定の許可方式を定義する名前付き方式リストを作成します。

手順

ステップ1 configure

ステップ2 **aaa authorization** {**commands** | **eventmanager** | **exec** | **network**} {**default** | *list-name*} {**none** | **local** | **group** {**tacacs+** | **radius** | *group-name*}}

例 :

```
RP/0/RP0/CPU0:router(config)# aaa authorization commands listname1 group tacacs+
```

一連の認可方式、つまり方式リストを作成します。

- **commands** キーワードは、すべての XR EXEC モード シェル コマンドに対して許可を設定します。コマンドの認可は、ユーザにより発行される EXEC モード コマンドに適用されます。コマンドの認可では、すべての XR EXEC モード コマンドに対して許可が試行されます。
- **eventmanager** キーワードは、イベント マネージャ (障害マネージャ) を許可するための許可方式を適用します。
- **exec** キーワードは、インタラクティブセッション (XR EXEC モード) に対する許可を設定します。
- **network** キーワードは、PPP または IKE などのネットワーク サービスに対する許可を設定します。
- **default** キーワードを指定すると、このキーワードの後ろにリストされている許可方式が、許可のデフォルトの方式リストになります。
- **list-name** 文字列は、許可方式リストを識別します。方式リスト自体は、方式リスト名に続きます。方式リスト タイプは、目的の順序で入力します。リストされる方式リスト タイプは、次のいずれかにできます。
 - **none** : ネットワーク アクセス サーバ (NAS) は、許可情報を要求しません。認可は常に成功します。以降の認可方式は試行されません。ただし、タスク ID の許可は常に必要であり、ディセーブルにはできません。
 - **local** : ローカル データベースを認可に使用します。
- **group tacacs+** : 設定されているすべての TACACS+ サーバのリストを許可に使用します。NAS は、認可情報を TACACS+ セキュリティ デモンと交換します。TACACS+ 認可は、AV ペアを関連付けることでユーザに特定の権限を定義します。AV は適切なユーザとともに TACACS+ セキュリティ サーバのデータベースに保存されます。
- **group radius** : 設定されているすべての RADIUS サーバのリストを許可に使用します。
- **group group-name** : **aaa group server tacacs+** または **aaa group server radius** コマンドによって定義されている名前付きサーバグループ、TACACS+ または RADIUS サーバのサブセットを許可に使用します。

ステップ3 **commit**

一連のアカウントング方式の作成

aaa accounting コマンドを使用して、デフォルトまたは名前付き方式リストを作成し、各回線またはインターフェイスに使用可能な特定のアカウントング方式を定義します。

現時点では、アカウントングに対して TACACS+ および RADIUS の両方の方式がサポートされています。ルータは、アカウントングレコードの形式で TACACS+ または RADIUS セキュリティ サーバにユーザ アクティビティを報告します。各アカウントングレコードは、アカウントング AV ペアが含まれ、セキュリティサーバ上で保管されます。

アカウントング方式リストには、アカウントングの実行方法が定義されます。このリストを使用して、特定のタイプのアカウントングサービスに固有の回線またはインターフェイスに使用する特定のセキュリティプロトコルを指定できます。方式リストの名前を付ける場合、TACACS+ などの方式の名前を使用しないでください。

最低限のアカウントングを行う場合は、**stop-only** キーワードを指定して、要求されたユーザプロセスの終了時に「stop accounting」通知を送信するようにします。詳細なアカウントングを行う場合は、**start-stop** キーワードを使用できます。このキーワードを指定すると、外部 AAA サーバは要求されたプロセスの開始時に「start accounting」通知を送信し、プロセスの終了時に「stop accounting」通知を送信します。また、**aaa accounting update** コマンドを使用して、累積情報を含む更新レコードを定期的に送信できます。アカウントングレコードは、TACACS+ または RADIUS サーバだけに格納されます。

AAA アカウントングをアクティブにすると、ルータは、これらの属性をアカウントングレコードとして報告します。そのアカウントングレコードは、セキュリティサーバ上のアカウントングログに格納されます。

手順

ステップ 1 configure

ステップ 2 次のいずれかを実行します。

- **aaa accounting** {commands | exec | network} {default | list-name} {start-stop | stop-only}
- {none | method}

例：

```
RP/0/RP0/CPU0:router(config)# aaa accounting commands default stop-only group tacacs+
```

(注) コマンドアカウントングは RADIUS ではサポートされませんが、TACACS ではサポートされます。

一連のアカウントング方式、つまり方式リストを作成します。

- **commands** キーワードは、XR EXEC モードシェルコマンドでアカウントングを有効にします。
- **exec** キーワードは、インタラクティブ (XR EXEC モード) セッションに対するアカウントングを有効にします。

- **network** キーワードは、ポイントツーポイント プロトコル (PPP) など、ネットワーク関連のすべてのサービス要求に対してアカウントングを有効にします。
- **default** キーワードを指定すると、このキーワードの後ろにリストされているアカウントング方式が、アカウントングのデフォルトの方式リストになります。
- **list-name** 文字列は、アカウントング方式リストを識別します。
- **start-stop** キーワードは、プロセスの開始時に「start accounting」通知を送信し、プロセスの終了時に「stop accounting」通知を送信します。要求されたユーザ プロセスは、「start accounting」通知がアカウントング サーバで受信されたかどうかに関係なく開始されません。
- **stop-only** キーワードは、要求されたユーザ プロセスの終了時に「stop accounting」通知を送信します。
- **none** キーワードは、アカウントングを行わないことを指定します。
- 方式リスト自体は、**start-stop** キーワードの後ろに続きます。方式リスト タイプは、目的の順序で入力します。method 引数には次のタイプがあります。
 - **group tacacs+** : アカウントングにすべての設定済み TACACS+ サーバのリストを使用します。
 - **group radius**—アカウントングにすべての設定済み RADIUS サーバのリストを使用します。
 - **group group-name** : **aaa group server tacacs+** または **aaa group server radius** コマンドによって定義されている名前付きサーバグループ、TACACS+ または RADIUS サーバのサブセットをアカウントングに使用します。
- この例は、**default** コマンドによるアカウントング方式リストの定義を示しています。アカウントング サービスは TACACS+ セキュリティ サーバによって提供され、**stop-only** 制限が設定されています。

ステップ 3 commit

中間アカウントングレコードの生成

アカウントング サーバに送信される定期的中間アカウントングレコードをイネーブルにします。**aaa accounting update** コマンドをアクティブにすると、システム上のすべてのユーザの中間アカウントングレコードが発行されます。



- (注) 中間アカウントングレコードは、インターネットキー交換 (IKE) アカウントングなど、ネットワークセッションに対してのみ生成されます。これは、**network** キーワードを指定した **aaa accounting** コマンドで制御されます。システム、コマンドまたは EXEC アカウントングセッションでは、中間レコードは生成されません。

手順

ステップ1 **configure**ステップ2 **aaa accounting update {newinfo | periodic minutes}**

例：

```
RP/0/RP0/CPU0:router(config)# aaa accounting update periodic 30
```

アカウントिंग サーバに送信される定期的中間アカウントング レコードをイネーブルにします。

- **newinfo** キーワードを使用した場合は、レポートする新しいアカウントング情報が発生するたびに、中間アカウントングレコードがアカウントングサーバに送信されます。たとえば、IPCP がリモートピアとの間で IP アドレスのネゴシエーションを完了したときなどです。中間アカウントングレコードには、リモートピアに使用されるネゴシエート済み IP アドレスが含まれます。
- **periodic** キーワードを使用すると、中間アカウントングレコードは引数の数字で定義されたとおり定期的に送信されます。中間アカウントングレコードには、中間アカウントングレコードが送信される時間までに、そのユーザについて記録されたすべてのアカウントング情報が含まれます。

注意 **periodic** キーワードを使用すると、多数のユーザがネットワークにログインしているときに、大きな輻輳が生じる場合があります。

ステップ3 **commit**

方式リストの適用

aaa authorization コマンドを使用して、特定のタイプの許可に対して名前付き許可方式リストを定義（またはデフォルトの方式リストを使用）した後、許可を実行する該当の回線に、定義済みのリストを適用する必要があります。**authorization** コマンドを使用し、選択した回線または回線グループに指定の方式リスト（または、方式リストを指定していない場合はデフォルトの方式リスト）を適用します。

手順

ステップ1 **configure**ステップ2 **line { console | default | template template-name}**

例：

```
RP/0/RP0/CPU0:router(config)# line console
```

回線テンプレート 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 authorization {commands | exec} {default | list-name}

例：

```
RP/0/RP0/CPU0:router(config-line)# authorization commands listname5
```

AAA 認可を特定の回線または回線のグループに対してイネーブルにします。

- **commands** キーワードは、すべてのコマンドに対して、選択した回線における許可を有効にします。
- **exec** キーワードは、インタラクティブ (XR EXEC モード) セッションに対する許可を有効にします。
- **default** キーワードを入力し、**aaa authorization** コマンドで定義されているように、デフォルトの方式リストの名前を適用します。
- 使用する認可方式リストの名前を入力します。リスト名を指定しない場合は、デフォルト名が使用されます。このリストは **aaa authorization** コマンドを使用して作成されます。
- 次に、方式リスト listname5 を使用したコマンド認可の例を示します。

ステップ 4 commit

次のタスク

AAA 認可をイネーブルにして認可方式リストを適用したら、AAA アカウンティングをイネーブルにしてアカウンティング方式リストを適用します

アカウントिंगサービスのイネーブル化

アカウントिंगサービスを特定の回線または回線のグループに対してイネーブルにします。

手順

ステップ 1 configure**ステップ 2 line { console | default | template template-name}**

例：

```
RP/0/RP0/CPU0:router(config)# line console
```

回線テンプレート 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 accounting {commands | exec} {default | list-name}

例：

```
RP/0/RP0/CPU0:router(config-line)# accounting commands listname7
```

AAA アカウンティングを特定の回線または回線のグループに対してイネーブルにします。

- **commands** キーワードは、すべての XR EXEC モード シェル コマンドに対して、選択した回線におけるアカウントिंगを有効にします。
- **exec** キーワードは、インタラクティブ (XR EXEC モード) セッションに対するアカウントिंगを有効にします。
- **default** キーワードを入力し、**aaa accounting** コマンドで定義されているように、デフォルトの方式リストの名前を適用します。
- 使用するアカウントिंग方式リストの名前を指定します。リスト名を指定しない場合は、デフォルト名が使用されます。このリストは **aaa accounting** コマンドを使用して作成されます。
- 次に、方式リスト `listname7` を使用したコマンドアカウントिंगの例を示します。

ステップ 4 commit

次のタスク

AAA アカウントिंग サービスをイネーブルにしてアカウントING方式リストを適用したら、ログインパラメータを設定します

ログインパラメータの設定

サーバがログインの応答を待機する間隔を設定します。

手順

ステップ 1 configure

ステップ 2 line template *template-name*

例 :

```
RP/0/RP0/CPU0:router(config)# line template alpha
```

設定する回線を指定して、回線テンプレート 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 timeout login response *seconds*

例 :

```
RP/0/RP0/CPU0:router(config-line)# timeout login response 20
```

サーバがログインの応答を待つ時間を設定します。

- *seconds* 引数には、タイムアウト間隔 (秒単位) を 0 ~ 300 の範囲で指定します。デフォルトは 30 秒です。
- この例では、インターバル タイマーを 20 秒に変更します。

ステップ 4 commit

タスク マップ

外部 TACACS+ サーバおよび RADIUS サーバを使用して認証されるユーザに対して、Cisco IOS XR ソフトウェア AAA は、タスク ID をリモートで定義する方式をサポートします。

タスク スtring の形式

TACACS+サーバのコンフィギュレーションファイルのタスク文字列は、カンマ (,) で区切られたトークンで構成されます。各トークンは、タスク ID 名およびその許可、またはこの特定のユーザを含むユーザ グループのいずれかで構成されます (次の例を参照)。

```
task = " permissions : taskid name , # usergroup name , ..."
```



- (注) Cisco IOS XR ソフトウェアでは、タスク ID を外部 RADIUS または TACACS+ サーバの属性として指定できます。サーバが非 Cisco IOS XR ソフトウェア システムと共有される場合、これらの属性には、サーバマニュアルで示されているように、オプションマークが付けられます。たとえば、CiscoSecure ACS および Cisco のフリーウェア TACACS+ サーバでは、オプション属性の属性値の前に等号記号 (=) ではなく、アスタリスク (*) が必要です。属性をオプションとして設定する場合、TACACS+ サーバのマニュアルを参照してください。

たとえば、user1 BGP という名前のユーザに、read、write および execute 許可を付与し、user1 を operator という名前のユーザ グループに含める場合、外部サーバの TACACS+ コンフィギュレーション ファイルのユーザ名エントリは次のようになります。

```
user = user1{
member = some-tac-server-group
opap = cleartext "lab"
service = exec {
task = "rwx:bgp,#operator"
}
}
```

r、w、x、d はそれぞれ read、write、execute、debug に対応し、ポンド記号 (#) はユーザ グループが続くことを示します。



- (注) CiscoIOS ソフトウェアに基づいたシステムとの相互運用性をイネーブルにするには、「task」の前にオプション キーワードを追加する必要があります。

CiscoSecure ACS が使用される場合、次の手順を実行して、タスク ID とユーザ グループを指定します。

手順

- ステップ1 ユーザ名とパスワードを入力します。
- ステップ2 **[Group Setup]** ボタンをクリックすると、**[Group Setup]** ウィンドウが表示されます。
- ステップ3 **[Group]** ドロップダウンリストから、更新するグループを選択します。
- ステップ4 **[Edit Settings]** ボタンをクリックします。
- ステップ5 スクロール矢印を使用して、**[Shell (exec)]** チェックボックスを探します。
- ステップ6 **[Shell (exec)]** チェックボックスをオンにして、カスタム属性設定を有効にします。
- ステップ7 **[Custom attributes]** チェックボックスをオンにします。
- ステップ8 フィールドに空白や引用符を含めずに次のタスク文字列を入力します。

例：

```
task=rwx:bgp,#netadmin
```

- ステップ9 **[Submit + Restart]** ボタンをクリックしてサーバを再起動します。

次のRADIUSベンダー固有属性（VSA）の例では、ユーザは、sysadmin 事前定義タスクグループに含まれ、BGPを設定でき、OSPFの設定を表示できます。

例：

```
user Auth-Type := Local, User-Password == lab
  Service-Type = NAS-Prompt-User,
  Reply-Message = "Hello, %u",
  Login-Service = Telnet,
  Cisco-AVPair = "shell:tasks=#sysadmin,rwx:bgp,r:ospf"
```

user1 がユーザ名 user1 と適切なパスワードを使用して、正常に外部 TACACS+ サーバに接続してログインすると、XR EXEC モードで **show user tasks** コマンドを使用して、user1 が実行可能なタスクをすべて表示できます。次に例を示します。

例：

```
Username:user1
Password:
RP/0/RP0/CPU0:router# show user tasks

Task:      basic-services  :READ   WRITE   EXECUTEDEBUG
Task:      bgp             :READ   WRITE   EXECUTE
Task:      cdp             :READ
Task:      diag            :READ
Task:      ext-access     :READ           EXECUTE
Task:      logging        :READ
```

タスク文字列が指定されていない user2 という名前のユーザが外部サーバにログインすると、次の情報が表示されます。

例：

```
Username:user2
Password:
```



```
RP/0/RP0/CPU0:router# show user tasks
No task ids available
```

AAA サービスに関する情報

この項には、AAA でユーザグループやタスクグループを設定したり、リモート認証ダイヤルインユーザサービス (RADIUS) サーバまたはTACACS+サーバを設定したりする前に、ソフトウェアユーザが理解しておく必要があるすべての概念情報が記載されています。概念情報では、AAA について、およびなぜ重要なのかについても説明します。

ユーザ、ユーザグループおよびタスクグループ

ユーザ属性は、このCisco ソフトウェアの管理モデルの基礎となるものです。各ルータユーザには、次の属性が関連付けられます。

- 管理ドメイン内でユーザを一意に特定するユーザ ID (ASCII 文字列)
- 253 文字以下のパスワードおよび一方向の暗号化シークレット
- ユーザがメンバである (タスク ID などの属性をイネーブルにした) ユーザグループ (1 つ以上) のリスト

ユーザ カテゴリ

ルータユーザは、次のカテゴリに分類されます。

- ルートセキュアドメインルータ (SDR) ユーザ (特定の SDR 管理権限)
- SDR ユーザ (特定の SDR ユーザアクセス)

ルートシステムユーザ

ルートシステムユーザは、ルータシャーシ全体の「所有」が許可されたエンティティです。ルートシステムユーザは、すべてのルータコンポーネントに対して最も高い特権を持って操作し、システム内のすべてのセキュアドメインルータをモニタできます。ルータの設定時に、少なくとも1つのルートシステムユーザアカウントを作成する必要があります。ルートシステムユーザは複数作成できます。

ルートシステムユーザは次を含む設定またはモニタタスクを実行できます。

- セキュアドメインルータを設定します。
- ルート SDR ユーザを作成、削除、変更します (ルートシステムとしてセキュアドメインルータにログイン後)。
- セキュアドメインルータユーザを作成、削除、変更し、ユーザタスクの権限を設定します (ルートシステムとしてセキュアドメインルータにログイン後)。
- セキュアドメインルータに割り当てられていないファブリックラックまたはルータリソースにアクセスし、セキュアドメインルータの設定に関係なく、ルートシステムユーザが任意のルータノードに認証されるようにします。

ルート SDR ユーザ

ルート SDR ユーザは、特定の SDR の設定およびモニタリングを制御します。ルート SDR ユーザは、ユーザを作成し、SDR 内での権限を設定できます。複数のルート SDR ユーザが独立して作業できます。1つの SDR に、複数の SDR ユーザを作成できます。

ルート SDR ユーザは、特定の SDR に対して次の管理タスクを実行できます。

- SDR のセキュア ドメイン ルータ ユーザおよび権限を作成、削除、変更します。
- SDR にアクセスできるユーザ グループを作成、削除、変更します。
- SDR のほぼすべてを管理します。

ルート SDR ユーザは、ルート システム ユーザへのアクセスを拒否できません

セキュア ドメイン ルータ (SDR) ユーザ

SDR ユーザには、ルート SDR ユーザによって定義された SDR への制限付きアクセス権があります。SDR ユーザは、日常のシステムおよびネットワーク管理アクティビティを行います。セキュア ドメイン ルータ ユーザが実行できるタスクは、そのユーザが属するユーザグループに関連付けられているタスク ID によって決まります。シャーン内での複数の SDR はサポートされません。

ユーザ グループ

ユーザ グループは、アクセス権限など、属性のセットを共有するユーザの集まりです。Cisco ソフトウェアでは、システム管理者は、ユーザのグループ、およびユーザのグループに共通するジョブ特性を設定できます。ユーザはデフォルトでグループに割り当てられないので、明示的に割り当てる必要があります。ユーザは、複数のグループに割り当てることができます。

各ユーザは、1つ以上のユーザグループに関連付けることができます。ユーザグループは、次の属性を持ちます。

- ユーザグループは、ユーザの認証を定義するタスクグループのリストから構成されます。`cisco-support`以外のすべてのタスクは、デフォルトで、ルートシステムユーザに許可されています
- 各ユーザタスクには、読み取り、書き込み、実行またはデバッグ権限を割り当てることができます。

事前定義ユーザ グループ

この Cisco ソフトウェアには、属性が定義済みの一連のユーザグループが用意されています。事前定義されているグループは次のとおりです。

- **cisco-support** : このグループは、Cisco サポート チームが使用します。
- **netadmin** : すべてのシステムおよびネットワーク パラメータを制御およびモニタできます。
- **operator** : 基本権限を持つデモンストレーション グループ。

- **root-lr** : 特定のセキュア ドメイン ルータを制御およびモニタできます。
- **sysadmin** : すべてのシステム パラメータを制御およびモニタできますが、ネットワーク プロトコルを設定できません。
- **serviceadmin** : セッション ボーダー コントローラ (SBC) などのサービス管理タスク。

ユーザ定義ユーザ グループ

管理者は、特定のニーズを満たすために自分のユーザ グループを設定できます。

ユーザ グループの継承

ユーザ グループは、別のユーザ グループから属性を継承できます。(同様に、タスク グループは、別のタスク グループから属性を継承できます)。たとえば、ユーザ グループ A がユーザ グループ B から属性を継承すると、ユーザ グループ A の新しいタスク属性セットは、A と B との集合になります。グループ A がグループ B から属性を継承する場合は、グループ B を明示的に再継承しなくても、グループ B での変更がグループ A に反映されます。この点で、ユーザ グループ間の継承関係は動的です。

タスク グループ

タスク グループは、アクションのタイプ (**read**、**write** など) に対応した許容タスク ID のリストによって定義します。タスク ID は、ルータ システムで基本的に定義されます。外部ソフトウェアのタスク グループを設定するには、タスク ID 定義が事前にサポートされている必要があります。

タスク ID は、外部 TACACS+ サーバまたは RADIUS サーバで設定できます。

事前定義タスク グループ

次に、管理者が通常の初期設定で使用できる事前定義タスク グループを示します。

- **cisco-support** : Cisco サポート担当タスク
- **netadmin** : ネットワーク管理者タスク
- **operator** : オペレータの日常業務 (デモンストレーション目的)
- **root-lr** : セキュア ドメイン ルータ管理者タスク
- **sysadmin** : システム管理者タスク
- **serviceadmin** : SBC などのサービス管理タスク

ユーザ定義タスク グループ

ユーザは、特定のニーズを満たすために独自のタスク グループを設定できます。

グループ継承

タスク グループは、他のタスク グループからの継承をサポートします (同様に、ユーザ グループは、別のユーザ グループから属性を継承できます)。たとえば、タスク グループ A がタスク

XR モードおよび管理モードでのコマンド アクセス

グループ B から継承する場合、タスク グループ A の新しい属性セットは A と B の集合となります。

XR モードおよび管理モードでのコマンド アクセス

admin コマンドを使用して XR モードからシステム管理モードにアクセスすると、XR ユーザグループとタスクがシステム管理 VM グループにマッピングされます。ユーザはシステム管理 VM で対応するアクセス権限を使用できます。現時点では、AAA グループ、管理タスクグループ、root lr グループのみがシステム管理 VM グループまたはタスクにマッピングされます。プロトコルなどの他のタスクはマッピングされません。システム管理 VM ではこれらのサービスがサポートされていないからです。システム管理 VM のディザスタリカバリ ユーザは、ホスト VM と同期されます。

XR のタスク またはグループ	Sysadmin VM グループ	アクセス	例
root-lr	ルートシステムグループ	システム設定へのフルアクセス	<pre>RP/0/RP0/CPU0:ios#show user group Mon Nov 3 13:48:54.536 UTC root-lr, cisco-support RP/0/RP0/CPU0:ios#show user tasks inc root-lr Mon Nov 3 13:49:06.495 UTC Task: root-lr : READ WRITE EXECUTE DEBUG (reserved) RP/0/RP0/CPU0:ios#admin sysadmin-vm:0_RP0# show aaa user-group Mon Nov 3 13:48:00.790 UTC User group : root-system</pre>
Admin-r/w/x/d	Admin-r	Sysadmin VM で読み取り専用コマンド	<pre>taskgroup tg-admin-write task write admin task execute admin ! usergroup ug-admin-write taskgroup tg-admin-write ! username admin-write group ug-admin-write password admin-write ! RP/0/RP0/CPU0:ios#show user group Mon Nov 3 14:09:29.676 UTC ug-admin-write RP/0/RP0/CPU0:ios#show user tasks Mon Nov 3 14:09:35.244 UTC Task: admin : READ WRITE EXECUTE RP/0/RP0/CPU0:ios#admin Mon Nov 3 14:09:40.401 UTC admin-write connected from 127.0.0.1 using console on xr-vm_node0_RP0_CPU0 sysadmin-vm:0_RP0# show aaa user-group Mon Nov 3 13:53:00.790 UTC User group : admin-r</pre>

XRのタスク またはグ ループ	Sysadmin VM グループ	アクセス	例
Netadmin ま たは sysadmin グ ループ Admin-r/ wx /d, aaa -r/w/x/d	Aaa -r およ び admin -r	Sysadmin VM で読み 取り専用コマンド	<pre>RP/0/RP0/CPU0:ios#show user group Mon Nov 3 13:44:39.176 UTC netadmin RP/0/RP0/CPU0:ios#show user tasks inc aaa Mon Nov 3 13:45:00.999 UTC Task: aaa : READ RP/0/RP0/CPU0:ios#show user tasks inc admin Mon Nov 3 13:45:09.567 UTC Task: admin : READ RP/0/RP0/CPU0:ios#admin Mon Nov 3 13:46:21.183 UTC netadmin connected from 127.0.0.1 using console on xr-vm_node0_RP0_CPU0 sysadmin-vm:0_RP0# show aaa user-group Mon Nov 3 13:44:23.939 UTC User group : admin-r,aaa-r sysadmin-vm:0_RP0#</pre>

管理モデル

ルータは、管理 (admin) プレーンとセキュア ドメイン ルータ (SDR) プレーンの 2 つのプレーンで機能します。admin (共有) プレーンは、すべての SDR で共有されるリソースで構成され、SDR プレーンは、特定の SDR に固有なリソースで構成されます。

各 SDR には、ローカル ユーザ、グループ、TACACS+ および RADIUS 設定など、独自の AAA 設定があります。1 つの SDR で作成されたユーザは、これらのユーザが他の SDR で設定されない限り、他の SDR にアクセスできません。

管理アクセス

システムへの管理アクセスは、次の操作を十分理解していない場合、または注意して計画していない場合、失われる可能性があります。

- 使用できないリモート AAA サーバを使用する認証 (特にコンソールの認証) を設定する。



(注) 他の方式リストを指定せずに **none** オプションを使用することはサポートされていません。

- コンソールでコマンド認可または XR EXEC モード認可を設定する場合は十分に注意してください。これは、この設定により TACACS+ サーバが使用できなくなる、またはすべてのコマンドが拒否され、ユーザがロックアウトされる場合があるためです。このロックアウトは、特に、TACACS+ サーバで認識されていないユーザで認証が行われる場合、あるいは TACACS+ ユーザで何らかの理由によりほとんど、またはすべてのコマンドが拒否される場合に発生します。

ロックアウトを回避するには、次のいずれか、または両方を推奨します。

- コンソールで TACACS+ コマンド認可または XR EXEC モード認可を設定する前に、認可を設定するユーザが、TACACS+ プロファイルの適切なユーザ権限を使用してログインしていることを確認してください。
- サイトのセキュリティポリシーで許可されている場合は、コマンド認可や XR EXEC モード認可に **none** オプションを使用します。これにより、TACACS+ サーバに到達できない場合に、AAA が **none** 方式にロールオーバーするので、ユーザはコマンドを実行することができます。

AAA データベース

AAA データベースは、ユーザ、グループ、およびシステムへのアクセスを制御するタスクの情報を保存します。AAA データベースはローカルまたはリモートにできます。特定の状況で使用されるデータベースは、AAA 設定により異なります。

ローカル データベース

ユーザ、ユーザグループ、タスクグループなどの AAA データは、セキュアドメインルータ内でローカルに保存できます。このデータは、メモリ内データベースに保存され、コンフィギュレーションファイルに保存されます。保存されたパスワードは暗号化されます。



(注) データベースは、保存されている特定のセキュアドメインルータ (SDR) に対してローカルで、定義されているユーザまたはグループは、同じシステムの他の SDR に表示されません。

残りすべてのユーザをローカルデータベースから削除できます。すべてのユーザを削除すると、ユーザが次にログインするときに、設定ダイアログが表示され、新しいユーザ名およびパスワードを入力するよう求められます。



(注) 設定ダイアログは、ユーザがコンソールにログインするときだけ表示されます。

リモート データベース

AAA データは、CiscoSecure ACS などの外部セキュリティサーバに保存することができます。サーバに保存されているデータは、クライアントがサーバの IP アドレスと共有秘密がわかっているならば、任意のクライアント (ネットワークアクセスサーバ [NAS] など) が使用できます。

リモート AAA 設定

CiscoSecure ACS のような製品は、共有または外部 AAA データベースを管理するために使用できます。ルータは、標準的な IP ベースのセキュリティプロトコル (TACACS+ または RADIUS など) を使用してリモート AAA サーバと通信します。

クライアント設定

セキュリティサーバは、ルータと共有するシークレットキーおよびクライアントのIPアドレスで設定する必要があります。

ユーザグループ

外部サーバで作成されるユーザグループは、ルータのローカルAAAデータベース設定のユーザグループとは関係がありません。外部TACACS+サーバまたはRADIUSサーバユーザグループの管理は別であるため、ルータはユーザグループ構造を認識しません。リモートユーザまたはグループプロファイルには、ユーザが属するグループ（ルータで定義）、および個々のタスクIDを指定する属性を含めることができます。

外部サーバのユーザグループの設定は、個々のサーバ製品の設計により異なります。該当するサーバ製品のマニュアルを参照してください。

タスクグループ

タスクグループは、アクションのタイプ（read、writeなど）に対応した許容タスクIDのリストによって定義します。タスクIDは、ルータシステムで基本的に定義されます。外部ソフトウェアのタスクグループを設定するには、タスクID定義が事前にサポートされている必要があります。

タスクIDは、外部TACACS+サーバまたはRADIUSサーバで設定できます。

AAA 設定

ここでは、AAA設定についての情報を提供します。

方式リスト

AAAデータは、さまざまなデータソースに保存できます。AAA設定は、方式リストを使用して、AAAデータのソースの優先順位を定義します。AAAは、複数の方式リストを定義でき、アプリケーション（ログインなど）は、これらのいずれかを選択できます。たとえば、コンソールポートとVTYポートとで異なる方式リストを使用できます。方式リストが指定されていない場合、アプリケーションは、デフォルトの方式リストを使用します。デフォルトの方式リストが存在しない場合、AAAは、ローカルデータベースとしてソースを使用します。

ロールオーバーメカニズム

AAAは、データベースオプションの優先順位リストを使用するように設定できます。システムがデータベースを使用できない場合、リストの次のデータベースに自動的にロールオーバーします。認証、認可またはアカウントिंग要件がデータベースで拒否されると、ロールオーバーは発生せず、要求が拒否されます。

次の方法が選択可能です。

- **Local** : ローカルで設定されるデータベースを使用します（アカウントングや一部の認可には適していません）。
- **TACACS+** : TACACS+サーバ（CiscoSecure ACSなど）を使用します。
- **RADIUS** : RADIUSサーバを使用します。
- **Line** : 回線パスワードおよびユーザグループを使用します（認証のみに適しています）。

- **None** : 要求を許可します (認証には適していません)。

サーバのグループ化

サーバの1つのグローバルリストを保持する代わりに、さまざまなAAAプロトコル (RADIUS や TACACS+ など) 用のサーバグループを作成し、AAA アプリケーション (PPP や XR EXEC モードなど) に関連付けることができます。

認証

認証は、プリンシパル (ユーザまたはアプリケーション) がシステムへのアクセスを取得する最も重要なセキュリティプロセスです。プリンシパルは、管理ドメインで一意であるユーザ名 (またはユーザ ID) により定義されます。ユーザにサービスを提供するアプリケーション (または管理エージェントなど) は、ユーザからユーザ名およびクレデンシャルを取得します。AAA は、アプリケーションにより渡されたユーザ名およびクレデンシャルに基づいて認証を実行します。認証ユーザのロールは、ユーザが属する1つ以上のグループにより決まります (ユーザは、1つ以上のユーザグループのメンバにすることができます)。

所有者以外のセキュア ドメイン ルータ ユーザの認証

所有者以外のセキュア ドメイン ルータにログインする場合、ルート システム ユーザは、「@admin」サフィクスをユーザ名に追加する必要があります。「@admin」サフィクスを使用すると、認証要求が所有者のセキュア ドメイン ルータに送信され、確認されます。所有者のセキュア ドメイン ルータは、認証方式の選択にリスト名 **remote** を使用します。**remote** 方式リストは、**aaa authentication login remote method1 method2 ...** コマンドを使用して設定されます。

所有者のセキュア ドメイン ルータ ユーザの認証

所有者のセキュア ドメイン ルータ ユーザは、所有者のセキュア ドメイン ルータ ユーザに関連付けられている特定のセキュア ドメイン ルータに属するノードだけにログインできます。ユーザが **root-sdr** グループのメンバである場合、ユーザは、所有者のセキュア ドメイン ルータ ユーザとして認証されます。

セキュア ドメイン ルータ ユーザの認証

セキュア ドメイン ルータ ユーザの認証は、所有者のセキュア ドメイン ルータ ユーザの認証と似ています。あるユーザが、指定された所有者のセキュア ドメイン ルータ ユーザグループのメンバーであると判明しなかった場合、そのユーザはセキュア ドメイン ルータ ユーザとして認証されます。

認証フロー制御

AAA は、次のプロセスに従い認証を実行します。

1. ユーザが、ユーザ名およびパスワード (またはシークレット) を提供して認証を要求します。
2. AAA が、ユーザのパスワードを検証して、パスワードがデータベースのものと一致しない場合ユーザを拒否します。
3. AAA が、ユーザのロールを決定します (ルート SDR ユーザまたは SDR ユーザ)。

- ユーザが所有者のセキュア ドメイン ルータ ユーザ グループのメンバとして設定されている場合、AAA は、そのユーザを所有者のセキュア ドメイン ルータ ユーザとして認証します。
- ユーザがある所有者のセキュア ドメイン ルータ ユーザ グループのメンバーとして設定されていない場合、AAA は、そのユーザを所有者のセキュア ドメイン ルータ ユーザとして認証します。

クライアントは、ユーザの許可されているタスク ID を認証中に取得できます。この情報は、ユーザが属するユーザ グループで指定されているすべてのタスク グループ定義の集合を形成することで取得されます。このような情報を使用するクライアントは、通常、タスク ID セットが静的であるユーザのセッション（API セッションなど）を作成します。XR EXEC モードおよび外部 API クライアントは、どちらもこの機能を使用して操作を最適化できます。XR EXEC モードは該当しないコマンドを非表示にでき、EMS アプリケーションは、たとえば、該当しないグラフィカル ユーザ インターフェイス（GUI）メニューを無効にできます。

ユーザ グループ メンバーシップなどのユーザの属性やタスク権限が変更されると、これらの変更された属性は、ユーザの現在アクティブなセッションでは反映されません。これらは、ユーザの次のセッションで有効になります。

パスワードタイプ

ユーザおよびそのユーザのグループ メンバーシップを設定する場合、暗号化またはクリア テキストの 2 つのパスワードを指定できます。

ルータは、二方向および一方向（シークレット）の両方の暗号化ユーザパスワードをサポートします。オリジナルの暗号化されていないパスワード文字列が暗号化シークレットからは推測できないため、シークレットパスワードはユーザ ログイン アカウントに適しています。アプリケーションによっては（PPP など）、パケットでのパスワードの送信など、独自の機能のための保存パスワードを復号化する必要があるため、二方向のみのパスワードが必要です。ログインユーザでは、両方のタイプのパスワードを設定できますが、一方のパスワードがすでに設定されている状態でももう一方のパスワードを設定すると、警告メッセージが表示されます。

シークレットとパスワードの両方をユーザに設定すると、ログインなど、復号化できるパスワードを必要としないすべての操作で、シークレットが優先されます。PPP などのアプリケーションでは、シークレットが存在する場合でも、二方向の暗号化パスワードが使用されます。

タスクベースの認可

AAA は、CLI または API を介した操作の任意の制御、設定またはモニタに「タスク許可」を使用します。特権レベルに関する Cisco IOS ソフトウェアの概念は、ソフトウェアではタスクベースの認可システムに置き換えられています。

タスク ID

ユーザによる Cisco ソフトウェアの制御、設定、モニタを可能にする操作タスクは、タスク ID 別に示されます。タスク ID は、コマンドで操作をする許可を定義します。ユーザには、ルータに許可されているアクセスの範囲を定義するタスク ID のセットが関連付けられます。

タスク ID は、次のようにしてユーザに割り当てられます。

各ユーザは、1つの以上のユーザグループに関連付けられます。各ユーザグループは1つ以上のタスクグループに関連付けられます。次に、各タスクグループは、一連のタスク ID によって定義されます。つまり、ユーザと特定のユーザグループを関連付けることで、そのユーザとタスク ID の特定のセットが関連付けられます。タスク ID に関連付けられたユーザは、そのタスク ID に関連する処理を実行できます。

タスク ID に関する一般的な使用上のガイドライン

大部分のルータ制御、設定、モニタリング操作（CLI、Netconf、Restconf、XML API）には、特定のタスク ID セットが関連付けられています。通常、特定の CLI コマンドまたは API インベーションは、1つ以上のタスク ID が関連付けられます。config および commit コマンドでは、特定のタスク ID 許可は必要ありません。設定およびコミット操作では、特定のタスク ID 許可は必要ありません。エイリアスでもタスク ID 許可は必要ありません。コンフィギュレーション交換は、root-lr 許可が割り当てられるまで実行できません。コンフィギュレーションモードを開始しない場合、TACACS+ コマンド認可を使用して、config コマンドを拒否できます。これらの関連付けは、ルータ内でハードコード化されていて、変更できません。タスク ID は、特定のタスクを実行する許可を付与します。タスク ID では、タスクを実行する許可は拒否されません。タスク ID 操作は、次の表にリストされているクラスの 1 つ、すべて、または任意の組み合わせにすることができます。



(注) Restconf は今後のリリースでサポートされる予定です。

表 1: タスク ID クラス

動作	説明
Read	読み取り専用操作を許可します。
Write	変更操作を許可、および読み取り操作を暗黙的に許可します。
Execute	ping や Telnet などのアクセス操作を許可します。
Debug	デバッグ操作を許可します。

システムは、各 CLI コマンドおよび API インベーションがユーザのタスク ID 許可リストと一致しているか検証します。CLI コマンドの使用時に問題が発生した場合、システム管理者に連絡してください。

スラッシュで区切られた複数のタスク ID 操作（read/write など）は、両方の操作が指定のタスク ID に適用されることを示します。

カンマで区切られた複数のタスク ID 操作（read/write, execute など）は、両方の操作が個々のタスク ID に適用されることを示します。たとえば、**copy ipv4 access-list** コマンドを使用すると、読み取りおよび書き込み操作を acl タスク ID に適用し、実行操作を filesystem タスク ID に適用できます。

タスク ID と操作の列に値が指定されていない場合は、このコマンドの使用に際して、タスク ID および操作との以前の関連付けは考慮されません。また、ROM モニタ コマンドを使用するために、ユーザにタスク ID を関連付ける必要はありません。

コマンドが特定のコンフィギュレーションサブモードで使用される場合、そのコマンドを使用するための追加タスク ID をユーザに関連付ける必要があります。たとえば、**show redundancy** コマンドを実行するには、システム (read) タスク ID と操作をユーザに関連付ける必要があります (次の例を参照)。

```
RP/0/RP0/CPU0:router# show redundancy
```

TACACS+ および RADIUS 認証ユーザのタスク ID

Cisco ソフトウェアの AAA では、TACACS+ および RADIUS 方式で認証されるユーザに次の方法でタスク許可を割り当てることができます。

- タスクマップのテキストバージョンを、外部 TACACS+ および RADIUS サーバのコンフィギュレーションファイルに直接指定します。
- 外部 TACACS+ および RADIUS サーバのコンフィギュレーションファイルで特権レベルを指定します。
- TACACS+ および RADIUS 方式で認証するユーザと同じユーザ名でローカルユーザを作成します。
- 許可が TACACS+ および RADIUS 方式で認証する任意のユーザに適用されるデフォルトタスクグループを設定別に指定します。

特権レベルマッピング

タスク ID の概念をサポートしない TACACS+ デモンとの互換性のため、AAA は外部 TACACS+ サーバ設定ファイル内でユーザに定義されている特権レベルとローカルユーザグループとの間のマッピングをサポートします。TACACS+ 認証後、外部 TACACS+ サーバから返される権限レベルからマッピングされたユーザグループのタスクマップがユーザに割り当てられます。たとえば、特権レベル 5 が外部 TACACS+ サーバから返された場合、AAA はローカルユーザグループ `priv5` のタスクマップを取得することを試みます。このマッピングプロセスは、1 ~ 13 の他の特権レベルでも同様です。特権レベル 14 は、ユーザグループオーナー SDR にマッピングされます。

たとえば、Cisco のフリーウェア TACACS+ サーバでは、コンフィギュレーションファイルは、次の例に示すようにコンフィギュレーションファイルで `priv_lvl` を指定する必要があります。

```
user = sampleuser1{
  member = bar
  service = exec-ext {
    priv_lvl = 5
  }
}
```

この例の 5 という数値は、ユーザの *sampleuser* に割り当てる必要がある任意の特権レベルに置き換えることができます。

AAA サービスの XML スキーマ

Extensible Markup Language (XML) インターフェイスは、XML ドキュメント形式で要求と応答を使用して、AAA を設定およびモニタします。AAA コンポーネントは、設定およびモニタリングに使用されるデータの内容と構造に対応する XML スキーマを発行します。XML ツールおよびアプリケーションは、このスキーマを使用して、XML エージェントと通信して設定を実行します。

次のスキーマは、AAA を使用して発行されます。

- 認証、認可、アカウント設定
- ユーザ、ユーザ グループおよびタスク グループ設定
- TACACS+ サーバおよびサーバ グループ設定
- RADIUS サーバおよびサーバ グループ設定

AAA サービスの *Netconf* および *Restconf*

XML スキーマと同様、*Netconf* および *Restconf* では、ユーザ名とパスワードはローカルサービスまたはトリプル A (AAA) サービスによって制御されます。



(注) *Restconf* は今後のリリースでサポートされる予定です。

RADIUS について

RADIUS は、不正なアクセスからネットワークのセキュリティを保護する分散クライアント/サーバシステムです。シスコの実装では、RADIUS クライアントは Cisco ルータ上で稼働します。認証要求とアカウント要求は、すべてのユーザ認証情報とネットワーク サービスアクセス情報が格納されている中央の RADIUS サーバに送信されます。

RADIUS は完全にオープンなプロトコルであり、ソースコード形式で配布されているため、現在使用できる任意のセキュリティシステムと連携するように変更できます。

シスコは、AAA セキュリティ パラダイムの下で RADIUS をサポートしています。RADIUS は、TACACS+、Kerberos、ローカルユーザ名の検索など、他の AAA セキュリティ プロトコルと併用できます。



(注) RADIUS はすべての Cisco プラットフォームでサポートされますが、RADIUS でサポートされる一部の機能は、指定されたプラットフォームだけで実行されます。

RADIUS は、リモートユーザのネットワーク アクセスを維持すると同時に高度なレベルのセキュリティを必要とするさまざまなネットワーク環境に実装されています。

RADIUS は、アクセスのセキュリティが必要な次のネットワーク環境で使用できます。

- それぞれが RADIUS をサポートする、マルチベンダー アクセス サーバによるネットワーク。たとえば、複数のベンダーのアクセスサーバが、1つの RADIUS サーバベースセキュリティ データベースを使用します。複数ベンダーのアクセス サーバからなる IP ベースのネットワークでは、ダイヤルインユーザは RADIUS サーバを通じて認証されます。RADIUS サーバは、Kerberos セキュリティシステムで動作するようにカスタマイズされています。
- アプリケーションが RADIUS プロトコルをサポートするターンキー ネットワーク セキュリティ環境。たとえば、「スマートカード」アクセス コントロール システムを使用するアクセス環境。ある事例では、RADIUS と Enigma のセキュリティカードを併用してユーザを検証し、ネットワーク リソースに対するアクセス権を付与しています。
- すでに RADIUS を使用中のネットワーク。RADIUS 機能を持つ Cisco ルータをネットワークに追加できます。Terminal Access Controller Access Control System Plus (TACACS+) サーバに移行する場合、これが最初の手順となります。
- ユーザが単一のサービスにだけアクセスする必要があるネットワーク。RADIUS を使用すると、単一ホスト、単一ユーティリティ (Telnet など)、または単一プロトコル (ポイントツーポイント プロトコル (PPP)) に対するユーザアクセスを制御できます。たとえば、ユーザがログインすると、RADIUS は、IP アドレス 10.2.3.4 を使用してそのユーザが PPP を実行する権限を持っていることを識別し、定義済みのアクセスリストが開始されます。
- リソースアカウンティングが必要なネットワーク。RADIUS アカウンティングは、RADIUS 認証または RADIUS 認可とは個別に使用できます。RADIUS アカウンティング機能を使用すると、サービスの開始および終了時に、セッション中に使用したリソース (時間、パケット、バイトなど) の量を示すデータを送信できます。インターネット サービス プロバイダー (ISP) は、RADIUS アクセス コントロールおよびアカウンティング ソフトウェアのフリーウェアバージョンを使用して、セキュリティおよび課金の独自ニーズを満たすこともできます。
- 事前認証をサポートしているネットワーク。ネットワークに RADIUS サーバを導入すると、AAA 事前認証を設定し、事前認証のプロファイルを設定できます。サービス プロバイダーが事前認証を使用すると、既存の RADIUS ソリューションを使用するポートの管理性が向上し、共有リソースを効率的に管理して、各種のサービスレベル契約を提供できるようになります。

RADIUS が適さないネットワーク セキュリティ状況

RADIUS は次のネットワーク セキュリティ状況には適していません。

- マルチプロトコル アクセス環境。RADIUS は次のプロトコルをサポートしていません。
 - NetBIOS Frame Control Protocol (NBFCP)
 - NetWare Asynchronous Services Interface (NASI)
 - X.25 PAD 接続

- ルータ間で接続している環境。RADIUS は、双方向認証を行いません。RADIUS は、ルータと RADIUS 認証を必要とするシスコ製以外のルータとの認証に使用できます。
- 各種のサービスを使用するネットワーク。RADIUS は、一般に 1 人のユーザを 1 つのサービス モデルにバインドします。

RADIUS の動作

ユーザがログインを試行し、RADIUS を使用してアクセスサーバから認証を受ける場合、次の手順が発生します。

1. ユーザが、ユーザ名とパスワードの入力を求められ、入力します。
2. ユーザ名および暗号化されたパスワードが、ネットワーク経由で RADIUS サーバに送信されます。
3. ユーザは、RADIUS サーバから次のいずれかの応答を受信します。
 1. ACCEPT : ユーザが認証されたことを表します。
 1. REJECT : ユーザは認証されず、ユーザ名とパスワードの再入力を求められるか、アクセスを拒否されます。
 1. CHALLENGE : RADIUS サーバによってチャレンジが発行されます。チャレンジは、ユーザから追加データを収集します。
 1. CHANGE PASSWORD : RADIUS サーバからユーザに対して新しいパスワードの選択を求める要求が発行されます。

ACCEPT または REJECT 応答には、XR EXEC モードまたはネットワーク認可に使用される追加データが含まれています。RADIUS 認可を使用するには、まず RADIUS 認証を完了する必要があります。ACCEPT または REJECT パケットに含まれる追加データの内容は次のとおりです。

- ユーザがアクセスできるサービス。これには、Telnet、rlogin、ローカルエリア トランスポート (LAT) の各接続、および PPP、Serial Line Internet Protocol (SLIP)、XR EXEC モードの各サービスなどが該当します。
- ホストまたはクライアントの IP アドレス、アクセスリスト、ユーザタイムアウトなどの接続パラメータ。



第 2 章

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 \(41 ページ\)](#)

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ールエンベロープ (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。

- 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/CPU0: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/CPU0: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。

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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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/CPU0: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入力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/CPU0: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に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 \(51 ページ\)](#) 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：	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

	コマンドまたはアクション	目的
	<pre>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</pre>	

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 : <pre>RP/0/RP0/CPU0:IMC0 (config)#crypto ca trustpool policy RP/0/RSP0/CPU0:IMC0 (config-trustpool)#</pre>	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。

	コマンドまたはアクション	目的
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。

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 の公開キーを認識する必要があります。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



第 3 章

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 \(57 ページ\)](#)

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/CPU0:router(config)# key chain isis-keys
RP/0/RP0/CPU0: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せん。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/CPU0: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 mykey9labcd
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/RP0/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/RP0/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/RP0/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/RP0/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/RP0/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/RP0/CPU0:router(config-isis-keys)# key 8  
RP/0/RP0/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/RP0/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/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/RP0/CPU0:router(config-isis-keys)# key 8  
RP/0/RP0/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 **accept-lifetime** *start-time* [**duration** *duration-value* | **infinite** | *end-time*]

例：

```
RP/0/RP0/CPU0:router(config-isis-keys)# key 8  
RP/0/RP0/CPU0:router(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/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/RP0/CPU0:router(config-isis-keys)# key 8
RP/0/RP0/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 send-lifetime *start-time* [**duration** *duration-value* | **infinite** | *end-time*]

例：

```
RP/0/RP0/CPU0:router(config-isis-keys)#key 8
RP/0/RP0/CPU0:router(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/CPU0:router(config)# key chain isis-keys  
RP/0/RP0/CPU0: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/CPU0:router(config-isis-keys)# key 8  
RP/0/RP0/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 `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/CPU0: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。



第 4 章

URPF の実装

ここでは、URPF の実装について説明します。

- [URPF について \(65 ページ\)](#)
- [URPF ルーズモードの設定 \(65 ページ\)](#)

URPF について

偽装した IP アドレスを使用し（この手口は IP アドレススプーフィングと呼ばれます）、サービスプロバイダーによる検出を回避するために送信元 IP アドレスを頻繁に変更することは、DoS 攻撃を計画するハッカーの常套手段となっています。

Unicast Reverse Path Forwarding (URPF) は、ルータで受信されたパケットの送信元 IP アドレスを検証するためのメカニズムです。URPF を設定されたルータは、FIB テーブルでリバースパスルックアップを実行して、送信元 IP アドレスの存在を確認します。送信元 IP アドレスがテーブルにリストされていれば、送信元は到達可能かつ有効です。送信元 IP アドレスが FIB テーブルにない場合、パケットはルータによって悪意のあるものとして扱われ、廃棄されます。

ルータでは、ルーズモードでの URPF の使用がサポートされています。URPF ルーズモードは、ルータが FIB 内の送信元 IP アドレスのプレフィックスのみを検証し、パケットがルータに到達するために使用するインターフェイスは検証しないように設定されている場合に有効となっています。ルーズモードを設定することで、代替インターフェイスを使用してルータに到達する正当なトラフィックが悪意のあるものと誤解されることがなくなります。URPF ルーズモードは、マルチホームプロバイダーエッジネットワークで非常に役立ちます。

URPF ルーズモードの設定

ここでは、IPv4 と IPv6 の両方のネットワークに対し、ルータに URPF ルーズモードを設定する方法について説明します。

はじめる前に

ルータにURPFルーズモードを設定する前に、ここで説明するように、ラインカードのデフォルトスケールを無効にする必要があります。



- (注) IPv6 uRPF 設定では、すべてのタイプのカード (TCAM カードと非 TCAM カードの両方) に対し、**hw-module fib ipv6 scale internet-optimized-disable** コマンドが必要です。デフォルトでは、IPv6 はプレフィックスに内部メモリを使用します。したがって、**hw-module fib ipv6 scale internet-optimized-disable** コマンドを設定してから、ラインカードをリロードする必要があります。



- (注) ラインカードは、デフォルトスケールを無効にした後でリロードする必要があります。このようにするのは、**hw-module** コマンドの設定がすぐに有効になるようにするためです。



- (注) 外部 TCAM (eTCAM) システムを使用した Jericho+ では、uRPF を有効にするためにデュアルキャパシティモードを無効にする必要はありません。

すべてのタイプの TCAM ラインカードの場合 :

```
RP/0/RP0/CPU0:router(config)# hw-module tcam fib ipv4 scaledisable
RP/0/RP0/CPU0:router(config)# hw-module fib ipv6 scale internet-optimized-disable
RP/0/RP0/CPU0:router(config)# commit
RP/0/RP0/CPU0:router(config)# end
RP/0/RP0/CPU0:router# reload location all
Proceed with reload? [confirm]
```

すべてのタイプの非 TCAM ラインカードの場合 :

```
RP/0/RP0/CPU0:router(config)# hw-module fib ipv4 scale host-optimized-disable
RP/0/RP0/CPU0:router(config)# hw-module fib ipv6 scale internet-optimized-disable
RP/0/RP0/CPU0:router(config)# commit
RP/0/RP0/CPU0:router(config)# end
RP/0/RP0/CPU0:router# reload location all
Proceed with reload? [confirm]
```

設定

ルータにURPFルーズモードを設定するには、次の設定を使用します。



- (注) URPF を動作させるには、(この項で説明するとおり) IPv4 コマンドと IPv6 コマンドの両方を設定する必要があります。


```
RP/0/RP0/CPU0:router(config)# interface bundle-ether1
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.0.0.1 255.255.255.0
RP/0/RP0/CPU0:router(config-if)# ipv4 verify unicast source reachable-via any
RP/0/RP0/CPU0:router(config-if)# ipv6 address 2001::1/64
RP/0/RP0/CPU0:router(config-if)# ipv6 verify unicast source reachable-via any
RP/0/RP0/CPU0:router(config-if)# commit
```

実行コンフィギュレーション

次のように設定されていることを確認します。

```
RP/0/RP0/CPU0:router(config-if)# show running-config
Thu Jul 27 14:40:38.167 IST
...
!
interface Bundle-Ether1
  ipv4 address 10.0.0.1 255.255.255.0
  ipv4 verify unicast source reachable-via any
  ipv6 address 2001::1/64
  ipv6 verify unicast source reachable-via any
!
```

これで、ルータに URPF ルーズ モードが正常に設定されました。



第 5 章

管理プレーン保護の実装

管理プレーン保護（MPP）機能では、ネットワーク管理パケットのデバイスへの着信を許可するインターフェイスを制限できます。ネットワーク オペレータは MPP 機能を使用して、1つ以上のルータ インターフェイスを管理インターフェイスとして指定できます。

MPP 保護機能は、MPP 配下のすべての管理プロトコルと同様、デフォルトではディセーブルになっています。インターフェイスをアウトオブバンドまたはインバンドとして設定すると、MPP が自動的に有効になります。これにより、MPP 配下のすべてのプロトコルもイネーブルになります。MPP がディセーブルでプロトコルがアクティブな場合、トラフィックはすべてのインターフェイスを通過できます。

アクティブなプロトコルが存在する状態で MPP がイネーブルになると、管理トラフィックを許可するデフォルトの管理インターフェイスはルートプロセッサ（RP）およびスタンバイルートプロセッサ（SRP）のイーサネット インターフェイスのみになります。MPP をイネーブルにする他のすべてのインターフェイスについては、手動で管理インターフェイスとして設定する必要があります。

以後は、デフォルト管理インターフェイスと事前に MPP インターフェイスとして設定したインターフェイスのみがデバイス宛のネットワーク管理パケットを受け付けます。他のすべてのインターフェイスは、デバイス宛のネットワーク管理パケットをドロップします。論理インターフェイス（またはデータプレーンに存在しない他のすべてのインターフェイス）は、入力物理インターフェイスに基づいてパケットをフィルタリングします。

- [管理プレーン保護の利点（69 ページ）](#)
- [管理プレーン保護の実装に関する制約事項（70 ページ）](#)
- [インバンド インターフェイスの管理プレーン保護のデバイスの設定（70 ページ）](#)
- [アウトオブバンド インターフェイスの管理プレーン保護のデバイスの設定（73 ページ）](#)
- [管理プレーン保護の実装について（77 ページ）](#)

管理プレーン保護の利点

MPP 機能を設定すると、次の利点があります。

- すべてのインターフェイスで管理プロトコルを許可することを超える、デバイスの管理目的でのアクセス制御。

- 非管理インターフェイスでのデータ パケットのパフォーマンス向上。
- ネットワークの拡張性のサポート。
- インターフェイス単位のアクセス コントロール リスト (ACL) を使用することによる、デバイスへの管理アクセス制限の作業の簡易化。
- デバイスへのアクセスを制限するために必要な ACL 数の削減。
- スwitチングインターフェイスおよびルーティングインターフェイス上でパケットフラッディングの CPU への到達を防止。

管理プレーン保護の実装に関する制約事項

管理プレーン保護 (MPP) の実装には次の制約事項があります。

- 現在、MPP は拒否またはドロップされたプロトコル要求を追跡していません。
- MPP 設定では、プロトコル サービスをイネーブルにはできません。MPP はさまざまなインターフェイスでサービスを利用可能にする役割のみを果たします。プロトコルは明示的にイネーブル化されます。
- インバンドインターフェイスで受信する管理要求は、その場で必ずしも認知されるわけではありません。
- MPP 設定に加えた変更は、その変更よりも前に確立されているアクティブなセッションには影響を与えません。
- 現在、MPP は、TFTP、Telnet、簡易ネットワーク管理プロトコル (SNMP)、セキュアシェル (SSH)、XML、Netconf などのプロトコルに対して着信する管理要求のみを制御します。
- MPP は MIB をサポートしていません。

インバンドインターフェイスの管理プレーン保護のデバイスの設定

インバンド管理インターフェイスは、データ転送パケットだけでなく管理パケットも処理する、物理インターフェイスまたは論理インターフェイスです。インバンド管理インターフェイスは、共有管理インターフェイスとも呼ばれています。ネットワークに追加した直後のデバイスや、ネットワークですでに動作しているデバイスを設定するには、この作業を実行します。この作業では、特定のインターフェイスを通じてのみ Telnet のルータへのアクセスが許可されるインバンドインターフェイスとして、MPP を設定する方法について説明します。

デフォルトでない VRF でインバンド MPP インターフェイスを設定するには、次の作業を追加で実行します。

- デフォルトでないインバンド VRF のインターフェイスを設定します。
- グローバル インバンド VRF を設定します。
- Telnet の場合は、インバンド VRF に対して Telnet VRF サーバを設定します。

手順

ステップ 1 configure

ステップ 2 control-plane

例 :

```
RP/0/RP0/CPU0:router(config)# control-plane
RP/0/RP0/CPU0:router(config-ctrl)#
```

コントロールプレーン コンフィギュレーション モードを開始します。

ステップ 3 management-plane

例 :

```
RP/0/RP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0/CPU0:router(config-mpp)#
```

管理プレーン保護を設定してプロトコルを許可および拒否し、管理プレーン保護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 inband

例 :

```
RP/0/RP0/CPU0:router(config-mpp)# inband
RP/0/RP0/CPU0:router(config-mpp-inband)#
```

インバンドインターフェイスを設定し、管理プレーン保護インバンド 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 interface {type instance | all}

例 :

```
RP/0/RP0/CPU0:router(config-mpp-inband)# interface HundredGigE 0/9/0/0
RP/0/RP0/CPU0:router(config-mpp-inband-Gi0_0_1_0)#
```

特定のインバンド インターフェイスを設定するか、すべてのインバンド インターフェイスを設定します。管理プレーン保護インバンド インターフェイス コンフィギュレーション モードを開始するには、**interface** コマンドを使用します。

- **all** キーワードを使用して、すべてのインターフェイスを設定します。

ステップ 6 **allow** {*protocol* | **all**} [**peer**]

例 :

```
RP/0/RP0/CPU0:router(config-mpp-inband-Gi0_0_1_0)# allow Telnet peer
RP/0/RP0/CPU0:router(config-telnet-peer)#
```

指定されたプロトコルまたはすべてのプロトコルに対するインバンド インターフェイスとして、インターフェイスを設定します。

- *protocol* 引数を使用して、指定管理インターフェイスで管理プロトコルを許可します。
 - SNMP (バージョンも)
 - セキュア シェル (v1 および v2)
 - TFTP
 - Telnet
 - Netconf
 - XML
- **all** キーワードを使用して、プロトコルのリストで指定されるすべての管理トラフィックを許可するようにインターフェイスを設定します。
- (任意) **peer** キーワードを使用して、インターフェイスでピア アドレスを設定します。

ステップ 7 **address ipv4** {*peer-ip-address* | *peer ip-address/length*}

例 :

```
RP/0/RP0/CPU0:router(config-telnet-peer)# address ipv4 10.1.0.0/16
```

このインターフェイス上で管理トラフィックが許可されるピア IPv4 アドレスを設定します。

- *peer-ip-address* 引数を使用して、このインターフェイス上で管理トラフィックが許可されるピア IPv4 アドレスを設定します。
- *peer ip-address/length* 引数を使用して、ピア IPv4 アドレスのプレフィックスを設定します。

ステップ 8 **commit**

ステップ 9 **show mgmt-plane** [**inband** |] [**interface** {*type instance*}]

例：

```
RP/0/RP0/CPU0:router# show mgmt-plane inband interface HundredGigE 0/9/0/0
```

インターフェイスのタイプやインターフェイスでイネーブルにされるプロトコルなど、管理プレーンに関する情報を表示します。

- (任意) **inband** キーワードを使用して、管理パケットおよびデータ転送パケットを処理するインターフェイスであるインバンド管理インターフェイスの設定を表示します。
- (任意) **interface** キーワードを使用して、特定のインターフェイスの詳細を表示します。

アウトオブバンドインターフェイスの管理プレーン保護のデバイスの設定

アウトオブバンドは、管理プロトコルトラフィックの転送または処理だけを許可するインターフェイスを意味します。アウトオブバンド管理インターフェイスは、ネットワーク管理トラフィックだけを受信するようネットワークオペレータによって定義されます。転送（またはカスタマー）トラフィックの利点は、ルータの管理が妨害されないことであり、これにより、サービス拒否攻撃の可能性が大幅に低減します。

アウトオブバンドインターフェイスは、アウトオブバンドインターフェイス間のトラフィックのみを転送するか、ルータ宛の管理パケットを終端します。また、アウトオブバンドインターフェイスをダイナミックルーティングプロトコルに加えることができます。サービスプロバイダーはルータのアウトオブバンドインターフェイスに接続し、ルータが提供可能なすべてのルーティングツールおよびポリシーツールを使用して、独立したオーバーレイ管理ネットワークを構築します。

アウトオブバンド MPP インターフェイスを設定するには、次の作業を実行します。

- アウトオブバンド VRF のインターフェイスを設定します。
- グローバル アウトオブバンド VRF を設定します。
- Telnet の場合は、アウトオブバンド VRF に対して Telnet VRF サーバを設定します。

手順

ステップ 1 **configure**

ステップ 2 **control-plane**

例：

```
RP/0/RP0/CPU0:router(config)# control-plane
```

```
RP/0/RP0/CPU0:router(config-ctrl)#
```

コントロールプレーン コンフィギュレーション モードを開始します。

ステップ3 management-plane

例：

```
RP/0/RP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0/CPU0:router(config-mpp)#
```

管理プレーン保護を設定してプロトコルを許可および拒否し、管理プレーン保護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 out-of-band

例：

```
RP/0/RP0/CPU0:router(config-mpp)# out-of-band
RP/0/RP0/CPU0:router(config-mpp-outband)#
```

帯域外インターフェイスまたはプロトコルを設定し、管理プレーン保護帯域外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 vrf vrf-name

例：

```
RP/0/RP0/CPU0:router(config-mpp-outband)# vrf target
```

帯域外インターフェイスのバーチャルプライベートネットワーク (VPN) Routing and Forwarding (VRF; VPN ルーティングおよび転送) リファレンスを設定します。

- *vrf-name* 引数を使用して、VRF に名前を割り当てます。

ステップ6 interface {type instance | all}

例：

```
RP/0/RP0/CPU0:router(config-mpp-outband)# interface HundredGigE 0/9/0/0
RP/0/RP0/CPU0:router(config-mpp-outband-if)#
```

特定のアウトオブバンドインターフェイス、またはすべてのアウトオブバンドインターフェイスをアウトオブバンドインターフェイスとして設定します。管理プレーン保護アウトオブバンド コンフィギュレーションモードを開始するには、**interface** コマンドを使用します。

- **all** キーワードを使用して、すべてのインターフェイスを設定します。

ステップ 7 `allow {protocol | all} [peer]`

例 :

```
RP/0/RP0/CPU0:router(config-mpp-outband-if)# allow TFTP peer
RP/0/RP0/CPU0:router(config-tftp-peer)#
```

指定されたプロトコルまたはすべてのプロトコルに対するアウトオブバンドインターフェイスとして、インターフェイスを設定します。

- *protocol* 引数を使用して、指定管理インターフェイスで管理プロトコルを許可します。
 - HTTP または HTTPS
 - SNMP (バージョンも)
 - セキュア シェル (v1 および v2)
 - TFTP
 - Telnet
 - Netconf
- **all** キーワードを使用して、プロトコルのリストで指定されるすべての管理トラフィックを許可するようにインターフェイスを設定します。
- (任意) **peer** キーワードを使用して、インターフェイスでピア アドレスを設定します。

ステップ 8 `address ipv6 {peer-ip-address | peer ip-address/length}`

例 :

```
RP/0/RP0/CPU0:router(config-tftp-peer)# address ipv6 33::33
```

このインターフェイス上で管理トラフィックが許可されるピア IPv6 アドレスを設定します。

- *peer-ip-address* 引数を使用して、このインターフェイス上で管理トラフィックが許可されるピア IPv6 アドレスを設定します。
- *peer ip-address/length* 引数を使用して、ピア IPv6 アドレスのプレフィックスを設定します。

ステップ 9 `commit`**ステップ 10** `show mgmt-plane [inband | out-of-band] [interface {type instance} | vrf]`

例 :

```
RP/0/RP0/CPU0:router# show mgmt-plane out-of-band interface HundredGigE 0/9/0/0
```

インターフェイスのタイプやインターフェイスでイネーブルにされるプロトコルなど、管理プレーンに関する情報を表示します。

- (任意) **inband** キーワードを使用して、管理パケットおよびデータ転送パケットを処理するインターフェイスであるインバンド管理インターフェイスの設定を表示します。
- (任意) **out-of-band** キーワードを使用して、アウトオブバンドインターフェイス設定を表示します。
- (任意) **interface** キーワードを使用して、特定のインターフェイスの詳細を表示します。
- (任意) **vrf** キーワードを使用して、アウトオブバンドインターフェイスのバーチャルプライベート ネットワーク (VPN) ルーティングおよび転送リファレンスを表示します。

例

次に、MMP 下での特定の IP アドレスに対するインバンドおよびアウトオブバンドインターフェイスを設定する例を示します。

```
configure
control-plane
management-plane
inband
interface all
allow SSH
!
interface HundredGigE 0/9/0/0
allow all
allow SSH
allow Telnet peer
address ipv4 10.1.0.0/16
!
!
interface HundredGigE 0/9/0/0
allow Telnet peer
address ipv4 10.1.0.0/16
!
!
out-of-band
vrf my_out_of_band
interface HundredGigE 0/9/0/0
allow TFTP peer
address ipv6 33::33
!
!
!
!
show mgmt-plane

Management Plane Protection

inband interfaces
```

```
-----  
interface - HundredGigE0_9_0_0  
  ssh configured -  
    All peers allowed  
  telnet configured -  
    peer v4 allowed - 10.1.0.0/16  
  all configured -  
    All peers allowed  
interface - HundredGigE0_9_0_0  
  telnet configured -  
    peer v4 allowed - 10.1.0.0/16  
  
interface - all  
  all configured -  
    All peers allowed  
  
outband interfaces  
-----  
interface - HundredGigE0_9_0_0  
  tftp configured -  
    peer v6 allowed - 33::33  
  
show mgmt-plane out-of-band vrf  
  
Management Plane Protection -  
  out-of-band VRF - my_out_of_band
```

管理プレーン保護の実装について

管理プレーン保護機能をイネーブルにする前に、次の概念について理解しておく必要があります。

インターフェイス上のピア フィルタリング

ピア フィルタリング オプションでは、特定のピアまたはピア範囲からの管理トラフィックの設定を許可します。

コントロールプレーン保護

コントロールプレーンは、ルートプロセッサ上のプロセスレベルで稼働するプロセスの集合であり、ほとんどの Cisco ソフトウェアの機能に高レベルの制御を提供します。直接的または間接的にルータを宛先とするすべてのトラフィックは、コントロールプレーンによって処理されます。管理プレーン保護はコントロールプレーンインフラストラクチャ内で動作します。

管理プレーン

管理プレーンは、ルーティングプラットフォームの管理に関連するすべてのトラフィックの論理的なパスです。層およびプレーンで構成される通信アーキテクチャの3つのプレーンの1つである管理プレーンは、ネットワークの管理機能を実行し、すべてのプレーン（管理、制御、

およびデータ)間で機能を調整します。また、管理プレーンはネットワークとの接続を通じてデバイスの管理に使用されます。

管理プレーンで処理されるプロトコルの例は、簡易ネットワーク管理プロトコル (SNMP)、Telnet、SSH、XML および Netconf です。これらの管理プロトコルは、モニタリングやコマンドラインインターフェイス (CLI) のアクセスに使用されます。デバイスに対し、内部送信元 (信頼ネットワーク) へのアクセスを制限することが重要です。



第 6 章

サードパーティ製アプリケーションのための トラフィック保護

サードパーティ製アプリケーションのためのトラフィック保護は、ルータ上の管理トラフィックを保護するためのメカニズムを提供します。サードパーティ製アプリケーションのためのトラフィック保護を使用しないと、サービスが有効になっている場合は Cisco IOS XR により、サービストラフィックはネットワークアドレスを持つすべてのインターフェイスを通過できるようになります。

サードパーティ製アプリケーションのためのトラフィック保護は、LPTS を使用した設定を通じてトラフィックに対してレート制限やスロットリングを行うのに役立ちます。サードパーティ製アプリケーションのためのトラフィック保護では、アドレスファミリー、VRF、ポート、インターフェイス、ローカルアドレス、およびリモートアドレスの各タプルに基づいてトラフィックをフィルタリングします。



(注) アドレスファミリー、プロトコル、ローカルポート、および VRF に加え、少なくとも 1 つのインターフェイスまたはローカルもしくはリモートアドレスを設定する必要があります。

- [gRPC プロトコル \(79 ページ\)](#)
- [サードパーティ製アプリケーションのためのトラフィック保護の制限事項 \(80 ページ\)](#)
- [gRPC を介したサードパーティ製アプリケーションのためのトラフィック保護の前提条件 \(80 ページ\)](#)
- [サードパーティ製アプリケーションのための MPP の設定 \(81 ページ\)](#)
- [サードパーティ製アプリケーションのためのトラフィック保護のトラブルシューティング \(81 ページ\)](#)

gRPC プロトコル

Google 定義されたリモートプロシージャコール (gRPC) は、オープンソースの RPC フレームワークです。これはプロトコルバッファ (Protobuf) に基づいたオープンソースのバイナリシリアル化プロトコルです。gRPC は、XML などの構造化されたデータをシリアル化するため

の柔軟で効率的な自動メカニズムですが、小型で使いやすくなっています。ユーザは、.proto ファイルにプロトコルバッファメッセージタイプを定義することで構造を定義する必要があります。各プロトコルバッファメッセージは、一連の名前と値のペアを含む情報の小型の論理レコードです。

Cisco gRPC インターフェイス定義言語 (IDL) は、一連のサポートされている RPC (get-config、merge-config、replace-config、cli-config、delete-config、cli-show、get-models、action-json、commit、commit-replace など) を使用します。gRPC サーバは Extensible Manageability Services Daemon (EMSD) プロセス内で動作します。gRPC クライアントは任意のマシン上に配置できます。

gRPC は要求および応答をバイナリでエンコードします。gRPC は、Protobuf とともに他のコンテンツタイプに拡張可能です。gRPC の Protobuf バイナリ データ オブジェクトは HTTP/2 を介して転送されます。



- (注) gRPC を有効にする前に、TLS を設定することを推奨します。gRPC プロトコルを有効にすると、TCP で TLS が有効になっていないデフォルトの HTTP/2 トランスポートが使用されます。gRPC では、すべての gRPC 要求に対して AAA 認証および認可が義務付けられています。TLS が設定されていない場合、認証クレデンシャルは暗号化されていないネットワークを介して転送されます。非 TLS モードは、セキュアな内部ネットワークでのみ使用できます。

gRPC はクライアントとサーバ間の分散型のアプリケーションやサービスをサポートします。gRPC はサーバとクライアント間の設定データと運用データを交換するためにデバイス管理サービスを構築するインフラストラクチャを提供します。そのデータの構造は YANG モデルによって定義されます。

サードパーティ製アプリケーションのためのトラフィック保護の制限事項

サードパーティ製アプリケーションのためのトラフィック保護には、以下の制限事項が適用されます。

- TPA エントリがアクティブな RP 管理インターフェイスだけを使用して設定されている場合に冗長スイッチオーバーが実行されると、gRPC 接続が失敗します。

gRPCを介したサードパーティ製アプリケーションのためのトラフィック保護の前提条件

gRPC が設定されていることを確認します。

gRPC の設定

```
Router(config)# grpc port port-number
Router(config)# grpc no-tls
Router(config-grpc)# commit
```

実行コンフィギュレーション

```
Router# show running-config grpc

grpc port 57600
no-tls
!
```

サードパーティ製アプリケーションのための MPP の設定

次のタスクは、サードパーティ製アプリケーションのためのトラフィック保護を設定する方法を示しています。

```
RP/0/0/CPU0:ios#configure
RP/0/0/CPU0:ios(config)#tpa
RP/0/0/CPU0:ios(config-tpa)#vrf default
RP/0/0/CPU0:ios(config-tpa-vrf)#address-family ipv4
RP/0/0/CPU0:ios(config-tpa-vrf-afi)#protection
RP/0/0/CPU0:ios(config-tpa-vrf-afi-prot)#allow protocol tcp local-port port-number
remote-address IP remote address interface interface-name local-address IP local address
```

実行コンフィギュレーション

```
Router# show running-config
tpa
vrf-default
address-family ipv4
protection
allow protocol tcp local-port 57600 remote-address 10.0.0.2/32 local-address 192.168.0.1/32
allow protocol tcp local-port 57600 remote-address 10.0.1.1/24 local-address 192.168.0.1/32
allow protocol tcp local-port 57600 remote-address 10.0.2.3/24 local-address 192.168.0.1/32

address-family ipv6
allow protocol tcp local-port 57600 remote-address 2001:DB8::1/128 local-address
2001:DB8:0:ABCD::1/128
allow protocol tcp local-port 57600 remote-address 2001:DB8::1/128 local-address
2001:DB8:0:ABCD::1/128
allow protocol tcp local-port 57600 remote-address 2001:DB8::1/128 local-address
2001:DB8:0:ABCD::1/128
!
!
```

サードパーティ製アプリケーションのためのトラフィック保護のトラブルシューティング

次の show コマンドの出力は、TPA が設定されているかどうかを確認します。

```
Router# show running-config grpc
```

```
grpc
no-tls
!
```

次の show コマンドの出力は、TPA の設定を表示しています。

```
Router# show running-config tpa
```

```
tpa
vrf default
address-family ipv4
allow local-port 57600 protocol tcp inter mgmtEth 0/RP0/CPU0/0 local-address
192.168.0.1/32 remote-address 10.0.0.2/32
!
```

TPA を使用しない gRPC の設定

```
Router# show kim lpts database
```

```
State:
Prog - Programmed in hardware
Cfg - Configured, not yet programmed
Ovr - Not programmed, overridden by user configuration
Intf - Not programmed, interface does not exist
```

Owner	AF	Proto	State	Interface	VRF	Local ip,port	>	Remote ip,port
Linux	2	6	Prog			global-vrf		any,57600
						> any,0		

```
Router# show lpts bindings brief | include TPA
0/RP0/CPU0 TPA LR IPV4 TCP default any any,57600 any
```

TPA を使用する gRPC の設定

次の show コマンドの出力は、LPTS データベースに設定されている内容を表示しています。また、gRPC の設定がフィルタを使用せずに Linux によって所有されているかどうかも確認します。

```
Router# show kim lpts database
```

```
State:
Prog - Programmed in hardware
Cfg - Configured, not yet programmed
Ovr - Not programmed, overridden by user configuration
Intf - Not programmed, interface does not exist
```

Owner	AF	Proto	State	Interface	VRF	Local ip,port	>	Remote ip,port
Client	2	6	Prog		default	192.168.0.1/32,57600	>	10.0.0.2/32,0
Linux	2	6	Ovr		global-vrf	any,57600	>	any,0

```
Router# show lpts bindings brief | include TPA
0/RP0/CPU0 TPA LR IPV4 TCP default Mg0/RP0/CPU0/0 192.168.0.1,57600 10.0.0.2
Router#
Router# 0/RP0/ADMIN0:Mar 19 15:22:26.837 IST: pm[2433]:
%INFRA-Process_Manager-3-PROCESS_RESTART : Process tams (IID: 0) restarted
```




第 7 章

セキュア シェルの実装

セキュア シェル (SSH) は、Berkeley の `r` ツールへのセキュアな置換を提供するアプリケーションおよびプロトコルです。プロトコルは標準の暗号メカニズムを使用してセッションの安全を確保します。アプリケーションは Berkeley の `rexec` および `rsh` ツールと同様に使用できます。

SSH サーバとして、SSH バージョン 1 (SSHv1) と SSH バージョン 2 (SSHv2) の 2 種類のバージョンを使用できます。SSHv1 は Rivest、Shamir、Adelman (RSA) キーを使用し、SSHv2 はデジタル署名アルゴリズム (DSA) キーまたは Rivest、Shamir、Adelman (RSA) キー、または楕円曲線デジタル署名アルゴリズム (ECDSA) キーを使用します。Cisco ソフトウェアは SSHv1 と SSHv2 の両方をサポートしています。

このモジュールでは、セキュア シェルの実装方法について説明します。

- [セキュア シェルの実装 \(83 ページ\)](#)

セキュア シェルの実装

セキュア シェル (SSH) は、Berkeley の `r` ツールへのセキュアな置換を提供するアプリケーションおよびプロトコルです。プロトコルは標準の暗号メカニズムを使用してセッションの安全を確保します。アプリケーションは Berkeley の `rexec` および `rsh` ツールと同様に使用できます。

SSH サーバとして、SSH バージョン 1 (SSHv1) と SSH バージョン 2 (SSHv2) の 2 種類のバージョンを使用できます。SSHv1 は Rivest、Shamir、Adelman (RSA) キーを使用し、SSHv2 はデジタル署名アルゴリズム (DSA) キーまたは Rivest、Shamir、Adelman (RSA) キー、または楕円曲線デジタル署名アルゴリズム (ECDSA) キーを使用します。Cisco ソフトウェアは SSHv1 と SSHv2 の両方をサポートしています。

このモジュールでは、セキュア シェルの実装方法について説明します。

セキュア シェルの実装に関する前提条件

セキュア シェルを実装するには、次の前提条件を満たす必要があります。

- 必要なイメージをルータにダウンロードします。SSH サーバと SSH クライアントでは、暗号化パッケージ（データ暗号規格（DES）、トリプルDES、およびAES）をシスコからご使用のルータにダウンロードする必要があります。
- ローカル アクセスまたはリモート アクセス用にユーザ認証を設定します。認証、許可、アカウントिंग（AAA）の有無に関係なく、認証を設定できます。
- AAA 認証および認可はセキュア ファイル転送プロトコル（SFTP）で機能するように正しく設定されている必要があります。

セキュア シェルの実装に関する制約事項

SSH の基本的な制約事項と SFTP 機能の制限は、次のとおりです。

- 外部クライアントがルータに接続するには、ルータに RSA（SSHv1 または SSHv2 の場合）または DSA（SSHv2 の場合）または ECDSA（SSHv2 の場合）キーペアが設定されている必要があります。ルータから外部ルーティング デバイスに SSH クライアント接続を開始する場合、ECDSA、DSA および RSA キーは必要ありません。これは SFTP も同様です。SFTP はクライアント モードでのみ動作するため、ECDSA、DSA および RSA キーは必要ありません。
- SFTP が正常に動作するには、リモート SSH サーバは SFTP サーバ機能をイネーブルにする必要があります。たとえば、`/etc/ssh2/sshd2_config` などの行を使用して、SFTP サブシステムを処理するように SSHv2 サーバを設定します。
- **subsystem-sftp /usr/local/sbin/sftp-server**
- SFTP サーバは通常パブリック ドメインの SSH パッケージの一部として含まれており、デフォルトの構成では有効になっています。
- SFTP は、SFTP サーババージョン OpenSSH_2.9.9p2 以上と互換性があります。
- SSH サーバおよび SFTP サーバでは、RSA ベースのユーザ認証がサポートされています。ただし、SSH クライアントではこの認証はサポートされていません。
- サポートされるアプリケーションは、実行シェルおよび SFTP のみです。
- SFTP クライアントは、ワイルドカード（*?、[]）を含むリモート ファイル名をサポートしません。ソース ファイルをルータにダウンロードするには、ユーザは **sftp** コマンドを複数回発行するか、リモート ホストからすべてのソース ファイルを表示する必要があります。アップロードについては、この項の 1 番目から 3 番目までの箇条書きで示した問題が解決されている場合、ルータ SFTP クライアントはワイルドカードを使用した複数ファイルの指定をサポートできます。
- SSH サーバの暗号化設定は、AES128、AES192、AES256、トリプルDES の順です。サポートされていない暗号の場合、サーバはクライアントの要求をすべて拒否し、SSH セッションは続行されません。
- vt100 以外の端末タイプの使用はサポートされていません。この場合、ソフトウェアは警告メッセージを生成します。

- SSH クライアントでは、パスワードメッセージとして「none」を使用することはサポートされていません。
- ルータ インフラストラクチャは UNIX 同様のファイル権限をサポートしていないため、ローカル デバイスに作成されたファイルは元の権限情報を失います。リモート ファイル システム上に作成されたファイルの場合、ファイル権限は宛先ホストの `umask` に従い、変更時間および最終アクセス時間はコピーの時間になります。

SSH の設定

SSH を設定するには、次のタスクを実行します。



(注) SSHv1 の設定では、ステップが 1 ~ 4 が必須です。SSHv2 の設定では、ステップ ~ ステップ 4 はオプションです。

手順

ステップ 1 **configure**

ステップ 2 **hostname** *hostname*

例 :

```
RP/0/RP0/CPU0:router(config)# hostname router1
```

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/CPU0:router(config)# domain name cisco.com
```

このソフトウェアで使用するデフォルトのドメイン名を定義し、不完全なホスト名のドメインを補完します。

ステップ 4 **commit**

ステップ 5 **crypto key generate rsa** [**usage keys** | **general-keys**] [*keypair-label*]

例 :

```
RP/0/RP0/CPU0:router# crypto key generate rsa general-keys
```

RSA キー ペアを生成します。RSA キー モジュラスの範囲は 512 ~ 4096 ビットです。

- RSA キー ペアを削除するには、**crypto key zeroize rsa** コマンドを使用します。
- このコマンドは SSHv1 だけに使用されます。

ステップ 6 `crypto key generate dsa`

例 :

```
RP/0/RP0/CPU0:router# crypto key generate dsa
```

ルータでローカルおよびリモート認証を行う SSH サーバをイネーブルにします。サポートされているキーのサイズは 512、768 および 1024 ビットです。

- 推奨する最小絶対サイズは 1024 ビットです。
- DSA キー ペアを生成します。
DSA キー ペアを削除するには、**crypto key zeroize dsa** コマンドを使用します。
- このコマンドは、SSHv2 だけに使用されます。

ステップ 7 `crypto key generate ecdsa [nistp256 | nistp384 | nistp521]`

例 :

```
RP/0/RP0/CPU0:router# crypto key generate ecdsa nistp256
```

ECDSA キーペアを生成します。サポートされている ECDSA 曲線タイプは、Nistp256、Nistp384 および Nistp521 です。

- ECDSA キー ペアを削除するには、**crypto key zeroize ecdsa [nistp256 | nistp384 | nistp521]** コマンドを使用します。
- このコマンドは SSHv2 だけに使用されます。

ステップ 8 `configure`

例 :

```
RP/0/RP0/CPU0:router# configure
```

モードを開始します。

ステップ 9 `ssh timeout seconds`

例 :

```
RP/0/RP0/CPU0:router(config)# ssh timeout 60
```

(任意) AAA へのユーザ認証に対するタイムアウト値を設定します。

- 設定された時間内にユーザ自身の認証が AAA に対してできないと、接続は中断されます。
- 値を設定しなければ、30 秒のデフォルト値が使用されます。範囲は 5 ~ 120 です。

ステップ 10 次のいずれかを実行します。

- **ssh server [vrf vrf-name]**
- **ssh server v2**

例 :

```
RP/0/RP0/CPU0:router(config)# ssh server v2
```

- (任意) 32 文字までの指定された VRF を使用して SSH サーバを起動します。VRF が指定されていない場合、デフォルトの VRF が使用されます。

SSH サーバが指定された VRF の接続をこれ以上受信しないようにするには、このコマンドの **no** 形式を使用します。VRF が指定されていない場合、デフォルトが使用されます。

(注) SSH サーバは複数の VRF を使用するように設定できます。

- (任意) **ssh server v2** コマンドを使用して SSHv2 オプションを設定すると、SSH サーバは SSHv2 クライアントだけを受け入れるようになります。**ssh server v2** コマンドを選択すると、SSH v2 クライアント接続だけが許可されます。

ステップ 11 **commit**

ステップ 12 **show ssh**

例 :

```
RP/0/RP0/CPU0:router# show ssh
```

(任意) ルータへの着信および発信の SSHv1 接続と SSHv2 接続をすべて表示します。

ステップ 13 **show ssh session details**

例 :

```
RP/0/RP0/CPU0:router# show ssh session details
```

(任意) ルータに対する SSHv2 接続の詳細レポートを表示します。

ステップ 14 **show ssh history**

例 :

```
RP/0/RP0/CPU0:router# show ssh history
```

(オプション) 終了した最後の 100 個の SSH 接続を表示します。

ステップ 15 **show ssh history details**

例 :

```
RP/0/RP0/CPU0:router# show ssh history details
```

(オプション) 終了した最後の 100 個の SSH 接続を詳細情報とともに表示します。このコマンドは **show ssh session details** コマンドに似ていますが、セッションの開始時刻と終了時刻も示されます。

ステップ 16 **show tech-support ssh**

例 :

```
RP/0/RP0/CPU0:router# show tech-support ssh
```

(オプション) システム情報を表示する `show` コマンドを自動的に実行します。



(注) SSH 接続のネゴシエーションを行う際の優先順位は、次のとおりです。

1. ecdsa-nistp-521
2. ecdsa-nistp-384
3. ecdsa-nistp-256
4. rsa
5. dsa

SSH クライアントの設定

SSH クライアントを設定するには、次の作業を実行します。

手順

ステップ 1 `configure`

ステップ 2 `ssh client knownhost device :/filename`

例 :

```
RP/0/RP0/CPU0:router(config)# ssh client knownhost slot1:/server_pubkey
```

(任意) この機能がクライアント側でサーバの公開キー (pubkey) を認証し、確認できるようにします。

(注) ファイル名の完全なパスが必要です。コロン (:) とスラッシュ (/) も必要です。

ステップ 3 `commit`

ステップ 4 `ssh {ipv4-address | ipv6-address | hostname} [username user- cipher | source-interface type instance]`

発信 SSH 接続をイネーブルにします。

- SSHv2 サーバを実行するには、VRF が必要です。これはデフォルトの VRF でも固有の VRF でも構いません。VRF に関する変更は SSH v2 サーバのみに適用されます。
- SSH クライアントにより、リモートピアへの SSHv2 接続が試みられます。リモートピアで SSHv1 サーバしかサポートされていない場合、そのピアでリモートサーバへの SSHv1 接続が内部生成されます。

- **cipher des** オプションは、SSHv1 クライアントでのみ使用可能です。
- SSHv1 クライアントは、3DES 暗号化アルゴリズム オプションだけをサポートします。このオプションは、これらの SSH クライアントに対してだけ、まだデフォルトで使用可能です。
- **hostname** 引数が使用され、ホストに IPv6 と IPv4 の両方のアドレスがある場合、IPv6 アドレスが使用されます。

-
- SSHv1 を使用しており、SSH 接続が拒否されている場合は、ルータの RSA キー ペアが適切に生成されていません。ホスト名およびドメインを指定していることを確認します。次に、**crypto key generate rsa** コマンドを使用して RSA キー ペアを生成し、SSH サーバをイネーブルにします。
 - SSHv2 を使用しており、SSH 接続が拒否されている場合は、ルータの DSA キー ペア、または RSA キー ペア、または ECDSA キー ペアが適切に生成されていません。ホスト名およびドメインを指定していることを確認します。次に、**crypto key generate dsa** コマンドを使用して DSA キー ペアを生成するか、または **crypto key generate rsa** コマンドを使用して RSA キー ペアを生成するか、または **crypto key generate ecdsa [nistp256 | nistp384 | nistp521]** コマンドを使用して ECDSA キー ペアを生成し、SSH サーバを有効にします。
 - ECDSA、RSA または DSA キー ペアを設定する場合、次のエラー メッセージが表示されることがあります。

- No hostname specified

hostname コマンドを使用して、ルータのホスト名を設定する必要があります。

- No domain specified

domain-name コマンドを使用して、ルータのホストドメインを設定する必要があります。

- 使用できる SSH 接続数は、ルータに設定されている仮想端末回線の最大数に制限されません。各 SSH 接続は vty リソースを使用します。
- SSH では、ルータで AAA によって設定されるローカルセキュリティまたはセキュリティ プロトコルが、ユーザ認証に使用されます。AAA を設定する場合、コンソール上で AAA を無効にするためにグローバル コンフィギュレーション モードでキーワードを適用することにより、コンソールが AAA の下で実行されていないことを確認する必要があります。



(注) PuTTY バージョン 0.63 以降を使用して SSH クライアントに接続する場合は、PuTTY 設定の [SSH] > [Bugs] で [Chokes on PuTTYs SSH2 winadj request] オプションを [On] に設定します。これにより、大量の出力が IOS XR から PuTTY クライアントに送信されるたびにセッションが中断する可能性を回避できます。

セキュア シェルの設定：例

次に、SSHv2サーバを設定する方法の例を示します。この例では、ホスト名を作成し、ドメイン名を定義し、DSA キー ペアを生成することでルータでのローカルおよびリモート認証に対して SSH サーバをイネーブルにし、SSH サーバを起動し、コンフィギュレーションファイルを実行するためのコンフィギュレーションコマンドを保存します。

SSH の設定が完了すると、ルータで SFTP 機能が使用できます。

```
configure
hostname router1
domain name cisco.com
exit
crypto key generate rsa/dsa
configure
ssh server
end
```

セキュア シェルの実装について

SSH を実装するには、次の概念について理解しておく必要があります。

SSH サーバ

SSH サーバの機能によって、SSH クライアントは Cisco ルータに対してセキュアで暗号化された接続を実行できます。この接続は、インバウンド Telnet 接続の機能と同様です。SSH 以前は、セキュリティは Telnet のセキュリティに限定されていました。SSH を Cisco ソフトウェアの認証と併用することで、強力な暗号化が可能になります。Cisco ソフトウェアの SSH サーバは、市販の一般的な SSH クライアントと相互運用できます。

SSH クライアント

SSH クライアント機能は、SSH プロトコルを介して実行されるアプリケーションで、認証と暗号化を行います。SSH クライアントによって、Cisco ルータは他の Cisco ルータ、または SSH サーバを実行する他のデバイスに対して、セキュアで暗号化された接続を実行できます。この接続は、接続が暗号化されている点を除き、アウトバウンド Telnet 接続の機能と同様です。認証と暗号化により、SSH クライアントは、セキュリティ保護されていないネットワーク上でもセキュアな通信を実現できます。

SSH クライアントは、市販の一般的な SSH サーバと連動します。SSH クライアントは、AES、3DES、メッセージダイジェストアルゴリズム 5 (MD5)、SHA1、およびパスワード認証の暗号をサポートしています。ユーザ認証はルータへの Telnet セッションで実行されます。SSH がサポートするユーザ認証メカニズムには、Remote Authentication Dial-In User Service (RADIUS)、TACACS+、およびローカルに格納されたユーザ名とパスワードを使用した認証があります。

SSH クライアントでは、発信パケットに DSCP 値を設定することができます。

```
ssh client dscp <value from 0 - 63>
```


設定しない場合は、（クライアントとサーバの両方の）パケットにデフォルトの DSCP 値 16 が設定されます。

SSH クライアントは次のオプションをサポートしています。

- **DSCP** : SSH クライアントセッションの DSCP 値。

```
RP/0/5/CPU0:router#configure
RP/0/5/CPU0:router(config)#ssh ?
  client  Provide SSH client service
  server  Provide SSH server service
  timeout Set timeout value for SSH
RP/0/5/CPU0:router(config)#ssh client ?
```

- **Knownhost** : ローカルデータベースでのホストの **pubkey** チェックをイネーブルにします。
- **Source-interface** : SSH クライアントセッションの送信元インターフェイス。

```
RP/0/5/CPU0:router(config)#ssh client source-interface ?
  ATM          ATM Network Interface(s)
  BVI          Bridge-Group Virtual Interface
  Bundle-Ether Aggregated Ethernet interface(s)
  CEM          Circuit Emulation interface(s)
  GigabitEthernet GigabitEthernet/IEEE 802.3 interface(s)
  IMA          ATM Network Interface(s)
  IMtestmain   IM Test Interface
  Loopback     Loopback interface(s)
  MgmtEth      Ethernet/IEEE 802.3 interface(s)
  Multilink    Multilink network interface(s)
  Null         Null interface
  PFItestmain  PFI Test Interface
  PFItestnothw PFI Test Not-HW Interface
  PW-Ether     PWHE Ethernet Interface
  PW-IW        PWHE VC11 IP Interworking Interface
  Serial       Serial network interface(s)
  VASILeft     VASI Left interface(s)
  VASIRight    VASI Right interface(s)
  test-bundle-channel Aggregated Test Bundle interface(s)
  tunnel-ipsec IPsec Tunnel interface(s)
  tunnel-mte   MPLS Traffic Engineering P2MP Tunnel interface(s)
  tunnel-te    MPLS Traffic Engineering Tunnel interface(s)
  tunnel-tp    MPLS Transport Protocol Tunnel interface
RP/0/5/CPU0:router(config)#ssh client source-interface
RP/0/5/CPU0:router(config)#
```

SSH では、次のようにリモート コマンドを実行することもできます。

```
RP/0/5/CPU0:router#ssh ?
  A.B.C.D IPv4 (A.B.C.D) address
  WORD   Hostname of the remote node
  X:X::X IPv6 (A:B:C:D...:D) address
  vrf     vrf table for the route lookup
RP/0/5/CPU0:router#ssh 1.1.1.1 ?
  cipher      Accept cipher type
  command     Specify remote command (non-interactive)
  source-interface Specify source interface
  username    Accept userid for authentication
  <cr>
RP/0/5/CPU0:router#ssh 12.28.46.6 username admin command "show redundancy sum"
Password:
```

```
Wed Jan  9 07:05:27.997 PST
  Active Node   Standby Node
```

```

-----
0/4/CPU0          0/5/CPU0 (Node Ready, NSR: Not Configured)
RP/0/5/CPU0:router#

```

SFTP 機能の概要

SSH には、SSHv2 で導入された新たな標準ファイル転送プロトコルである Standard File Transfer Protocol (SFTP) のサポートが含まれています。この機能は、ルータ設定またはルータイメージファイルをコピーするセキュアで認証された方法を提供します。

SFTP クライアント機能は SSH コンポーネントの一部として提供され、ルータで常にイネーブルになっています。このため、適切なレベルのユーザは、ルータへのファイルのコピーおよびルータからのファイルのコピーが可能です。**copy** コマンドと同様に、**sftp** コマンドは XREXEC モードでのみ使用できます。

SFTP クライアントは VRF 対応であるため、接続の試行時に特定の送信元インターフェイスに関連付けられた VRF を使用するようにセキュア FTP クライアントを設定することもできます。SFTP クライアントはインタラクティブ モードもサポートしています。このモードでは、ユーザはサーバにログインして特定の作業を UNIX サーバ経由で実行できます。

SFTP サーバは SSH サーバのサブシステムです。つまり、SSH サーバが SFTP サーバ要求を受信すると、SFTP API は SSH サーバに対して子プロセスとして SFTP サーバを作成します。新たな要求のたびに、新しい SFTP サーバインスタンスが作成されます。

SFTP は、次の手順で新たな SFTP サーバを要求します。

- ユーザが必要な引数を指定して **sftp** コマンドを実行します
- SFTP API は SSH サーバと通信する子プロセスを内部に作成します
- SSH サーバは SFTP サーバ子プロセスを作成します
- SFTP サーバおよびクライアントは暗号化形式で相互に通信します
- SFTP 転送は LPTS ポリサー「SSH-Known」の影響を受けます。ポリサーの値が低いと、SFTP 転送の速度に影響します。



(注) IOS-XR ソフトウェア リリース 4.3.1 以降では、SSH-Known のデフォルトのポリサー値が 2500 pps から 300 pps にリセットされました。この変更により転送速度の低下が予想されます。このバントの原因となる LPTS ポリサー値を高い値に調整することにより、転送速度を上げることができます

SSH サーバが SSH クライアントと新たな接続を確立すると、サーバデーモンは新たな SSH サーバ子プロセスを作成します。子サーバプロセスは、キー交換とユーザ認証プロセスによって、SSH クライアントとサーバとの間にセキュアな通信チャネルを構築します。SSH サーバがサブシステムを SFTP サーバにする要求を受信した場合、SSH サーバデーモンは SFTP サーバ子プロセスを作成します。SFTP サーバサブシステム要求を受信するたびに、新たな SSH サーバ子インスタンスおよび SFTP サーバインスタンスが作成されます。SFTP サーバはユーザ

セッションを認証し、接続を開始します。ユーザのデフォルトディレクトリおよびクライアントの環境を設定します。

初期化が実行されると、SFTP サーバはクライアントからの SSH_FXP_INIT メッセージを待機します。このメッセージは、ファイル通信セッションを開始するためには不可欠です。このメッセージの後に、クライアントの要求に基づいたメッセージが続く場合があります。ここでは、プロトコルは「要求応答」モデルを採用しています。クライアントがサーバに要求を送信すると、サーバはこの要求を処理し応答を送信します。

SFTP サーバは次の応答を表示します。

- ステータス応答
- 処理応答
- データ応答
- 名前応答



(注) サーバは、着信する SFTP 接続を受け付けるために稼働している必要があります。

RSA ベースのホスト認証

サーバの正当性を検証することは、セキュアな SSH 接続を実現する最初の手順です。このプロセスはホスト認証と呼ばれ、クライアントが有効なサーバに接続していることを確認するために実施されます。

ホスト認証はサーバの公開キーを使用して実行されます。サーバは、キー交換フェーズの間に公開キーをクライアントに提供します。クライアントはこのサーバの既知ホストのデータベースと、対応する公開キーをチェックします。クライアントでサーバの IP アドレスが見つからなかった場合は、ユーザに警告メッセージを表示し、ユーザは公開キーを保存するか廃棄するかを選択できます。サーバの IP アドレスは見つかったものの公開キーが一致しない場合、クライアントは接続を終了します。公開キーが有効な場合、サーバは検証され、セキュアな SSH 接続が確立されます。

IOS XR SSH サーバおよびクライアントは、DSA ベースのホスト認証をサポートしていましたが、ただし、IOS などの他の製品との互換性のため、RSA ベースのホスト認証のサポートも追加されました。

RSA ベースのユーザ認証

SSH プロトコルにおいてユーザを認証する方法の 1 つに、RSA 公開キー ベースのユーザ認証があります。秘密キーの保持がユーザ認証の役割を果たします。この方法は、ユーザの秘密キーで作成した署名を送信することで機能します。各ユーザは RSA キーペアをクライアントマシンに保持しています。RSA キーペアの秘密キーはクライアントマシンに残ったままです。

ユーザは、ssh-keygen などの標準的なキー生成メカニズムを使用して、RSA 公開キーと秘密キーのキーペアを UNIX クライアント上に生成します。サポートされているキーの最大の長さ

は 4096 ビットで、最小の長さは 512 ビットです。次に、一般的なキー生成アクティビティの例を示します。

```
bash-2.05b$ ssh-keygen -b 1024 -t rsa
Generating RSA private key, 1024 bit long modulus
```

公開キーを正常にボックスにインポートするには、公開キーが Base64 エンコード (バイナリ) 形式である必要があります。インターネットで入手できるサードパーティのツールを使用して、キーをバイナリ形式に変換できます。

公開キーがルータにインポートされると、SSH クライアントは内部で「-o」オプションを使用して要求を指定することで、公開キー認証方式を使用できるようになります。次に例を示します。

```
client$ ssh -o PreferredAuthentications=publickey 1.2.3.4
```

公開キーが RSA 方式によってルータにインポートされていない場合、SSH サーバはパスワードベースの認証を開始します。公開キーがインポートされている場合、サーバは両方の方式の使用を提案します。SSH クライアントはいずれかの方式を使用して、接続を確立します。SSH クライアントからの発信接続の数は 10 まで許可されます。

現時点では、SSH バージョン 2 および SFTP サーバのみが RSA ベースの認証をサポートしています。



(注) 推奨される認証方法は SSH RFC に記載されています。RSA ベース認証のサポートはローカル認証のみです。TACACS/RADIUS サーバに対してはサポートされていません。

認証、許可、およびアカウントティング (AAA) は、Cisco ルータまたはアクセスサーバにアクセス コントロールを設定できる主要なフレームワークを提供する一連のネットワーク セキュリティ サービスです。

SSHv2 クライアント キーボード インタラクティブ 認証

キーボードを使用して認証情報を入力する認証方式は、キーボードインタラクティブ認証と呼ばれます。この方式は、SSH プロトコルのインタラクティブな認証方式です。この認証方式では、SSH クライアントは、認証方法の基本的メカニズムを考慮することなく、さまざまな認証方法をサポートできます。

現在、SSHv2 クライアントはキーボードインタラクティブ認証をサポートしています。この認証方式は、インタラクティブなアプリケーションでのみ機能します。



(注) パスワード認証はデフォルトの認証方式です。キーボードインタラクティブ認証方式は、キーボードインタラクティブ認証のみをサポートするようにサーバが設定されている場合に選択されます。