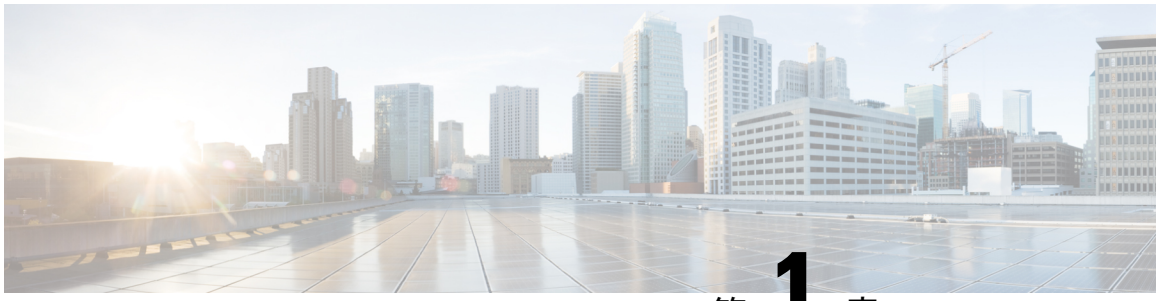
Cisco IOS リリース 15.2(8)E（Catalyst マイクロスイッチ シリーズ）セキュリティ コンフィギュレーション ガイド

初版：2021 年 4 月 26 日

最終更新：2021 年 12 月 21 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/

目 次



第 1 章

セキュリティ機能の概要

セキュリティ機能の概要

セキュリティ機能は次のとおりです。

- ACL および RADIUS Filter-Id 属性を使った IEEE 802.1x 認証。
- 管理インターフェイス（デバイス マネージャ、Network Assistant、CLI）へのパスワード保護付きアクセス（読み取り専用および読み書きアクセス）。不正な設定変更を防止します。
- セキュリティ レベル、通知、および対応するアクションを選択できる、マルチレベル セキュリティ。
- セキュリティを確保できるスタティック MAC アドレッシング。
- 保護ポート オプション。同一スイッチ上の指定ポートへのトラフィック転送を制限します。
- ポートにアクセスできるステーションの MAC アドレスを制限または特定するポート セキュリティ オプション。
- VLAN 認識ポート セキュリティ オプション。違反の発生時にポート全体をシャットダウンするのではなく、そのポート上の VLAN をシャットダウンします。
- ポート セキュリティ エージング。ポートのセキュア アドレスにエージング タイムを設定します。
- 指定した入力割合を超えたパケットをドロップして、スイッチへの着信プロトコル トラフィックの割合を制御する、プロトコル ストーム プロテクション。
- BPDU ガード。無効なコンフィギュレーションが発生した場合に、PortFast が設定されているポートをシャットダウンします。
- 標準および拡張 IP アクセス コントロール リスト（ACL）は、レイヤ 2 インターフェイス（ポート ACL）でのインバウンドなセキュリティ ポリシーを定義します。

- MAC 拡張アクセス コントロール リスト。レイヤ 2 インターフェイスの着信方向のセキュリティ ポリシーを定義します。
- 信頼できないホストと DHCP サーバの間の信頼できない DHCP メッセージをフィルタリングする DHCP スヌーピング。
- 不正な ARP 要求や応答を同じ VLAN 上のその他のポートにリレーしないことにより、スイッチに対する悪意のある攻撃を回避するためのダイナミック ARP インスペクション。
- IEEE 802.1x ポートベース認証。不正なデバイス（クライアント）によるネットワーク アクセスを防止します。次の 802.1x 機能がサポートされます。
  - シングルホスト、マルチホスト、マルチ認証、およびマルチドメイン認証モードのサポート。

モード	説明
単一ホスト	認証できるホストは 1 つだけです。複数のクライアントが認証を試みると、セキュリティ違反が発生します。
複数ホスト	最初のホストのみが認証を必要とします。残りのホストは認証なしでアクセスできます。
マルチ認証	すべてのクライアントが認証される必要があります。
マルチドメイン認証	1 つの VoIP クライアントと 1 つのデータ クライアントが認証を許可されます。複数のクライアントが認証を試みると、セキュリティ違反が発生します。

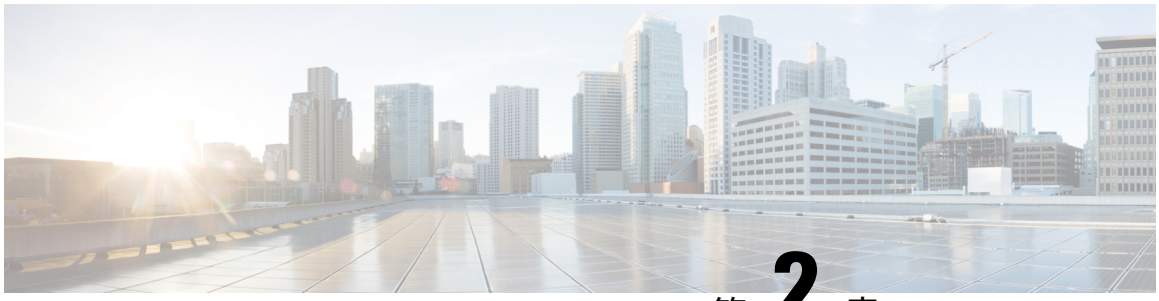
  - データ装置と IP Phone などの音声装置（シスコ製品またはシスコ以外の製品）の両方が、同じ IEEE 802.1x 対応スイッチ ポートにおいて、単独で認証できるようにするマルチドメイン認証（MDA）。
  - MDA のダイナミック音声 VLAN（仮想 LAN）。ダイナミック音声 VLAN が MDA 対応ポートで可能になります。
  - VLAN 割り当て。802.1x 認証ユーザを特定の VLAN に制限します。
  - マルチ認証モードで設定されたポートでの VLAN 割り当てのサポート。RADIUS サーバは、ポートで最初に認証されるホストに VLAN を割り当て、後続のホストは同じ VLAN を使用します。音声 VLAN 割り当ては、1 つの IP Phone に対してサポートされます。
  - 音声 VLAN。ポートが許可ステートか無許可ステートかにかかわらず、Cisco IP Phone の音声 VLAN へのアクセスを許可します。
  - IP Phone 検出機能拡張。Cisco IP Phone を検出し識別します。

- ゲスト VLAN。802.1x に適合しないユーザに限定的なサービスを提供します。
- 制限付き VLAN。802.1x に準拠はしているが、標準の 802.1x で認証するためのクレデンシャルを持っていないユーザに制限付きのサービスを提供します。
- 802.1x アカウンティング。ネットワーク使用をトラッキングします。
- 802.1x と LAN の Wake-on-LAN（WoL）機能。休止状態の PC に、特定のイーサネット フレームを送信して起動させます。
- 802.1x 準備状態チェック。スイッチで IEEE 802.1x を設定する前に、接続されたエンド ホストの準備状態を判断します。
- セキュリティ違反が発生した VLAN だけでトラフィック違反アクションを適用するための音声認識 802.1x セキュリティ。
- MAC 認証バイパス（MAB）。クライアント MAC アドレスに基づいてクライアントを許可します。
- デバイスのネットワーク アクセスを許可する前の、エンドポイント システムやクライアントのウイルス対策の状態またはポスチャに関する Network Admission Control（NAC）レイヤ 2 802.1x 検証。
- 802.1X スイッチ サプリカントを持つ Network Edge Access Topology（NEAT）、CISP を使ったホスト認証、および自動イネーブル化。これらにより、別のスイッチへのサプリカントとして、配線クローゼットの外のスイッチが認証されます。
- 認証される前にネットワークへのアクセスをホストに許可するための、オープン アクセスを使用した IEEE 802.1x。
- リダイレクト URL を使用した IEEE 802.1x 認証。RADIUS サーバーまたは Cisco Identity Services Engine（ISE）から認証されたスイッチへのユーザー単位の ACL ダウンロードを使用できるようになります。
- 新しいホストを認証するときに、ポートが思考する認証メソッドの順序を設定するための柔軟な認証シーケンス。

- TACACS+。IPv4 および IPv6 対応の TACACS サーバを介してネットワーク セキュリティを管理する独自の機能。
- 認証、許可、およびアカウンティング（AAA）サービスを使用して、リモートユーザの ID の検証、アクセスの許可、アクションの追跡を実行するための RADIUS。
- RADIUS、TACACS+、および SSH の機能拡張。
- ACL および RADIUS Filter-Id 属性を使った IEEE 802.1x 認証。
- RADIUS 認証の変更（CoA）。特定のセッション認証された後で、その属性を変更します。AAA でユーザ、またはユーザ グループのポリシーに変更がある場合、管理者は Cisco Identity Services Engine または Cisco Secure ACS などの AAA サーバから、RADIUS CoA パケットを送信し、新しいポリシーに適用することができます。

セキュリティ機能の概要

- IEEE 802.1x User Distribution。さまざまな VLAN にわたってユーザをロード バランシングすることにより、（ユーザ グループに対して）複数の VLAN を使った配置で、ネットワークのスケーラビリティを向上させることができます。認証されたユーザは、RADIUS サーバにより割り当てられた、グループ内で最も空いている VLAN に割り当てられます。
- クリティカル VLAN のサポート：AAA サーバーが到達不能になった場合に、重要なリソースへのアクセスを許可するために、マルチホスト/マルチ認証対応ポートが重要な VLAN に配置されます。
- ポート ホスト モードを変更し、オーセンティケータのスイッチ ポートに標準ポート設定を適用するために Network Edge Access Topology（NEAT）をサポート。
- MAC 認証バイパス（MAB）を使用した MAC アドレスベースの認証。認証済みホストは、許可されていない VLAN からのネットワークアクセスを防止するために、ダイナミック VLAN に移動されます。
- MAC 移動。モビリティのイネーブル化を制約することなく、ホスト（IP Phone の背後で接続されたホストを含む）が同じスイッチ内のポート間を移動できるようになります。MAC 移動では、もう 1 つのポートに同じ MAC アドレスが再登場した場合、スイッチはこれをまったく新しい MAC アドレスと同様に扱います。
- 簡易ネットワーク管理プロトコル バージョン 3（SNMPv3）を使った 3DES および AES のサポート。このリリースでは、168 ビットの Triple Data Encryption Standard（3DES）と、SNMPv3 への 128 ビット、192 ビット、および 256 ビットの Advanced Encryption Standard（AES）暗号化アルゴリズムに対するサポートが追加されます。
- Cisco TrustSec SXP プロトコルはサポートされていません。



第 2 章

パスワードおよび権限レベルによるスイッチ アクセスの制御

パスワードおよび権限によるスイッチアクセスの制御の制約事項

グローバル コンフィギュレーション モードで **boot manual** コマンドを使用して、スイッチを手動で起動するように設定している場合は、パスワード回復をディセーブルにできません。このコマンドは、スイッチの電源の再投入後、ブートローダ プロンプト（*switch:*）を表示させます。

可逆的パスワードタイプの制約事項とガイドライン

- スタートアップ コンフィギュレーションにタイプ 6 のパスワードがあり、タイプ 6 のパスワードがサポートされていないバージョンにダウングレードすると、デバイスからロックアウトされる可能性があります。

不可逆的パスワードタイプの制約事項とガイドライン

- ユーザ名シークレット パスワード タイプ 5 およびイネーブル シークレット パスワード タイプ 5 は、より強力なパスワードタイプ 8 または 9 に移行する必要があります。詳細については、「暗号化によるイネーブルおよびイネーブル シークレット パスワードの保護（11 ページ）」を参照してください。

- プレーンテキストパスワードは、不可逆的暗号化パスワードタイプ 9 に変換されます。

(注) これは、Cisco IOS Release 15.2(7)E3 以降のリリースでサポートされます。

パスワードおよび権限レベルに関する情報

次の各項では、パスワードと権限レベルについて説明します。

不正アクセスの防止

不正ユーザによる、デバイスの再設定や設定情報の閲覧を防止できます。一般的には、ネットワーク管理者からデバイスへのアクセスを許可する一方、非同期ポートを用いてネットワーク外からダイヤルアップ接続するユーザや、シリアルポートを通じてネットワーク外から接続するユーザ、またはローカルネットワーク内の端末またはワークステーションから接続するユーザによるアクセスを制限します。

デバイスへの不正アクセスを防止するには、次のセキュリティ機能を 1 つまたは複数設定します。

- 最低限のセキュリティとして、各デバイスポートでパスワードおよび権限を設定します。このパスワードは、デバイスにローカルに保存されます。ユーザがポートまたは回線を通じてデバイスにアクセスしようとするとき、ポートまたは回線に指定されたパスワードを入力してからでなければ、デバイスにアクセスできません。
- 追加のセキュリティレイヤとして、ユーザ名とパスワードをペアで設定することもできます。このペアはデバイスでローカルに保存されます。このペアは回線またはポートに割り当てられ、各ユーザを認証します。ユーザは認証後、デバイスにアクセスできます。権限レベルを定義している場合は、ユーザ名とパスワードの各ペアに特定の権限レベルを、対応する権利および権限とともに割り当てることもできます。
- ユーザ名とパスワードのペアを使用したいが、そのペアをローカルではなく中央のサーバに保存したい場合は、セキュリティ サーバ上のデータベースに保存できます。これにより、複数のネットワーキング デバイスが同じデータベースを使用してユーザ認証情報を（必要に応じて許可情報も）得ることができます。
- また、失敗したログイン試行をログに記録するログイン拡張機能もイネーブルにすることもできます。ログイン拡張は、設定した回数のログインが失敗したあとに、それ以降のログイン試行をブロックするために設定することもできます。詳細については、『Cisco IOS Login Enhancements』マニュアルを参照してください。

デフォルトのパスワードおよび権限レベル設定

ネットワークで端末のアクセスコントロールを行う簡単な方法は、パスワードを使用して権限レベルを割り当てることです。パスワード保護によって、ネットワークまたはネットワークデバイスへのアクセスが制限されます。権限レベルによって、ネットワークデバイスにログイン後、ユーザがどのようなコマンドを使用できるかが定義されます。

次の表に、デフォルトのパスワードおよび権限レベル設定を示します。

表 *1:* デフォルトのパスワードおよび権限レベル設定

機能	デフォルト設定
イネーブルパスワードおよび権限レベル	パスワードは定義されていません。デフォルトはレベル 15 です（特権 EXEC レベル）。パスワードは、コンフィギュレーション ファイル内では暗号化されていない状態です。
イネーブル シークレット パスワードおよび権限レベル	パスワードは定義されていません。デフォルトはレベル 15 です（特権 EXEC レベル）。パスワードは、暗号化されてからコンフィギュレーション ファイルに書き込まれます。
回線パスワード	パスワードは定義されていません。

追加のパスワード セキュリティ

マスクされていないシークレットパスワード

特にネットワーク間を行き交う、または TFTP サーバに保存されるパスワードに対してセキュリティレイヤを追加するには、グローバル コンフィギュレーション モードで **enable password** コマンドまたは **enable secret** コマンドのいずれかを使用できます。コマンドの作用はどちらも同じです。このコマンドにより、暗号化されたパスワードを設定できます。特権 EXEC モード（デフォルト設定）または特定の権限レベルにアクセスするユーザは、このパスワードを入力する必要があります。

より高度な暗号化アルゴリズムが使用されるので、**enable secret** コマンドを使用することを推奨します。

enable secret コマンドを設定した場合、このコマンドは **enable password** コマンドよりも優先されます。同時に 2 つのコマンドを有効にはできません。

パスワードの暗号化をイネーブルにすると、ユーザ名パスワード、認証キーパスワード、イネーブル コマンド パスワード、コンソールおよび VTY パスワードなど、すべてのパスワードに適用されます。

マスクされたシークレットパスワード

enable secret コマンドを使用すると、パスワードは暗号化されますが、パスワードを入力するときに端末に表示されます。端末でパスワードをマスクするには、**masked-secret** 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。このパスワードの暗号化タイプは、デフォルトではタイプ 9 です。

このコマンドを使用して、コモンクライテリアポリシーのマスクされたシークレットパスワードを設定できます。

パスワードの回復

スイッチに物理的にアクセスできるエンドユーザは、デフォルトで、スイッチの電源投入時にブートプロセスに割り込み、新しいパスワードを入力することによって、失われたパスワードを回復できます。

パスワード回復ディセーブル化機能では、この機能の一部をディセーブルにすることによりスイッチのパスワードへのアクセスを保護できます。この機能がイネーブルの場合、エンドユーザは、システムをデフォルト設定に戻すことに同意した場合に限り、ブートプロセスに割り込むことができます。パスワード回復をディセーブルにしても、ブートプロセスに割り込んでパスワードを変更できますが、コンフィギュレーション ファイル（config.text）および VLAN データベース ファイル（vlan.dat）は削除されます。

パスワード回復をディセーブルにする場合は、エンド ユーザがブート プロセスに割り込んでシステムをデフォルトの状態に戻すような場合に備え、セキュア サーバにコンフィギュレーション ファイルのバックアップ コピーを保存しておくことを推奨します。スイッチ上でコンフィギュレーション ファイルのバックアップ コピーを保存しないでください。仮想端末プロトコル（VTP）トランスペアレントモードでスイッチが動作している場合は、VLAN データベースファイルのバックアップコピーも同様にセキュアサーバに保存してください。スイッチがシステムのデフォルト設定に戻ったときに、XMODEM プロトコルを使用して、保存したファイルをスイッチにダウンロードできます。

パスワードの回復を再びイネーブルにするには、グローバル コンフィギュレーション モードで **service password-recovery** コマンドを使用します。

端末回線の Telnet 設定

初めてスイッチに電源を投入すると、自動セットアップ プログラムが起動して IP 情報を割り当て、この後続けて使用できるようにデフォルト設定を作成します。さらに、セットアップ プログラムは、パスワードによる Telnet アクセス用にスイッチを設定することを要求します。セットアップ プログラムの実行中にこのパスワードを設定しなかった場合は、端末回線に対する Telnet パスワードを設定するときに設定できます。

ユーザ名とパスワードのペア

ユーザ名とパスワードのペアを設定できます。このペアはスイッチ上でローカルに保存されます。このペアは回線またはポートに割り当てられ、各ユーザを認証します。ユーザは認証後、スイッチにアクセスできます。権限レベルを定義している場合は、ユーザ名とパスワードの各ペアに特定の権限レベルを、対応する権利および権限とともに割り当てることもできます。

権限レベル

シスコデバイスでは、権限レベルを使用して、スイッチ動作の異なるレベルに対してパスワード セキュリティを提供します。デフォルトでは、Cisco IOS ソフトウェアは、パスワード セキュリティの 2 つのモード（権限レベル）で動作します。ユーザ EXEC（レベル 1）および特権 EXEC（レベル 15）です。各モードに、最大 16 個の階層レベルからなるコマンドを設定できます。複数のパスワードを設定することにより、ユーザ グループ別に特定のコマンドへのアクセスを許可することができます。

回線の権限レベル

ユーザは、回線にログインし、別の権限レベルをイネーブルに設定することにより、**privilege level** コマンドを使用して設定された権限レベルを上書きできます。また、**disable** コマンドを使用することにより、権限レベルを引き下げることができます。上位の権限レベルのパスワードがわかっていれば、ユーザはそのパスワードを使用して上位の権限レベルをイネーブルにできます。回線の使用を制限するには、コンソール回線に高いレベルまたは権限レベルを指定してください。

たとえば、多くのユーザに**clear line** コマンドへのアクセスを許可する場合、レベル2のセキュリティを割り当て、レベル2のパスワードを広範囲のユーザに配布できます。また、**configure** コマンドへのアクセス制限を強化する場合は、レベル3のセキュリティを割り当て、そのパスワードを限られたユーザグループに配布することもできます。

コマンド権限レベル

コマンドをある権限レベルに設定すると、構文がそのコマンドのサブセットであるコマンドはすべて、そのレベルに設定されます。たとえば、**show ip traffic** コマンドをレベル 15 に設定すると、**show** コマンドと **show ip** コマンドは、異なるレベルに個別に設定しない限り、権限レベルは自動的に 15 に設定されます。

AES パスワード暗号化およびマスター暗号キー

強力で、反転可能な 128 ビットの高度暗号化規格（AES）パスワード暗号化（タイプ 6 暗号化ともいう）を有効にできます。タイプ6暗号化の使用を開始するには、AESパスワード暗号化機能を有効にし、パスワードを暗号化および復号するためのマスター暗号キーを設定します。

AESパスワード暗号化を有効にしてマスターキーを設定すると、タイプ6パスワード暗号化を無効にしない限り、サポートされているアプリケーションの既存および新規作成されたクリアテキストパスワードがすべて、タイプ6暗号化の形式で保存されます。また、既存の弱いすべての暗号化パスワードをタイプ6暗号化パスワードに変換するようにデバイスを設定することもできます。

パスワードおよび権限レベルでスイッチアクセスを制御する方法

次の各項では、パスワードと権限レベルを使用したスイッチアクセスの制御方法に関するさまざまな設定例を示します。

スタティック 有効 パスワードの設定または変更

イネーブル パスワードは、特権 EXEC モードへのアクセスを制御します。

スタティック イネーブル パスワードを設定または変更するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**enable password** *password* 例： `Device(config)# enable password secret321`	特権 EXEC モードにアクセスするための新しいパスワードを定義するか、既存のパスワードを変更します。 デフォルトでは、パスワードは定義されません。 *password*：1 ～ 25 文字の英数字の文字列を指定します。ストリングを数字で始めることはできません。大文字と小文字を区別し、スペースを使用できますが、先行スペースは無視されます。疑問符（?）は、パスワードを作成する場合に、疑問符の前に Ctrl+v を入力すれば使用できます。たとえば、パスワード abc?123 を作成するときは、次のようにします。 1. **abc** を入力します。 2. Ctrl+v を入力します。

	コマンドまたはアクション	目的
		3. **?123** を入力します。 システムからイネーブル パスワードを入力するように求められた場合、疑問符の前に Ctrl+v を入力する必要はなく、パスワードのプロンプトにそのまま abc?123 と入力できます。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

暗号化によるイネーブルおよびイネーブル シークレット パスワードの保護

特権EXECモード（デフォルト）または任意の権限レベルにアクセスするためにユーザが入力する必要がある暗号化パスワードを確立するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	次のいずれかを使用します。 • **enable password** [**level** *level*] {*unencrypted-password* | *encryption-type encrypted-password*}	• 特権 EXEC モードにアクセスするための新しいパスワードを定義するか、既存のパスワードを変更します。

	コマンドまたはアクション	目的

	• **enable secret** [**level** *level*] {*unencrypted-password* | *encryption-type encrypted-password*}
	
	例：
	`Device(config)# enable password level 12 example123`
	
	または
	
	`Device(config)# enable secret 9 $9$sMLBsTFXLnnHTk$0L82`
		• シークレット パスワードを定義します。これは非可逆的な暗号化方式を使用して保存されます。
		
		• （任意）*level* に指定できる範囲は 0 ～ 15 です。レベル 1 が通常のユーザ EXEC モード権限です。デフォルト レベルは 15 です（特権 EXEC モード権限）。
		
		• *unencrypted-password* には、1 ～ 25 文字の英数字の文字列を指定します。ストリングを数字で始めることはできません。大文字と小文字を区別し、スペースを使用できますが、先行スペースは無視されます。デフォルトでは、パスワードは定義されません。
		
		• *encryption-type* の場合、**enable password** に使用可能なオプションはタイプ 0 と 7、**enable secret** に使用可能なオプションはタイプ 0、5、8、および 9 です。暗号化タイプを指定する場合は、暗号化されたパスワードを使用する必要があります。この暗号化パスワードは、別のスイッチの設定からコピーします。シークレット暗号化タイプ 9 はより安全であるため、アップグレードまたはダウングレード時に問題が発生しないように、タイプ 9 を選択することを推奨します。

	コマンドまたはアクション	目的
		（注）

	コマンドまたはアクション	目的

		• シークレットパスワードの暗号化タイプを指定しない場合、パスワードはタイプ 9 に自動的に変換されます。
		• 暗号化タイプを指定してクリアテキストパスワードを入力した場合は、エラーが発生します。
		• グローバル コンフィギュレーション モードで **algorithm-type scrypt** コマンドを使用して、シークレットパスワードにタイプ 9 暗号化を手動で設定することもできます。次に例を示します。
		`Device(config)# username user1 algorithm-type scrypt secret cisco`
		または
		`Device(config)# enable algorithm-type scrypt secret cisco`
		特権 EXEC モードで **write memory** コマンドを実行し、タイプ 9 シークレットをスター

	コマンドまたはアクション	目的
		トアップ コンフィギュレーションに永続的に書き込みます。
ステップ 4	**service password-encryption** 例： `Device(config)# service password-encryption`	（任意）パスワードの定義時または設定の書き込み時に、パスワードを暗号化します。 暗号化を行うと、コンフィギュレーション ファイル内でパスワードが読み取り可能な形式になるのを防止できます。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

マスクされたシークレットパスワードの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的

ステップ 3	次のいずれかを使用します。
	• **username** *name***masked-secret**
	• **username** *name***common-criteria-policy** *policy-name* **masked-secret**
	例：
	`Device(config)# username cisco masked-secret`
	または
	`Device(config)# username common-criteria-policy test-policy masked-secret`
		• マスクされたシークレットパスワードを定義します。これは非可逆的な暗号化方式を使用して保存されます。
		• コモンクライテリアポリシーのマスクされたシークレットパスワードを定義します。
		• マスクされたシークレットパスワードは5文字以上にする必要があります。マスクされたシークレットパスワードの最大長は256 文字です。デフォルトでは、パスワードは定義されません。

ステップ 4	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

端末回線に対する Telnet パスワードの設定

接続された端末回線の Telnet パスワードを設定するには、次の手順を実行します。

始める前に

- エミュレーション ソフトウェアを備えた PC またはワークステーションをスイッチ コンソール ポートに接続するか、または PC をイーサネット管理ポートに接続します。
- コンソール ポートのデフォルトのデータ特性は、9600 ボー、8 データ ビット、1 ストップ ビット、パリティなしです。コマンドラインプロンプトが表示されるまで、Return キーを何回か押す必要があります。

手順

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

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**line vty 0 15** 例： `Device(config)# line vty 0 15`	Telnet セッション（回線）の数を設定し、ライン コンフィギュレーション モードを開始します。 コマンド対応 device では、最大 16 のセッションが可能です。0および15を指定すると、使用できる 16 の Telnet セッションすべてを設定することになります。
ステップ 4	**password** *password* 例： `Device(config-line)# password abcxyz543`	1 つまたは複数の回線に対応する Telnet パスワードを設定します。 *password*：1 ～ 25 文字の英数字の文字列を指定します。ストリングを数字で始めることはできません。大文字と小文字を区別し、スペースを使用できますが、先行スペースは無視されます。デフォルトでは、パスワードは定義されません。
ステップ 5	**end** 例： `Device(config-line)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ユーザ名とパスワードのペアの設定

ユーザ名とパスワードのペアを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device configure terminal`	グローバル コンフィギュレーション モードを開始します。

ステップ 3	**username** *name* [**privilege** *level*] { **password** *encryption-type password*}
	例：
	`Device(config)# username adamsample privilege 1 password secret456`
	`Device(config)# username 111111111111 mac attribute`
		各ユーザのユーザ名、権限レベル、パスワードを設定します。
		• *name*：ユーザ ID を 1 ワードで指定するか、またはMAC アドレスを指定します。スペースと引用符は使用できません。
		• ユーザ名と MAC フィルタの両方に対し、最大 12000 のクライアントを個別に設定できます。
		• *level*：（任意）ユーザがアクセス権を取得した後に持つ特権レベルを指定します。指定できる範囲は 0 ～ 15 です。レベル 15 では特権 EXEC モードでのアクセスが可能です。レベル 1 では、ユーザ EXEC モードでのアクセスとなります。
		• *encryption-type*：暗号化されていないパスワードが続くことを指定する場合は0を入力します。暗号化されたパスワードが後ろに続く場合は7を指定します。
		• *password*：デバイスにアクセスするためにユーザが入力しなければならないパスワードを指定します。パスワードは 1 ～ 25 文字で、埋め込みスペースを使用でき、**username** コマンドの最後のオプションとして指定します。

ステップ 4	次のいずれかを使用します。 • **line console 0**	ライン コンフィギュレーション モードを開始し、コンソールポート（回線0）

	コマンドまたはアクション	目的

	• **line vty 0 15**
	例：
	`Device(config)# line console 0`
	または
	`Device(config)# line vty 15`
		または vty 回線（回線 0 ～ 15）を設定します。

ステップ 5	**login local** 例： `Device(config-line)# login local`	ログイン時のローカル パスワード チェックをイネーブルにします。認証は、ステップ 3 で指定されたユーザ名に基づきます。
ステップ 6	**end** 例： `Device(config-line)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

コマンドの特権レベルの設定

コマンドの権限レベルを設定するには、次の手順に従います。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**privilege** *mode* **level** *level command* 例：	コマンドの特権レベルを設定します。

	コマンドまたはアクション	目的
	`Device(config)# privilege exec level 14 configure`	• *mode*：グローバル コンフィギュレーション モードの場合は **configure** を、EXEC モードの場合は **exec** を、インターフェイス コンフィギュレーション モードの場合は **interface** を、ライン コンフィギュレーション モードの場合は **line** をそれぞれ入力します。 • *level*：範囲は 0 ～ 15 です。レベル 1 が通常のユーザ EXEC モード権限です。レベル 15 は、**enable** パスワードによって許可されるアクセス レベルです。 • *command*：アクセスを制限したいコマンドを指定します。
ステップ 4	**enable password level** *level password* 例： `Device(config)# enable password level 14 SecretPswd14`	権限レベルをイネーブルにするためのパスワードを指定します。 • *level*：範囲は 0 ～ 15 です。レベル 1 が通常のユーザ EXEC モード権限です。 • *password*：1 ～ 25 文字の英数字の文字列を指定します。ストリングを数字で始めることはできません。大文字と小文字を区別し、スペースを使用できますが、先行スペースは無視されます。デフォルトでは、パスワードは定義されません。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

回線のデフォルト特権レベルの変更

ユーザは、回線にログインし、別の権限レベルをイネーブルに設定することにより、**privilege level** コマンドを使用して設定された権限レベルを上書きできます。上位の権限レベルのパスワードがわかっていれば、ユーザはそのパスワードを使用して上位の権限レベルをイネーブルにできます。回線の使用を制限するには、コンソール回線に高いレベルまたは権限レベルを指定してください。

指定した回線のデフォルトの権限レベルを変更するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**line vty** *line* 例： `Device(config)# line vty 10`	アクセスを制限する VTY を選択します。
ステップ 4	**privilege level** *level* 例： `Device(config)# privilege level 15`	回線のデフォルト特権レベルを変更します。 *level*：範囲は 0 ～ 15 です。レベル 1 が通常のユーザ EXEC モード権限です。レベル 15 は、**enable** パスワードによって許可されるアクセス レベルです。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

権限レベルへのログインおよび終了

また、**disable** コマンドを使用することにより、権限レベルを引き下げることができます。

指定した権限レベルにログインし、指定した権限レベルを終了するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** *level* 例： `Device> enable 15`	指定された特権レベルにログインします。 この例で、レベル 15 は特権 EXEC モードです。 *level*：範囲は 0 ～ 15 です。
ステップ 2	**disable** *level* 例： `Device# disable 1`	指定した特権レベルを終了します。 この例で、レベル 1 はユーザ EXEC モードです。 *level*：範囲は 0 ～ 15 です。

暗号化事前共有キーの設定

暗号化事前共有キーを設定するには、次の手順を実行します。

始める前に

この手順は、パスワードタイプ 6 への自動変換の前提条件です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**key config-key password-encrypt** [*text*] 例： `Device(config)# key config-key password-encrypt`	タイプ 6 の暗号キーをプライベート NVRAM に保存します。 • （**Enter** キーを使用して）インタラクティブにキーボード操作を行う場合、暗号キーがすでに存在すれば、

	コマンドまたはアクション	目的
		Old key、New key、Confirm key という3つのプロンプトが表示されます。 • インタラクティブにキーボード操作を行う場合、暗号キーが存在しなければ、New key、Confirm key という2つのプロンプトが表示されます。 • すでに暗号化されているパスワードを削除しようとすると、次のプロンプトが表示されます。 `WARNING: All type 6 encrypted keys will become unusable. Continue with master key deletion? [yes/no]:"`
ステップ 4	**password encryption aes** 例： `Device(config)# password encryption aes`	暗号化事前共有キーのイネーブル化
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

パスワードおよび権限レベルによるスイッチアクセスの制御の設定例

次の項では、パスワードと権限レベルを使用したスイッチアクセスの制御の設定例を示します。

例：スタティック イネーブル パスワードの設定または変更

次の例は、イネーブルパスワードを *l1u2c3k4y5* に変更する方法を示しています。パスワードは暗号化されておらず、レベル15のアクセスが与えられます（従来の特権EXEC モード アクセス）。

```
Device> enable
Device# configure terminal
Device(config)# enable password l1u2c3k4y5
```

例：暗号化によるイネーブルおよびイネーブルシークレットパスワードの保護

次に、権限レベル 2 に対して暗号化パスワード *1FaD0$Xyti5Rkls3LoyxzS8* を設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# enable secret level 2 5 $1$FaD0$Xyti5Rkls3LoyxzS8
```

例：マスクされたシークレットパスワードの設定

次に、マスクされたシークレットパスワードを設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# username cisco masked-secret
Enter secret: ******
Confirm secret: ******
```

次に、コモンクライテリアポリシーのマスクされたシークレットパスワードを設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# username cisco common-criteria-policy test-policy masked-secret
Enter secret: ******
Confirm secret: ******
```

例：端末回線に対する Telnet パスワードの設定

次に、Telnet パスワードを *let45me67in89* に設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# line vty 10
Device(config-line)# password let45me67in89
```

例：コマンドの権限レベルの設定

次の例は、**configure** コマンドを権限レベル 14 に設定し、ユーザがレベル 14 のコマンドを使用する場合に入力するパスワードとして *SecretPswd14* を定義する方法を示しています。

```
Device> enable
Device# configure terminal
Device(config)# line vty 10
Device(config)# privilege exec level 14 configure
Device(config)# enable password level 14 SecretPswd14
```

例：暗号化事前共有キーの設定

以下に、タイプ 6 の事前共有キーに暗号化を行った場合の設定例を示します。この中には、ユーザに対して表示されるプロンプトやメッセージも含まれています。

```
Device> enable
Device# configure terminal
Device(config)# password encryption aes
Device(config)# key config-key password-encrypt
New key:
Confirm key:
Device(config)#
01:46:40: TYPE6_PASS: New Master key configured, encrypting the keys with
the new master key
Device(config)# end
```

スイッチ アクセスのモニタリング

表 *2 : DHCP* 情報を表示するためのコマンド

コマンド	目的
show privilege	権限レベルの設定を表示します。
show running | secret username	ユーザ名が作成され、デフォルトで type9 に暗号化されることを確認します。
show running | secret enable	シークレットパスワードがデフォルトで type9 に暗号化されることを確認します。

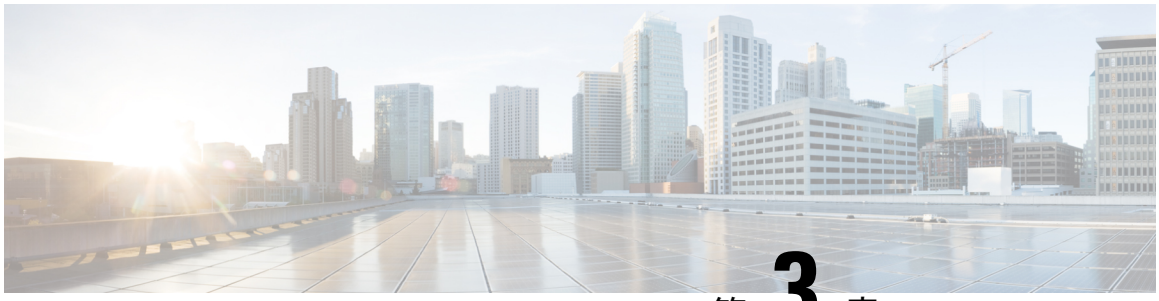
パスワードおよび権限レベルによるスイッチアクセスの制御の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	パスワードおよび権限レベルによるスイッチアクセスの制御	パスワード保護によって、ネットワークまたはネットワーク デバイスへのアクセスが制限されます。権限レベルによって、ネットワーク デバイスにログイン後、ユーザがどのようなコマンドを使用できるかが定義されます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 3 章

TACACS+ の設定

TACACS+は、ユーザによるルータまたはネットワーク アクセス サーバへのアクセス試行の集中的な確認を可能にするセキュリティ アプリケーションです。TACACS+ は、認証および許可プロセスについて詳細なアカウンティング情報と柔軟な管理コントロールを提供します。TACACS+ は、認証、許可、およびアカウンティング（AAA）を通じて効率化され、AAA コマンドでのみ有効化できます。

TACACS+ の前提条件

TACACS+ によるデバイスアクセスのセットアップと設定の前提条件は、次のとおりです（示されている順序で実行する必要があります）。

1. デバイスに TACACS+ サーバアドレスを設定します。
2. 認証キーを設定します。
3. TACACS+ サーバでステップ 2 からキーを設定します。
4. 認証、許可、アカウンティング（AAA）をイネーブルにする。
5. ログイン認証方式リストを作成します。
6. 端末回線にリストを適用します。
7. 認証およびアカウンティング方式のリストを作成します。

TACACS+ によるデバイスアクセス制御のための前提条件は、次のとおりです。

- デバイス上で TACACS+ 機能を設定するには、設定済みの TACACS+ サーバにアクセスする必要があります。また、通常 LINUX または Windows ワークステーション上で稼働する TACACS+ デーモンのデータベースで管理されている TACACS+ サービスにもアクセスする必要があります。
- デバイス上で TACACS+ を使用するには、TACACS+ デーモンソフトウェアが稼働するシステムが必要です。
- TACACS+ を使用するには、それをイネーブルにする必要があります。
- 使用するデバイス上で許可をイネーブルにする必要があります。
- ユーザは TACACS+ 許可に進む前に、まず TACACS+ 認証を正常に完了する必要があります。
- このセクションに記載されている AAA コマンドのいずれかを使用するには、まず **aaa new-model** コマンドを使用して AAA をイネーブルにする必要があります。
- 最低限、TACACS+ デーモンを維持するホスト（1 つまたは複数）を特定し、TACACS+ 認証の方式リストを定義する必要があります。また、任意で TACACS+ 許可およびアカウンティングの方式リストを定義できます。
- 方式リストは実行される認証のタイプと実行順序を定義します。このリストを特定のポートに適用してから、定義済み認証方式を実行する必要があります。唯一の例外はデフォルトの方式リスト（偶然に *default* と名前が付けられている）です。デフォルトの方式リストは、名前付き方式リストを明示的に定義されたインターフェイスを除いて、自動的にすべてのポートに適用されます。定義済みの方式リストは、デフォルトの方式リストに優先します。
- TACACS+ を使用して認証を行った場合は、TACACS+ を使用して特権 EXEC アクセスを許可します。
- 認証に TACACS+ を使用しなかった場合は、ローカル データベースを使用します。

TACACS+ の制約事項

TACACS+ をイネーブルにするには、AAA コマンドを使用する必要があります。

TACACS+ の概要

TACACS+ およびスイッチ アクセス

ここでは、TACACS+ について説明します。TACACS+ は詳細なアカウンティング情報を提供し、認証と許可のプロセスを柔軟に管理します。TACACS+ は、認証、許可、アカウンティン

グ（AAA）機能により拡張されており、TACACS+ をイネーブルにするには AAA コマンドを使用する必要があります。

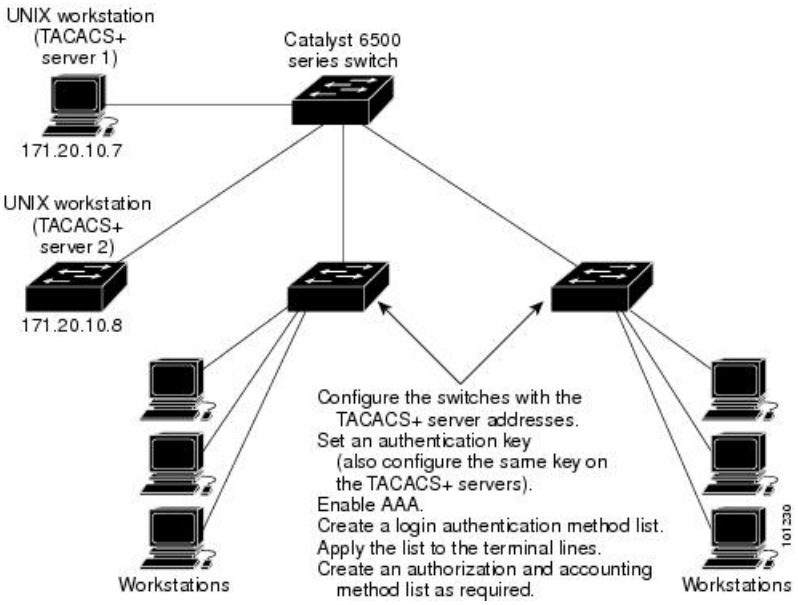
TACACS+ の概要

TACACS+は、スイッチにアクセスしようとするユーザの検証を集中的に行うセキュリティ アプリケーションです。

TACACS+ では、独立したモジュラ型の認証、許可、アカウンティング機能が提供されます。TACACS+ では、単一のアクセス コントロール サーバ（TACACS+ デーモン）が各サービス（認証、許可、およびアカウンティング）を別個に提供します。各サービスを固有のデータベースに結合し、デーモンの機能に応じてそのサーバまたはネットワークで使用できる他のサービスを使用できます。

TACACS+ の目的は、1 つの管理サービスから複数のネットワーク アクセス ポイントを管理する方式を提供することです。スイッチは、他の Cisco ルータやアクセス サーバとともにネットワーク アクセス サーバにできます。

図 *1 :* 一般的な *TACACS+* ネットワーク構成



TACACS+ は、AAA セキュリティ サービスによって管理され、次のようなサービスを提供します。

- 認証：ログインおよびパスワードダイアログ、チャレンジおよび応答、メッセージサポートによって認証の完全制御を行います。

  認証機能は、ユーザとの対話を実行できます（たとえば、ユーザ名とパスワードが入力された後、自宅の住所、母親の旧姓、サービスタイプ、社会保険番号などのいくつかの質問をすることによりユーザを試します）。TACACS+認証サービスは、ユーザ画面にメッセージを送信することもできます。たとえば、会社のパスワード有効期間ポリシーに従い、パスワードの変更の必要があることをユーザに通知することもできます。

- 許可：autocommand、アクセスコントロール、セッション期間、プロトコルサポートの設定といった、ユーザセッション時のユーザ機能についてきめ細かく制御します。また、TACACS+ 認可機能によって、ユーザが実行できるコマンドを制限することもできます。
- アカウンティング：課金、監査、およびレポートに使用する情報を収集して TACACS+ デーモンに送信します。ネットワークの管理者は、アカウンティング機能を使用して、セキュリティ監査のためにユーザの活動状況をトラッキングしたり、ユーザ課金用の情報を提供したりできます。アカウンティング レコードには、ユーザ ID、開始時刻および終了時刻、実行されたコマンド（PPP など）、パケット数、およびバイト数が含まれます。

TACACS+プロトコルは、スイッチと TACACS+デーモン間の認証を行い、スイッチと TACACS+ デーモン間のプロトコル交換をすべて暗号化することによって機密保持を実現します。

TACACS+ の動作

ユーザが、TACACS+ を使用しているデバイスに対して簡易 ASCII ログインを試行し、認証が必要になると、次のプロセスが発生します。

1. 接続が確立されると、デバイスは TACACS+ デーモンに接続してユーザ名プロンプトを取得し、これをユーザに表示します。ユーザがユーザ名を入力すると、デバイスはTACACS+ デーモンに接続してパスワードプロンプトを取得します。デバイスによってパスワードプロンプトが表示され、ユーザがパスワードを入力すると、そのパスワードが TACACS+ デーモンに送信されます。

   TACACS+ によって、デーモンとユーザとの間の対話が可能になり、デーモンはユーザを認証できるだけの情報を取得できるようになります。デーモンは、ユーザ名とパスワードの組み合わせを入力するよう求めますが、ユーザの母親の旧姓など、その他の項目を含めることもできます。

2. デバイスは、最終的に TACACS+ デーモンから次のいずれかの応答を得ます。

   - ACCEPT：ユーザが認証され、サービスを開始できます。許可を必要とするようにデバイスが設定されている場合は、この時点で許可処理が開始されます。
   - REJECT：ユーザは認証されません。TACACS+デーモンに応じて、ユーザはアクセスを拒否されるか、ログイン シーケンスを再試行するように求められます。
   - ERROR：デーモンによる認証サービスのある時点で、またはデーモンとデバイスの間のネットワーク接続においてエラーが発生しました。ERROR 応答が表示された場合は、デバイスは、通常別の方法でユーザを認証しようとします。
   - CONTINUE：ユーザは、さらに認証情報の入力を求められます。

   認証後、デバイスで許可がイネーブルになっている場合、ユーザは追加の許可フェーズに入ります。ユーザは TACACS+ 許可に進む前に、まず TACACS+ 認証を正常に完了する必要があります。

3. TACACS+ 許可が必要な場合は、再び TACACS+ デーモンに接続し、デーモンが ACCEPT または REJECT の許可応答を返します。ACCEPT 応答が返された場合は、その応答に、そ

のユーザおよびそのユーザがアクセスできるサービスの、EXEC または NETWORK セッション宛ての属性の形式でデータが含まれています。

- Telnet、セキュア シェル（SSH）、rlogin、または特権 EXEC サービス
- 接続パラメータ（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、およびユーザ タイムアウトを含む）

方式リスト

方式リストによって、ユーザの認証、許可、またはアカウント維持のための順序と方式を定義します。方式リストを使用して、使用するセキュリティ プロトコルを1つまたは複数指定できるので、最初の方式が失敗した場合のバックアップ システムが確保されます。ソフトウェアは、リスト内の最初の方式を使用してユーザの認証、許可、アカウントの維持を行います。その方式で応答が得られなかった場合、ソフトウェアはそのリストから次の方式を選択します。このプロセスは、リスト内の方式による通信が成功するか、方式リストの方式をすべて試し終わるまで続きます。

VTY 回線で方式リストを設定する場合、対応する方式リストを AAA に追加する必要があります。次の例は、VTY 回線の下に方式リストを設定する方法を示しています。

```
Device# configure terminal
Device(config)# line vty 0 4
Device(config)# authorization commands 15 auth1
```

次の例は、AAA で方式リストを設定する方法を示しています。

```
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authorization commands 15 auth1 group tacacs+
```

VTY 回線で方式リストを設定しない場合、デフォルトの方式リストを AAA に追加する必要があります。次の例は、方式リストを使用しない VTY 設定を示しています。

```
Device# configure terminal
Device(config)# line vty 0 4
```

次の例は、デフォルトの方式リストを設定する方法を示しています。

```
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authorization commands 15 default group tacacs+
```

TACACS の AV ペア

ネットワーク アクセス サーバが TACACS+ 許可機能およびアカウンティング機能を実装するには、各ユーザ セッションで TACACS+ の属性と値（AV）ペアを送受信します。

TACACS+ 認証および認可の AV ペア

次の表で、サポートされている TACACS+ 認証および認可の AV ペアの一覧と説明を示し、実装されている Cisco IOS リリースを指定しています。

表 *3*：サポートされている *TACACS+* 認証および認可の *AV* ペア

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
acl=x	接続アクセス リストを表す ASCII 数。service=shell の場合のみ使用されます。	あり	あり	あり	あり	あり	あり	あり
addr=x	ネットワーク アドレス。service=slip、service=ppp、および protocol=ip で使用されます。SLIP または PPP/IP 経由で接続する際にリモート ホストが使用する IP アドレスを含みます。たとえば、addr=10.2.3.4 となります。	あり	あり	あり	あり	あり	あり	あり
addr-pool=x	リモート ホスト アドレスの取得元とするローカルプールの名前を指定します。service=ppp および protocol=ip と使用されます。 **addr-pool** はローカル プーリングと連動して動作することに注意してください。ローカルプールの名前を指定します。これはネットワーク アクセス サーバで事前設定する必要があります。**ip-local pool** コマンドを使用して、ローカルプールを宣言します。次に例を示します。 ip address-pool local ip local pool boo 10.0.0.1 10.0.0.10 ip local pool moo 10.0.0.1 10.0.0.20 その後、TACACS+ を使用して addr-pool=boo または addr-pool=moo を返し、このリモート ノードのアドレスの取得元にするアドレス プールを指示することができます。	あり	あり	あり	あり	あり	あり	あり
autocmd=x	EXEC 起動時に実行する autocommand を指定します（たとえば autocmd=telnet example.com）。service=shell の場合のみ使用されます。	あり	あり	あり	あり	あり	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
callback- dialstring	コールバックの電話番号（例：callback-dialstring= 408-555-1212）を設定します。値はヌルまたはダイヤルストリングです。ヌル値は、サービスで他の手段を通じてダイヤルストリングを取得することもできることを示します。service=arap、service=slip、service=ppp、service=shell で使用されます。ISDN では無効です。	いいえ（No）	あり	あり	あり	あり	あり	あり
callback-line	コールバックで使用する TTY 回線の数（例：callback-line=4 ）です。service=arap、service=slip、service=ppp、service=shell で使用されます。ISDN では無効です。	いいえ（No）	あり	あり	あり	あり	あり	あり
callback-rotary	コールバックで使用するロータリー グループの数（0 ～ 100 の範囲）です（例：callback-rotary=34）。service=arap、service=slip、service=ppp、service=shell で使用されます。ISDN では無効です。	いいえ（No）	あり	あり	あり	あり	あり	あり
cmd-arg=x	シェル（EXEC）コマンドに渡す引数です。実行されるシェルコマンドの引数を示します。cmd-arg 属性を複数指定でき、順序依存です。 （注） この TACACS+ AV ペアは、RADIUS 属性 26 で使用できません。	あり	あり	あり	あり	あり	あり	あり
cmd=x	シェル（EXEC）コマンドです。実行するシェルコマンドのコマンド名を示します。この属性は、サービスが「シェル」と等しい場合に指定する必要があります。ヌル値は、シェル自身が参照されることを示します。 （注） この TACACS+ AV ペアは、RADIUS 属性 26 で使用できません。	あり	あり	あり	あり	あり	あり	あり
data-service	service=outbound および protocol=ip で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
dial-number	ダイヤルする番号を定義します。service=outbound および protocol=ip で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
dns-servers=	Microsoft PPP クライアントにより、IPCP ネゴシエーション中にネットワーク アクセス サーバから要求される可能性がある DNS サーバ（プライマリまたはセカンダリ）を識別します。service=ppp および protocol=ip で使用されます。DNS サーバを特定する IP アドレスはドット付き 10 進表記で入力します。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
force-56	チャネルの 64 K すべてが使用可能に見える場合でも、ネットワーク アクセス サーバが 56 K の部分のみを使用するかどうかを指定します。この属性をオンにするには、「true」値（force-56=true）を使用します。他の値は、false として扱われます。service=outbound および protocol=ip で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
gw-password	L2F トンネル認証中のホーム ゲートウェイのパスワードを指定します。service=ppp および protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	あり	あり	あり	あり	あり
idletime=x	値を分単位で設定します。その時間が経過すると、アイドルセッションが終了します。ゼロ値はタイムアウトなしを示します。	いいえ（No）	あり	あり	あり	あり	あり	あり
inacl#<n>	現在の接続期間に使用されるインターフェイスにインストールされ適用される、入力アクセス リストの ASCII アクセス リスト識別名です。service=ppp および protocol=ip、service service=ppp および protocol =ipx で使用されます。ユーザ単位のアクセス リストは、現在 ISDN インターフェイスでは使用できません。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
inacl=x	インターフェイス 入力アクセス リストの ASCII 識別名です。service=ppp および protocol=ip で使用されます。ユーザ単位のアクセス リストは、現在 ISDN インターフェイスでは使用できません。	あり	あり	あり	あり	あり	あり	あり
interface-config#<n>	仮想プロファイルを使用してユーザ固有の AAA インターフェイス設定情報を指定します。等号（=）が付いている情報は、すべての Cisco IOS インターフェイス コンフィギュレーション コマンドとして使用できます。この属性は複数インスタンスが許可されますが、各インスタンスは固有の番号を持つ必要があります。service=ppp および protocol=lcp で使用されます。 （注） 「interface-config=」属性はこの属性に置き換えられます。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
ip-addresses	トンネルのエンドポイントで使用できる IP アドレスの、スペースで区切ったリストです。service=ppp および protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	あり	あり	あり	あり	あり
l2tp-busy-disconnect	LNS の vpdn-group で、事前にコピーするよう設定された仮想テンプレートを使用している場合、この属性は、接続先の事前にコピーされたインターフェイスが検索されない、新しい L2TP セッションのディスポジションを制御します。属性が true（デフォルト）の場合、セッションが LNS により切断されます。そうでない場合は、新しいインターフェイスが仮想テンプレートからコピーされます。service=ppp および protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
l2tp-cm-local-window-size	L2TP 制御メッセージの最大受信ウィンドウサイズを指定します。この値は、トンネルの確立中にピアにアドバタイズされます。service=ppp および protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
l2tp-drop-out-of-order	正しくない順序で受信したデータパケットをドロップして、シーケンス番号を順守します。これは受信した場合の処理方法であって、データパケット上でシーケンス番号が送信されるわけではありません。service=ppp および protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
l2tp-hello- interval	hello キープアライブ インターバルの秒数を指定します。ここで指定した秒数、トンネルでデータが送信されないと、hello パケットが送信されます。service=ppp および protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
l2tp-hidden-avp	イネーブルにすると、L2TP制御メッセージで、大文字小文字を区別する AVP にスクランブルがかけられるか、または非表示になります。service=ppp および protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
l2tp-nosession-timeout	タイムアウトおよびシャットダウンまでに、セッションなしでトンネルがアクティブのままになる秒数を指定します。service=ppp および protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
l2tp-tos-reflect	LNS でトンネルに入るパケットに対して、IP ToS フィールドを各ペイロードパケットのIPヘッダーからトンネルパケットの IP ヘッダーにコピーします。service=ppp および protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
l2tp-tunnel- authen	この属性を設定すると、L2TP トンネル認証が実行されます。service=ppp および protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
l2tp-tunnel-password	L2TP トンネル認証および AVP 隠蔽に使用される共有秘密です。service=ppp および protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
l2tp-udp- checksum	これは認可属性で、L2TP がデータ パケットに対して UDP チェックサムを実行する必要があるかどうかを定義します。有効な値は「yes」と「no」です。デフォルトは「no」です。service=ppp と protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
link- compression=	PPP リンクで「stac」圧縮をオンまたはオフのどちらにするかを定義します。service=ppp で使用されます。 リンク圧縮は、次のように、数値で定義します。 • 0：なし • 1：Stac • 2：Stac-Draft-9 • 3：MS-Stac	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
load-threshold=<n>	マルチリンク バンドルに対して他のリンクを追加または削除する発信元の負荷のしきい値を設定します。負荷がこの指定した値を超えると、追加リンクが追加されます。負荷が指定の値を下回ると、リンクが削除されます。service=ppp および protocol=multilink で使用されます。<n> の範囲は、1 から 255 です。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
map-class	ユーザプロファイルに、ダイヤルアウトするネットワーク アクセス サーバ上で同じ名前のマップ クラスで設定される情報の参照を許可します。service=outbound および protocol=ip で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
max-links=<n>	ユーザがマルチリンクで保持できるリンク数を制限します。service=ppp および protocol=multilink で使用されます。<n> の範囲は、1 から 255 です。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
min-links	MLP に対するリンクの最小数を設定します。service=ppp と protocol=multilink、protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
nas-password	L2F トンネル認証でのネットワーク アクセス サーバのパスワードを指定します。service=ppp および protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	あり	あり	あり	あり	あり
nocallback-verify	コールバック検証が必要かを指定します。このパラメータで有効な値は 1 のみです（例：nocallback-verify=1）。service=arap、service=slip、service=ppp、service=shell で使用されます。コールバックに認証がありません。ISDN では無効です。	いいえ（No）	あり	あり	あり	あり	あり	あり
noescape=x	ユーザがエスケープ文字を使用できないようにします。service=shell で使用されます。true または false のどちらかです（例：noescape=true）。	あり	あり	あり	あり	あり	あり	あり
nohangup=x	service=shell で使用されます。nohangup オプションを指定します。このオプションで EXEC シェルの終了後、ユーザに他のログイン（ユーザ名）プロンプトを表示します。true または false のどちらかです（例：nohangup=false）。	あり	あり	あり	あり	あり	あり	あり
old-prompts	プロバイダーが以前のシステム（TACACS および拡張 TACACS）と同じプロンプトを TACACS+ で表示できます。これにより、管理者は、TACACS または拡張 TACACS から TACACS+ に、ユーザが気づくことなくアップグレードできます。	あり	あり	あり	あり	あり	あり	あり
outacl#<n>	現在の状態である限りインターフェイスにインストールされ、適用されるインターフェイス出力アクセス リストの ASCII アクセス リスト識別情報です。service=ppp および protocol=ip、service service=ppp および protocol=ipx で使用されます。ユーザ単位のアクセス リストは、現在 ISDN インターフェイスでは使用できません。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
outacl=x	インターフェイス 出力アクセス リストの ASCII 識別名です。service=ppp および protocol=ip、service service=ppp および protocol=ipx で使用されます。SLIP または PPP/IP の IP 出力アクセス リストが含まれます（outacl=4 など）。このアクセス リスト自身はルータで事前設定する必要があります。ユーザ単位のアクセス リストは、現在 ISDN インターフェイスでは使用できません。	あり（PPP/IP のみ）	あり	あり	あり	あり	あり	あり
pool-def#<n>	ネットワーク アクセス サーバで IP アドレス プールを定義します。service=ppp および protocol=ip で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
pool-timeout=	pool-def とともに、ネットワーク アクセス サーバ上の IP アドレス プールを定義します。IPCP アドレス ネゴシエーション中、IP プール名がユーザに指定されている場合（addr-pool 属性を参照）、指定された名前のプールがネットワーク アクセス サーバで定義されているかチェックされます。その場合、プールに IP アドレスがあるか参照します。service=ppp および protocol=ip で使用されます。	いいえ（No）	いいえ（No）	あり	あり	あり	あり	あり
port-type	ユーザを認証するためにネットワーク アクセス サーバで使用されている物理ポートのタイプを示します。 物理ポートは、次のように数値で示されます。 • 0：非同期 • 1：同期 • 2：ISDN 同期 • 3：ISDN 非同期（V.120） • 4：ISDN-非同期（V.110） • 5：仮想 service=any および protocol=aaa で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
ppp-vj-slot-compression	VJ 圧縮パケットを PPP リンク経由で送信する際に、Cisco ルータでスロット圧縮しないように指示します。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
priv-lvl=x	EXEC に割り当てられる権限レベルです。service=shell で使用されます。権限レベルの範囲は 0 ～ 15 で、15 が最高です。	あり	あり	あり	あり	あり	あり	あり
protocol=x	サービスのサブセットのプロトコルです。たとえば、任意の PPP NCP などです。現在知られている値は、 **lcp**、**ip**、**ipx**、**atalk**、**vines**、**lat**、**xremote**、**tn3270**、**telnet**、 **rlogin**、**pad**、**vpdn**、**osicp**、**deccp**、**ccp**、**cdp**、**bridging**、**xns**、**nbf**、**bap**、**multilink**、および **unknown** です。	あり	あり	あり	あり	あり	あり	あり
proxyacl#<n>	ダウンロード可能なユーザプロファイル（ダイナミック ACL）を、認証プロキシを使用して設定でき、これにより設定されたインターフェイスのトラフィックの通過を許可するよう、認証を設定できます。service=shell および protocol=exec で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
route	インターフェイスに適用されるルートを指定します。service=slip、service=ppp、および protocol=ip で使用されます。 ネットワークの許可中、route 属性はユーザ単位のスタティック ルートの指定に使用でき、TACACS+ により次のようにインストールされます。 route=”*dst_address mask* [*gateway*]” これは、一時的に適用されるスタティック ルートを示します。*dst_address*、*mask*、*gateway* は、通常のドット付き 10 進表記での記述を想定されており、よく使用されるネットワークアクセスサーバの **ip route** コンフィギュレーション コマンドと同じ意味を持ちます。 *gateway* を省略すると、ピアのアドレスがゲートウェイになります。ルートは接続が終了すると消去されます。	いいえ（No）	あり	あり	あり	あり	あり	あり
route#<n>	ルート AV ペアと同様にインターフェイスに適用されるルートを指定しますが、このルートは番号が付けられて複数のルートを適用できます。service=ppp と protocol=ip、および service=ppp と protocol=ipx で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
routing=x	ルーティング情報をインターフェイスに伝播し、このインターフェイスから受け入れるかどうかを指定します。service=slip、service=ppp、および protocol=ip で使用されます。機能上、SLIP および PPP コマンドの /routing フラグと同等です。true または false のいずれか（例：routing=true）です。	あり	あり	あり	あり	あり	あり	あり
rte-fltr-in#<n>	現在の接続中に、現在のインターフェイスのルーティングアップデートにインストールし、適用する入力アクセス リストの定義を指定します。service=ppp と protocol=ip、および service=ppp と protocol=ipx で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
rte-fltr-out#<n>	現在の接続中に、現在のインターフェイスのルーティング アップデートにインストールし、適用する出力アクセス リストの定義を指定します。service=ppp と protocol=ip、および service=ppp と protocol=ipx で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
sap#<n>	接続中にインストールされるスタティック サービス アドバタイジング プロトコル（SAP）エントリを指定します。service=ppp および protocol=ipx で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
sap-fltr-in#<n>	現在の接続中に、現在のインターフェイスにインストールし、適用する入力 SAP フィルタ アクセス リストの定義を指定します。service=ppp および protocol=ipx で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
sap-fltr-out#<n>	現在の接続中に、現在のインターフェイスにインストールし、適用する出力 SAP フィルタ アクセス リストの定義を指定します。service=ppp および protocol=ipx で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
send-auth	CLID 認証に続く、username-password 認証で使用するプロトコル（PAP または CHAP）を定義します。service=any および protocol=aaa で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
send-secret	NAS が発信コールの接続のリモート エンドからの chap/pap 要求に応答する際に必要なパスワードを指定します。service=ppp および protocol=ip で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
service=x	プライマリ サービスです。このサービスの認証またはアカウンティングを要求していることを示すサービス属性を指定します。現在の値は、**slip**、**ppp**、**arap**、**shell**、**tty-daemon**、**connection**、および **system** です。この属性は常に含める必要があります。	あり	あり	あり	あり	あり	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
source-ip=x	VPDN トンネルの一部として生成されたすべての VPDN パケットの発信元 IP アドレスとして使用されます。これは、Cisco **vpdn outgoing** グローバル コンフィギュレーション コマンドと同じ意義を持ちます。	いいえ（No）	いいえ（No）	あり	あり	あり	あり	あり
spi	登録中にホームエージェントがモバイル ノードの認証で必要とする認証情報を伝送します。この情報は、**ip mobile secure host <addr>** コンフィギュレーション コマンドと同じ構文です。基本的に、この文字列に続く残りのコンフィギュレーション コマンドはそのまま含まれます。これにはセキュリティ パラメータ インデックス（SPI）、キー、認証アルゴリズム、認証モード、およびリプレイ保護タイムスタンプ範囲が含まれています。service=mobileip および protocol=ip で使用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
timeout=x	EXEC または ARA セッションを切断するまでの分数です（例：timeout=60）。ゼロ値はタイムアウトなしを示します。service=arap で使用されます。	あり	あり	あり	あり	あり	あり	あり
tunnel-id	個々のユーザ MID が生成されるトンネルの認証に使用するユーザ名を指定します。**vpdn outgoing** コマンドの *remote name* と同様です。service=ppp および protocol=vpdn で使用されます。	いいえ（No）	いいえ（No）	あり	あり	あり	あり	あり
wins-servers=	IPCP ネゴシエーション中に、ネットワーク アクセス サーバから Microsoft PPP クライアントにより要求される可能性がある Windows NT サーバを特定します。service=ppp および protocol=ip で使用されます。各 Windows NT サーバを特定する IP アドレスはドット付き 10 進表記で入力します。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
zonelist=x	数字の zonelist の値です。service=arap で使用されます。ARA 向けの AppleTalk zonelist です（例：zonelist=5）。	あり	あり	あり	あり	あり	あり	あり

TACACS+ と、TACACS+ 認証および認可の設定に使用する資料については、「TACACS+ の設定」モジュールを参照してください。

TACACS アカウンティング AV ペア

次の表で、サポートされている TACACS+ アカウンティングの AV ペアの一覧と説明を示し、実装されている Cisco IOS リリースを指定しています。

表 *4*：サポートされる *TACACS+* アカウンティング *AV* ペア

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
Abort-Cause	ファクスセッションが中断した場合、中断の信号を送信したシステムコンポーネントを示します。中断する可能性のあるシステムコンポーネントには、FAP（Fax Application Process）、TIFF（TIFF リーダーまたは TIFF ライター）、fax-mail クライアント、fax-mail サーバー、ESMTP クライアント、ESMTP サーバーなどがあります。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
bytes_in	この接続中に転送される入力バイト数です。	あり	あり	あり	あり	あり	あり	あり
bytes_out	この接続中に転送される出力バイト数です。	あり	あり	あり	あり	あり	あり	あり
Call-Type	ファクスのアクティビティのタイプを、fax receive または fax send のどちらかで記述します。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
cmd	ユーザが実行したコマンドです。	あり	あり	あり	あり	あり	あり	あり
data-rate	この AV ペアは名前が変更されました。nas-rx-speed を参照してください。							

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
disc-cause	接続がオフラインになった理由を特定します。Disconnect-Cause 属性は、アカウンティング終了記録で送信されます。また、この属性で、認証が実行される前に接続が切断された場合、最初に開始レコードを生成せずに終了レコードが生成されます。Disconnect-Cause 値とその意味の一覧については、次の表（接続解除原因の拡張）を参照してください。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
disc-cause-ext	disc-cause 属性が、接続がオフラインになったベンダー固有の理由をサポートするよう拡張します。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
elapsed_time	処理の経過時間（秒）です。デバイスが実時間を保持していない場合に有用です。	あり	あり	あり	あり	あり	あり	あり
Email-Server-Address	オンランプ fax-mail メッセージを処理する E メール サーバの IP アドレスを示します。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
Email-Server-Ack-Flag	オンランプ ゲートウェイが fax-mail メッセージを受け入れる E メール サーバから肯定確認応答を受信したことを示します。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
event	ルータの状態変化を記述した、アカウンティングパケットに含める情報です。記述されたイベントは、アカウンティング開始およびアカウンティング終了です。	あり	あり	あり	あり	あり	あり	あり
Fax-Account-Id-Origin	**mmoip aaa receive-id** コマンドまたは **mmoip aaa send-id** コマンドについて、アカウント ID の発信元がシステム管理者によって定義されたものとして示します。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
Fax-Auth-Status	このファクスセッションに対する認証が成功したかどうかを示します。このフィールドに対する有効値は、success、failed、bypassed、または unknown です。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
Fax-Connect-Speed	この fax-mail が最初に送信または受信された時点のモデム速度を示します。有効値は、1200、4800、9600、および 14400 です。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
Fax-Coverpage-Flag	カバーページがこのファクスセッションのオフランプ ゲートウェイで生成されたかどうかを示します。true はカバーページが生成されたことを示します。false はカバーページが生成されなかったことを意味します。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
Fax-Dsn-Address	DSN の送信先のアドレスを示します。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
Fax-Dsn-Flag	DSN がイネーブルにされているかどうかを示します。true は DSN がイネーブルにされていることを示します。false は DSN がイネーブルにされていないことを示します。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
Fax-Mdn-Address	MDN の送信先のアドレスを示します。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
Fax-Mdn-Flag	メッセージ配信通知（MDN）がイネーブルにされているかどうかを示します。true は MDN がイネーブルにされていることを示します。false は MDN がイネーブルにされていないことを示します。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
Fax-Modem-Time	モデムがファクスデータを送信した時間（x）、およびファクス セッションの合計時間（y）を秒単位で示します。これには、fax-mail および PSTN 時間が x/y の形式で含まれます。たとえば、10/15 は送信時間が 10 秒で、合計ファクス セッションが 15 秒であったことを示します。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
Fax-Msg-Id=	Store and Forward Fax 機能によって割り当てられた一意のファクス メッセージ識別番号を示します。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
Fax-Pages	このファクスセッション中に送信または受信したページ数を示します。このページ数には、カバーページも含まれます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
Fax-Process-Abort-Flag	ファクスセッションが中断したこと、または正常に終了したことを示します。true はセッションが中断したことを示します。false はセッションが成功したことを示します。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
Fax-Recipient-Count	このファクス送信の受信者数を示します。E メール サーバがセッション モードをサポートするまで、この数字は 1 にする必要があります。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
Gateway-Id	ファクス セッションを処理したゲートウェイの名前を示します。この名前は、hostname.domain-name の形式で表示されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
mlp-links-max	アカウンティング レコードが生成された時点で特定のマルチリンク セッションにあるリンク数を示します。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
mlp-sess-id	セッションが終了した時のマルチリンク バンドルの ID 番号をレポートします。この属性は、マルチリンク バンドルの一部のセッションに適用されます。この属性は、認証応答パケットで送信されます。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
nas-rx-speed	接続のライフタイムでの平均ビット/秒値を指定します。この属性は、アカウンティング終了記録で送信されます。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
nas-tx-speed	2 つのモデムによってネゴシエートされた送信速度を報告します。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
paks_in	この接続中に転送される入力パケット数です。	あり	あり	あり	あり	あり	あり	あり
paks_out	この接続中に転送される出力パケット数です。	あり	あり	あり	あり	あり	あり	あり
port	ユーザがログインしたポートです。	あり	あり	あり	あり	あり	あり	あり
Port-Used	この fax-mail の送受信いずれかに使用される Cisco AS5300 のスロット/ポート番号を示します。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
pre-bytes-in	認証前の入力バイト数を記録します。この属性は、アカウンティング終了記録で送信されます。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
pre-bytes-out	認証前の出力バイト数を記録します。この属性は、アカウンティング終了記録で送信されます。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
pre-paks-in	認証前の入力パケット数を記録します。この属性は、アカウンティング終了記録で送信されます。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
pre-paks-out	認証前の出力パケット数を記録します。Pre-Output-Packets 属性は、アカウンティング終了記録で送信されます。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり

属性	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2
pre-session-time	コールが最初に接続された時から認証が完了した時までの時間長を秒で指定します。	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
priv_level	処理に関連付けられた権限レベルです。	あり	あり	あり	あり	あり	あり	あり
protocol	処理に関連付けられたプロトコルです。	あり	あり	あり	あり	あり	あり	あり
reason	システム変更により発生したイベントを記述した、アカウンティングパケットに含める情報です。記述されるイベントは、システムのリロード、システムのシャットダウン、またはアカウンティングが再設定（オンまたはオフ）された場合です。	あり	あり	あり	あり	あり	あり	あり
service	ユーザが使用するサービスです。	あり	あり	あり	あり	あり	あり	あり
start_time	処理を開始する時刻（エポック（1970 年 1 月 1 日 12:00 a.m.）からの秒数で指定）です。この情報を受信するよう、クロックを設定する必要があります。	あり	あり	あり	あり	あり	あり	あり
stop_time	処理を停止する時刻（エポックからの秒数で指定）です。この情報を受信するよう、クロックを設定する必要があります。	あり	あり	あり	あり	あり	あり	あり
task_id	同じ（一意の）task_id 番号を持つ同じイベントに対する開始レコードと終了レコードです。	あり	あり	あり	あり	あり	あり	あり
timezone	このパケットに含まれるすべてのタイムスタンプの時間帯（省略形）です。	あり	あり	あり	あり	あり	あり	あり
xmit-rate	この AV ペアは名前が変更されました。nas-tx-speed を参照してください。							

次の表で、Disconnect Cause Extended（disc-cause-ext）属性の原因のコードと説明の一覧を示しています。

表 *5*: 接続解除原因の拡張

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1000 – 理由なし	接続解除の理由はありません。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1001 – 接続解除なし	イベントは接続解除されませんでした。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1002 – 不明	接続解除の理由が不明です。このコードは、リモート接続が停止している場合に表示されることがあります。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1003 – コール接続解除	コールが接続解除されました。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1004 – CLID 認証失敗	Calling line ID（CLID）認証が失敗しました。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1009 – モデム使用不可	モデムが使用できません。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1010 – キャリアなし	モデムで、データキャリア検出（DCD）が検出されませんでした。このコードは、最初のモデム接続で切断が発生した場合に表示されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1011 – キャリアのロスト	モデムで DCD は検出されましたが、非アクティブになっています。このコードは、最初のモデム接続で切断が発生した場合に表示されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1012 – モデム結果なし	結果コードが解析できません。このコードは、最初のモデム接続で切断が発生した場合に表示されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1020 – TS ユーザ退出	ユーザがターミナルサーバから正常に退出しました。このコードは、ターミナル サーバ セッション中のイミディエート Telnet および raw TCP の切断に関連しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1021 – アイドル タイムアウト	アイドルタイマーの時間切れのため、ターミナル サーバからユーザが退出しました。このコードは、ターミナル サーバ セッション中のイミディエート Telnet および raw TCP の切断に関連しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1022 – TS Telnet 退出	ユーザが、Telnet セッションから正常に退出しました。このコードは、ターミナル サーバ セッション中のイミディエート Telnet および raw TCP の切断に関連しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1023 – TS IP アドレスなし	リモート ホストが IP アドレスを保持していないか、ダイナミック プールが割り当てられていないため、ユーザはシリアルライン インターネット プロトコル（SLIP）または PPP にスイッチできませんでした。このコードは、ターミナル サーバ セッション中のイミディエート Telnet および raw TCP の切断に関連しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1024 – TS TCP の raw 退出	ユーザが、 raw TCP セッションから正常に退出しました。このコードは、ターミナル サーバ セッション中のイミディエート Telnet および raw TCP の切断に関連しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1025 – TS パスワード不良	ユーザが 3 回、正しいパスワードの入力に失敗したため、ログイン処理が終了しました。このコードは、ターミナル サーバ セッション中のイミディエート Telnet および raw TCP の切断に関連しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1026 – TS raw TCP なし	raw TCP オプションがイネーブルになっていません。このコードは、ターミナル サーバ セッション中のイミディエート Telnet および raw TCP の切断に関連しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1027 – TS CNTL-C	ユーザが「Ctrl C」と入力したためログインプロセスが終了しました。このコードは、ターミナル サーバ セッション中のイミディエート Telnet および raw TCP の接続解除に関連しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1028 – TS セッション終了	ターミナル サーバ セッションが終了しました。このコードは、ターミナルサーバセッション中のイミディエート Telnet および raw TCP の切断に関連しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1029 – TS Vconn 終了	ユーザがバーチャル コネクションを終了しました。このコードは、ターミナルサーバセッション中のイミディエート Telnet および raw TCP の切断に関連しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1030 – TS Vconn 終了	バーチャル コネクションが終了しました。このコードは、ターミナル サーバ セッション中のイミディエート Telnet および raw TCP の切断に関連しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1031 – TS Rlogin 退出	ユーザが Rlogin セッションから正常に退出しました。このコードは、ターミナル サーバ セッション中のイミディエート Telnet および raw TCP の切断に関連しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1032 – TS Rlogin オプション無効	ユーザが無効な Rlogin オプションを選択しました。このコードは、ターミナルサーバセッション中のイミディエート Telnet および raw TCP の切断に関連しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1033 – TS 不十分なリソース	アクセス サーバにターミナル サーバセッションを行う十分なリソースがありません。このコードは、ターミナル サーバ セッション中のイミディエート Telnet および raw TCP の切断に関連しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1040 – PPP LCP タイムアウト	PPP リンク コントロール プロトコル（LCP）ネゴシエーションがピアからの応答を待機している間にタイム アウトしました。このコードは、PPP 接続と関係しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1041 – PPP LCP 失敗	PPP LCP ネゴシエーションで収束に失敗しました。このコードは、PPP 接続と関係しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1042 – PPP Pap 失敗	PPP パスワード認証プロトコル（PAP）認証が失敗しました。このコードは、PPP 接続と関係しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1043 – PPP CHAP 失敗	PPP チャレンジ ハンドシェーク認証プロトコル（CHAP）認証が失敗しました。このコードは、PPP 接続と関係しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1044 – PPP リモート失敗	リモートサーバからの認証が失敗しました。このコードは、PPP セッションと関係しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1045 – PPP 終了の受信	ピアが PPP 終了要求を送信しました。このコードは、PPP 接続と関係しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
PPP LCP 終了（1046）	LCP がオープン状態にある時に、LCP が上位層から終了要求を受信しました。このコードは、PPP 接続と関係しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1047 – PPP NCP なし	NCP がオープンでないため、LCP が終了しました。このコードは、PPP 接続と関係しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1048 – PPP MP エラー	ユーザに追加するマルチリンク PPP バンドルを特定できなかったため、LCP は終了しました。このコードは、PPP 接続と関係しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1049 – PPP 最大チャネル	アクセスサーバが MP セッションにこれ以上チャネルを追加できなかったため、LCP が終了しました。このコードは、PPP 接続と関係しています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1050 – TS テーブルが満杯	raw TCP または Telnet 内部セッションテーブルが満杯です。このコードは、イミディエート Telnet および raw TCP の切断に関連し、この表の前のほうに記載した Telnet および TCP コードよりも詳細な情報が含まれています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1051 – TS リソースが満杯	内部リソースが満杯です。このコードは、イミディエート Telnet および raw TCP の切断に関連し、この表の前のほうに記載した Telnet および TCP コードよりも詳細な情報が含まれています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1052 – TS 無効な IP アドレス	Telnet ホストの IP アドレスが無効です。このコードは、イミディエート Telnet および raw TCP の切断に関連し、この表の前のほうに記載した Telnet および TCP コードよりも詳細な情報が含まれています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1053 – TS ホスト名不良	アクセスサーバがホスト名を解決できませんでした。このコードは、イミディエート Telnet および raw TCP の切断に関連し、この表の前のほうに記載した Telnet および TCP コードよりも詳細な情報が含まれています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1054 – TS ポート不良	アクセスサーバが不良または欠落したポート番号を検出しました。このコードは、イミディエート Telnet および raw TCP の切断に関連し、この表の前のほうに記載した Telnet および TCP コードよりも詳細な情報が含まれています。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1060 – TCP リセット	ホストで TCP 接続がリセットされました。TCP スタックが、イミディエート Telnet または raw TCP セッション中に、この切断コードを返す場合があります。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1061 – TCP 接続拒否	ホストで TCP 接続が拒否されました。TCP スタックが、イミディエート Telnet または raw TCP セッション中に、この切断コードを返す場合があります。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1062 – TCP タイムアウト	TCP接続がタイムアウトしました。TCPスタックが、イミディエート Telnet または raw TCP セッション中に、この切断コードを返す場合があります。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1063 – TCP 外部ホストの終了	外部ホストでTCP接続が終了しました。TCPスタックが、イミディエート Telnet または raw TCPセッション中に、この切断コードを返す場合があります。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1064 – TCP ネット到達不能	TCPネットワークが到達不能でした。TCP スタックが、イミディエート Telnet または raw TCPセッション中に、この切断コードを返す場合があります。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1065 – TCP ホスト到達不能	TCPホストが到達不能でした。TCPスタックが、イミディエート Telnet または raw TCP セッション中に、この切断コードを返す場合があります。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1066 – TCP ネット管理到達不能	TCPネットワークが管理的に到達不能でした。TCP スタックが、イミディエート Telnet または raw TCP セッション中に、この切断コードを返す場合があります。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1067 – TCP ホスト管理到達不能	TCPホストが管理的に到達不能でした。TCPスタックが、イミディエート Telnet または raw TCPセッション中に、この切断コードを返す場合があります。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1068 – TCP ポート到達不能	TCPポートが到達不能でした。TCPスタックが、イミディエート Telnet または raw TCP セッション中に、この切断コードを返す場合があります。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1100 – セッション タイムアウト	PPP リンクでアクティビティがないため、セッションがタイムアウトしました。このコードは、すべてのセッションタイプに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1101 – セキュリティ障害	セキュリティ上の理由によりセッションが失敗しました。このコードは、すべてのセッション タイプに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1102 – コールバック	コールバックのためセッションが終了しました。このコードは、すべてのセッションタイプに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1120 – 非サポート	プロトコルがディセーブルまたは非サポートのため、片側がコールを拒否しました。このコードは、すべてのセッションタイプに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1150 – Radius 接続解除	RADIUS サーバが接続解除を要求しました。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1151 – ローカル管理者接続解除	ローカル管理者が接続解除しました。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1152 – SNMP 接続解除	簡易ネットワーク管理プロトコル（SNMP）が接続解除しました。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1160 – V110 リトライ	V110 同期で許可されたリトライ回数を超えました。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1170 – PPP 認証タイムアウト	認証がタイムアウトしました。このコードは、PPP セッションに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1180 – ローカル ハングアップ	ローカルがハングアップした結果、コールが接続解除しました。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1185 – リモート ハングアップ	リモートエンドがハングアップしたため、コールが接続解除しました。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1190 – T1 休止	伝送している T1 回線が休止したため、コールが接続解除しました。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1195 – コール期間	コール期間が、アクセス サーバの Max Call Mins または Max DS0 Mins パラメータで許可された時間を越えたため、コールが接続解除しました。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり	あり	あり
1600 - VPDN ユーザ接続解除	ユーザが接続解除しました。この値は、バーチャル プライベート ダイヤルアップ ネットワーク（VPDN）セッションに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
1601 - VPDN 搬送波消失	搬送波消失が発生しました。このコードは、VPDN セッションに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
1602 – VPDN リソースなし	リソースがありません。このコードは、VPDN セッションに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
1603 – VPDN 制御パケット不良	制御パケットが無効です。このコードは、VPDN セッションに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
1604 – VPDN 管理者接続解除	管理者が接続解除しました。このコードは、VPDN セッションに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
1605 – VPDN トンネル ダウン/確立失敗	トンネルがダウンしているか、確立に失敗しました。このコードは、VPDN セッションに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
1606 – VPDN ローカル PPP 接続解除	ローカル PPP が接続解除しました。このコードは、VPDN セッションに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1607 – VPDN ソフト停止/セッション制限	VPN トンネルで新しいセッションを確立できませんでした。このコードは、VPDN セッションに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
1608 – VPDN コール リダイレクト	コールがリダイレクトされました。このコードは、VPDN セッションに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり	あり
1801 – Q850 未割り当て番号	番号が割り当てられていません。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1802 – Q850 ルートなし	このコードを送信している機器が、認識されていない特定の中継ネットワークを使用したコールのルート要求を受信しました。このコードを送信している機器は、その中継ネットワークが存在しないか、その特定の中継ネットワークが存在していても、このコードを送信している機器で機能していないため、中継ネットワークを認識していません。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1803 – Q850 宛先へのルートなし	コールが選択した経路で通過するネットワークが、目的の宛先で機能していないため、着信側に到達できません。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1806 – Q850 チャネル受け入れ不能	直近で識別されたチャネルがこのコールで使用する送信エンティティに受け入れられません。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1816 – Q850 正常な消去	このコールに関係するユーザの誰かが、コールを消去するよう要求したためコールが消去されました。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1817 – Q850 ユーザ ビジー	ユーザビジー状態になっているため、着信側が他のコールを受けられません。このコードは、着信側のユーザまたはネットワークで生成されることがあります。ユーザにより生成された場合、ユーザの機器がこのコールに対応できます。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1818 – Q850 ユーザ応答なし	割り当てられた所定の時間内に、着信側が、コール確立メッセージに対してアラートまたは接続表示によって応答しないときに使用されます。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1819 – Q850 ユーザ応答なし	着信側にアラートが送信されましたが、所定の時間内に接続表示による応答がありません。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1821 – Q850 コール却下	このコードを送信している機器は、ビジーまたは非対応ではないためこのコールを受けられますが、このコールを受けたくありません。このコードはネットワークにより生成されることもあり、この場合、このコールが補足サービスの制約により消去されたことを示します。診断フィールドには、補足サービスの追加情報や却下の理由が含まれている場合があります。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1822 – Q850 番号の変更	着信側を示す番号が割り当てられていません。新しい着番号が、任意で診断フィールドに含まれている場合があります。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1827 – Q850 宛先故障	宛先へのインターフェイスが正常に機能していないため、ユーザが指示した宛先に到達できません。「正常に機能していない」とは、シグナリング メッセージをリモート側に配信できなかったことを意味しています。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1828 – Q850 無効な番号形式	着番号が有効な形式でないか、完全でないため、着信側に到達できません。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1829 – Q850 ファシリティ拒否	このコードは、ユーザが要求した補足サービスがネットワークで提供されていない場合に返されます。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1830 – Q850 状態問い合わせへの応答	このコードは、STATUS ENQUIRY メッセージよりも先に受領したために STATUS メッセージが生成された場合に、STATUS メッセージに含まれています。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1831 – Q850 未指定の原因	他のコードが適用されない場合に適用されます。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1834 – Q850 使用可能な回線なし	コールを処理できる回線またはチャネルがありません。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1838 – Q850 ネットワーク障害	ネットワークが正常に機能しておらず、この状態が比較的長期間続く見込みです。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1841 – Q850 一時障害	ネットワークが正常に機能していませんが、この状態は長期間続かない見込みです。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1842 – Q850 ネットワーク輻輳	ネットワークが輻輳しています。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1843 – Q850 アクセス情報破棄	このコードは、ネットワークがアクセス情報をリモートユーザの要求に従って配信できなかったことを示します。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1844 – Q850 要求チャネルが使用不可能	このコードは、要求エンティティにより指定された回線またはチャネルが、インターフェイスの片側から提供できなかった場合に返されます。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1845 – Q850 コール プリエンプション	コールがプリエンプションされました。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1847 – Q850 リソースが使用不可能	このコードは、リソース使用不可クラスの他のコードが適用されない場合にのみ、リソース使用不可イベントをレポートするために使用されます。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1850 – Q850 未登録ファシリティ	登録されているファシリティではありません。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1852 – Q850 発信コール除外	発信側が、発信非公開ユーザ グループ コールで非公開ユーザ グループのメンバーであっても、このメンバーに対して発信コールが許可されていません。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
Q850 着信コール除外（1854）	着信側が、着信非公開ユーザ グループ コールで非公開ユーザ グループのメンバーであっても、このメンバーに対して着信コールが許可されていません。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1858 – Q850 ベアラー機能が使用不可	ユーザが、このコードを生成した機器に実装されているベアラー機能を要求しましたが、その時点で使用できませんでした。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1863 – Q850 サービス使用不可	このコードは、サービスまたはオプション使用不可クラスの他のコードが適用されない場合にのみ、サービスまたはオプション使用不可イベントのレポートに使用されます。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1865 – Q850 ベアラー機能未実装	このコードを送信した機器は、要求されたベアラ機能をサポートしていません。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1866 – Q850 チャネル未実装	このコードを送信した機器は、要求されたチャネル タイプをサポートしていません。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1869 – Q850 ファシリティ未実装	ユーザが要求した補足サービスがネットワークで提供できません。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1881 – Q850 無効コール参照値	このコードを送信した機器は、ユーザネットワーク インターフェイスで現在使用されていないコール参照値が含まれたメッセージを受信しました。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1882 – Q850 チャネルが存在しない	直近で識別されたチャネルがこのコールで使用する送信エンティティに受け入れられません。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1888 – Q850 互換性がない宛先	このコードを送信中の機器が、対応できない下位レイヤの互換性または他の互換性属性を持つコールを確立するよう要求されました。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1896 – Q850 必須情報要素が喪失	このコードを送信中の機器が、メッセージが処理される前にメッセージに存在しなければならない情報要素が失われているメッセージを受信しました。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1897 – Q850 存在しないメッセージ タイプ	このコードを送信中の機器が、定義されていないメッセージであるか、定義されてはいるがこのコードを送信した機器で実装されていないため認識されないメッセージ タイプのメッセージを受信しました。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1898 – Q850 無効なメッセージ	このコードは、無効なメッセージクラスの他のコードが適用されない場合に無効なメッセージをレポートするために使用されます。このコードは、ISDN またはISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1899 – Q850 情報要素不良	情報要素が認識されません。このコードは、ISDN またはISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1900 – Q850 無効要素が含まれる	このコードを送信中の機器が、未実装の情報要素を受信しました。ただし、この情報要素の 1 つまたは複数のフィールドがこのコードを送信した機器で実装されていない方法で符号化されています。このコードは、ISDN またはISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1901 – Q850 誤った状態のメッセージ	受信したメッセージは、コール ステートと互換性がありません。このコードは、ISDN またはISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1902 – Q850 タイマーの期限切れからの回復	エラー処理手順に関連付けられたタイマーの期限切れによって、手順が初期化されました。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり

原因コード	説明	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1903 – Q850 情報要素エラー	このコードを送信中の機器が、情報要素識別名またはパラメータ名が定義されていないか、定義されてはいるがこのコードを送信した機器で実装されていないため、認識されない情報要素またはパラメータが含まれるメッセージを受信しました。このコードは、ISDN またはISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1911 – Q850 プロトコル エラー	このコードは、プロトコル エラー クラスの他のコードが適用されない場合にのみ、プロトコル エラー イベントをレポートするために使用されます。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり
1927 – Q850 未指定のインターネットワーキング イベント	行った処理に対してコードを提供しないネットワークでインターワーキングした場合にエラーになります。このコードは、ISDN または ISDN 経由のモデム コールに適用されます。	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	いいえ（No）	あり

TACACS+ 設定オプション

認証用に 1 つのサーバを使用するように設定することも、認証用に既存のサーバホストをグループ化するために AAA サーバグループを使用するように設定することもできます。サーバをグループ化して設定済みサーバホストのサブセットを選択し、特定のサービスにそのサーバを使用できます。サーバ グループは、グローバル サーバ ホスト リストとともに使用され、選択されたサーバ ホストの IP アドレスのリストが含まれています。

TACACS+ ログイン認証

方式リストは、ユーザ認証のためクエリ送信を行う手順と認証方式を記述したものです。認証に使用する1つまたは複数のセキュリティ プロトコルを指定できるので、最初の方式が失敗した場合のバックアップ システムが確保されます。ソフトウェアは、リスト内の最初の方式を使用してユーザを認証します。その方式で応答が得られなかった場合、ソフトウェアはそのリス

トから次の認証方式を選択します。このプロセスは、リスト内の認証方式による通信が成功するか、定義された方式をすべて試し終わるまで繰り返されます。この処理のある時点で認証が失敗した場合（つまり、セキュリティサーバまたはローカルのユーザ名データベースがユーザアクセスを拒否すると応答した場合）、認証プロセスは停止し、それ以上認証方式が試行されることはありません。

特権 EXEC アクセスおよびネットワーク サービス用の TACACS+ 許可

AAA 許可によってユーザが使用できるサービスが制限されます。AAA 許可がイネーブルに設定されていると、デバイスはユーザのプロファイルから取得した情報を使用します。このプロファイルは、ローカルのユーザデータベースまたはセキュリティサーバ上にあり、ユーザのセッションを設定します。ユーザは、ユーザプロファイル内の情報で認められている場合に限り、要求したサービスのアクセスが認可されます。

TACACS+ 認証

TACACS+ デーモンを指定し、関連する TACACS+ 暗号キーを定義したら、TACACS+ 認証の方式リストを定義する必要があります。TACACS+ 認証は AAA を介して実行されるため、認証方式として TACACS+ を指定して、**aaa authentication** コマンドを発行する必要があります。

TACACS+ 許可

AAA 許可により、ユーザによるネットワーク アクセスを制限するパラメータを設定することができます。TACACS+ を介する許可は、コマンド、ネットワーク接続、および EXEC セッションに適用できます。AAA によって TACACS+ 許可が容易になるため、認可方式として TACACS+ を指定して、**aaa authorization** コマンドを発行する必要があります。

TACACS+ Accounting

AAA アカウンティング機能は、ユーザがアクセスしたサービスと、消費したネットワーク リソース量をトラッキングします。AAA アカウンティングをイネーブルにすると、デバイスはユーザの活動状況をアカウンティングレコードの形式で TACACS+ セキュリティサーバに報告します。各アカウンティング レコードにはアカウンティングの Attribute-Value（AV）ペアが含まれ、レコードはセキュリティ サーバに格納されます。このデータを、ネットワーク管理、クライアント請求、または監査のために分析できます。

TACACS+ のデフォルト設定

TACACS+ および AAA は、デフォルトではディセーブルに設定されています。

セキュリティの失効を防止するため、ネットワーク管理アプリケーションを使用して TACACS+ を設定することはできません。TACACS+ をイネーブルに設定した場合、CLI を通じてスイッチにアクセスするユーザを認証できます。

（注） TACACS+ の設定は CLI を使用して行いますが、TACACS+ サーバは権限レベル 15 に設定された HTTP 接続を許可します。

TACACS+ を設定する方法

TACACS+ サーバ ホストの指定および認証キーの設定

TACACS+ サーバ ホストを特定し、認証キーを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**tacacs server** *server-name* 例： `Device(config)# tacacs server yourserver`	TACACS+ サーバを維持する IP ホストを特定します。このコマンドを複数回入力して、優先ホストのリストを作成します。ソフトウェアは、指定された順序でホストを検索します。 *server-name* にはサーバ名を指定します。
ステップ 4	**address ipv4** *ip address* 例： `Device(config-server-tacacs)# address ipv4 10.0.1.12`	TACACS サーバの IP アドレスを設定します。
ステップ 5	**exit** 例： `Device(config-server-tacacs)# exit`	TACACS サーバモードを終了して、グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 6	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をイネーブルにします。
ステップ 7	**aaa group server tacacs+** *group-name* 例： `Device(config)# aaa group server tacacs+ your_server_group`	（任意）グループ名で AAA サーバグループを定義します。 このコマンドによって、deviceをサーバ グループ サブコンフィギュレーション モードにします。
ステップ 8	**server** *ip-address* 例： `Device(config)# server 10.1.2.3`	（任意）特定の TACACS+ サーバを定義済みサーバ グループに関連付けます。AAA サーバ グループの TACACS+ サーバごとに、このステップを繰り返します。
ステップ 9	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

TACACS+ ログイン認証の設定

TACACS+ ログイン認証を設定するには、次の手順を実行します。

始める前に

AAA 認証を設定するには、認証方式の名前付きリストを作成してから、各種ポートにそのリストを適用します。

（注） AAA 方式を使用して HTTP アクセスに対しデバイスのセキュリティを確保するには、**ip http authentication aaa** グローバル コンフィギュレーション コマンドでデバイスを設定する必要があります。AAA 認証を設定しても、AAA 方式を使用した HTTP アクセスに対しデバイスのセキュリティは確保されません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をイネーブルにします。
ステップ 4	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] 例： `Device(config)# aaa authentication login default tacacs+ local`	ログイン認証方式リストを作成します。 • **login authentication** コマンドにリストが指定されていない場合に使用するデフォルトのリストを作成するには、**default** キーワードの後ろにデフォルト状況で使用される方式を指定します。デフォルトの方式リストは、自動的にすべてのポートに適用されます。 • *list-name* には、作成するリストの名前として使用する文字列を指定します。 • *method1...* には、認証アルゴリズムが試行する実際の方式を指定します。追加の認証方式は、その前の方式でエラーが返された場合に限り使用されます。前の方式が失敗した場合は使用されません。 次のいずれかの方式を選択します。 • *enable*：イネーブルパスワードを認証に使用します。この認証方式を使用するには、あらかじめ **enable** *password* グローバル コンフィギュ

	コマンドまたはアクション	目的
		レーション コマンドを使用してイネーブルパスワードを定義しておく必要があります。 • *group tacacs+*：TACACS+ 認証を使用します。この認証方式を使用するには、あらかじめ TACACS+ サーバを設定しておく必要があります。 • *line*：回線パスワードを認証に使用します。この認証方式を使用するには、あらかじめ回線パスワードを定義しておく必要があります。**password** *password* ライン コンフィギュレーション コマンドを使用します。 • *local*：ローカルユーザ名データベースを認証に使用します。データベースにユーザ名情報を入力しておく必要があります。**username** *password* 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。 • *local-case*：大文字と小文字が区別されるローカルユーザ名データベースを認証に使用します。**username** *name* **password** グローバル コンフィギュレーション コマンドを使用して、ユーザ名情報をデータベースに入力する必要があります。 • *none*：ログインに認証を使用しません。
ステップ 5	**line** [**console** | **tty** | **vty**] *line-number* [*ending-line-number*] 例： `Device(config)# line 2 4`	ライン コンフィギュレーション モードを開始し、認証リストを適用する回線を設定します。
ステップ 6	**login authentication** {**default** | *list-name*} 例： `Device(config-line)# login`	1 つの回線または複数回線に認証リストを適用します。 • **default** を指定する場合は、**aaa authentication login** コマンドで作成

	コマンドまたはアクション	目的
	authentication default	したデフォルトのリストを使用します。 • *list-name* には、**aaa authentication login** コマンドで作成したリストを指定します。
ステップ 7	**end** 例： `Device(config-line)# end`	特権 EXEC モードに戻ります。

特権 EXEC アクセスおよびネットワーク サービス用の TACACS+ 許可の設定

aaa authorization グローバル コンフィギュレーション コマンドと **tacacs+** キーワードを使用すると、ユーザのネットワークアクセスを特権EXECモードに制限するパラメータを設定できます。

（注） 許可が設定されていても、CLI を使用してログインし、認証されたユーザに対しては、許可は省略されます。

特権 EXEC アクセスおよびネットワーク サービスに関する TACACS+ 許可を指定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**aaa authorization network** *authorization-list* **tacacs+** 例： `Device(config)# aaa authorization network list1 tacacs+`	ネットワーク関連のすべてのサービス要求に対して、ユーザが TACACS+認可を受けるようにデバイスを設定します。
ステップ 4	**aaa authorization exec default tacacs+** 例： `Device(config)# aaa authorization exec default tacacs+`	ユーザの特権 EXEC アクセスに対してユーザ TACACS+認可を行うことを設定します。 **exec** キーワードを指定すると、ユーザプロファイル情報（**autocommand** 情報など）が返される場合があります。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

TACACS+ アカウンティングの起動

TACACS+ アカウンティングを開始するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa accounting network** *authorization-list* **start-stop tacacs+** 例： `Device(config)# aaa accounting network list1 start-stop tacacs+`	ネットワーク関連のすべてのサービス要求について、TACACS+アカウンティングをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 4	**aaa accounting exec default start-stop tacacs+** 例： `Device(config)# aaa accounting exec default start-stop tacacs+`	TACACS+アカウンティングをイネーブルにして、特権 EXEC プロセスの最初に記録開始アカウンティング通知、最後に記録停止通知を送信します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

次のタスク

AAA サーバが到達不能な場合にルータとのセッションを確立するには、**aaa accounting system guarantee-first** コマンドを使用します。これは、最初のレコードとしてシステム アカウンティングを保証します（これがデフォルトの条件です）。場合によっては、システムがリロードされるまでコンソールまたは端末接続でセッションを開始できない場合があります。システムのリロードにかかる時間は 3 分を超えることがあります。

ルータのリロード時に AAA サーバが到達不能な場合に、ルータとのコンソールセッションまたは Telnet セッションを確立するには、**no aaa accounting system guarantee-first** コマンドを使用します。

AAA サーバが到達不能な場合のルータとのセッションの確立

aaa accounting system guarantee-first コマンドは、システムアカウントを最初のレコードとして保証します。これは、デフォルトの条件です。場合によっては、システムがリロードされるまでコンソールまたは端末接続でセッションを開始できない場合があります。システムのリロードにかかる時間は 3 分を超えることがあります。

ルータのリロード時に AAA サーバが到達不能な場合に、ルータとのコンソールセッションまたは Telnet セッションを確立するには、**no aaa accounting system guarantee-first** コマンドを使用します。

TACACS+ のモニタリング

表 *6 : TACACS+* 情報を表示するためのコマンド

コマンド	目的
show tacacs	TACACS+ サーバの統計情報を表示します。

TACACS+ の設定例

例：TACACS 認可

次に、デフォルトの方式リストを使用して、PPP 認証用のセキュリティ プロトコルとして、TACACS+ を設定する例を示します。また、TACACS+ を介してネットワークの許可を設定する方法も示します。

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
aaa authorization network default group tacacs+
tacacs server server1
address IPv4 10.1.2.3
key goaway
exit

interface gigabitethernet 1/0/1
 switchport mode access
```

前述の設定例の回線は、次のように定義されます。

- **aaa new-model** コマンドは、AAA セキュリティ サービスをイネーブルにします。
- **aaa authentication** コマンドにより、PPP を実行するシリアル インターフェイスに使用する方式リスト「default」を定義します。キーワード **default** は、デフォルトですべてのインターフェイスに PPP 認証が適用されることを示します。**if-needed** キーワードは、ユーザが ASCII ログイン手順を介してすでに認証済みの場合には PPP 認証が不要なのでスキップできることを示します。認証が必要な場合、キーワード **group tacacs+** は、TACACS+ を介して認証が実行されることを示します。認証中に TACACS+ から何らかのエラーが返される場合、キーワード **local** は、ネットワーク アクセス サーバ上のローカル データベースを使用して認証が試行されることを示します。
- **aaa authorization** コマンドにより、TACACS+ を介するネットワークの許可を設定します。認証リストとは異なり、この許可リストは、ネットワーク アクセス サーバに対するすべての着信ネットワーク接続に常に適用されます。
- **tacacs server** コマンドにより TACACS+ デーモンが識別され、**address ipv4** コマンドにより 10.1.2.3 の IP アドレスが指定されます。**key** コマンドにより、共有暗号キーを「goaway」に定義します。
- **interface** コマンドで回線を選択します。**ppp authentication** コマンドは、デフォルト方式リストをこの回線に適用します。

例：TACACS アカウンティング

次に、デフォルトの方式リストを使用して、PPP 認証用のセキュリティ プロトコルとして、TACACS+ を設定する例を示します。また、TACACS+ を介してアカウンティングを設定する方法も示します。

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
aaa accounting network default stop-only group tacacs+
tacacs server server1
address IPv4 10.1.2.3
key goaway
exit
interface serial 0
 ppp authentication chap default
```

前述の設定例の回線は、次のように定義されます。

- **aaa new-model** コマンドは、AAA セキュリティ サービスをイネーブルにします。
- **aaa authentication** コマンドにより、PPP を実行するシリアル インターフェイスに使用する方式リスト「default」を定義します。キーワード**default**は、デフォルトですべてのインターフェイスに PPP 認証が適用されることを示します。**if-needed** キーワードは、ユーザが ASCII ログイン手順を介してすでに認証済みの場合には PPP 認証が不要なのでスキップできることを示します。認証が必要な場合、キーワード**group tacacs+**は、TACACS+ を介して認証が実行されることを示します。認証中に TACACS+ から何らかのエラーが返される場合、キーワード**local**は、ネットワーク アクセス サーバ上のローカル データベースを使用して認証が試行されることを示します。
- **aaa accounting** コマンドにより、TACACS+ を介するネットワーク アカウンティングを設定します。この例では、ネットワーク接続が終了するたびに、終了したセッションについて説明するアカウンティング レコードが、TACACS+ デーモンに送信されます。
- **tacacs server** コマンドにより TACACS+ デーモンが識別され、**address ipv4** コマンドにより 10.1.2.3 の IP アドレスが指定されます。**key** コマンドにより、共有暗号キーを「goaway」に定義します。
- **interface** コマンドで回線を選択します。**ppp authentication** コマンドは、デフォルト方式リストをこの回線に適用します。

例：TACACS 認証

次に、PPP 認証に使用するセキュリティ プロトコルとして TACACS+ を設定する例を示します。

```
aaa new-model
aaa authentication ppp test group tacacs+ local
tacacs server server1
address IPv4 10.1.2.3
key goaway
exit

interface gigabitethernet 1/0/1
 switchport mode access
```

前述の設定例の回線は、次のように定義されます。

- **aaa new-model** コマンドは、AAA セキュリティ サービスをイネーブルにします。

- **aaa authentication** コマンドにより、PPP を実行するシリアル インターフェイスに使用する方式リスト「test」を定義します。キーワード **group tacacs+** は、TACACS+ を介して認証を実行することを示します。認証中に TACACS+ から何らかのエラーが返される場合、キーワード **local** は、ネットワーク アクセス サーバ上のローカルデータベースを使用して認証が試行されることを示します。
- **tacacs server** コマンドにより TACACS+ デーモンが識別され、**address ipv4** コマンドにより 10.1.2.3 の IP アドレスが指定されます。**key** コマンドにより、共有暗号キーを「goaway」に定義します。
- **interface** コマンドで回線を選択します。**ppp authentication** コマンドは、テスト方式リストをこの回線に適用します。

次に、PPP 認証のセキュリティ プロトコルとして TACACS+ を設定する例を示します。ただし、「test」方式リストの代わりに、「default」方式リストが使用されます。

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
tacacs server server1
address IPv4 10.1.2.3
key goaway
exit

interface gigabitethernet 1/0/1
 switchport mode access
```

前述の設定例の回線は、次のように定義されます。

- **aaa new-model** コマンドは、AAA セキュリティ サービスをイネーブルにします。
- **aaa authentication** コマンドにより、PPP を実行するシリアル インターフェイスに使用する方式リスト「default」を定義します。キーワード **default** は、デフォルトですべてのインターフェイスに PPP 認証が適用されることを示します。**if-needed** キーワードは、ユーザが ASCII ログイン手順を介してすでに認証済みの場合には PPP 認証が不要なのでスキップできることを示します。認証が必要な場合、キーワード **group tacacs+** は、TACACS+ を介して認証が実行されることを示します。認証中に TACACS+ から何らかのエラーが返される場合、キーワード **local** は、ネットワーク アクセス サーバ上のローカルデータベースを使用して認証が試行されることを示します。
- **tacacs server** コマンドにより TACACS+ デーモンが識別され、**address ipv4** コマンドにより 10.1.2.3 の IP アドレスが指定されます。**key** コマンドにより、共有暗号キーを「goaway」に定義します。
- **interface** コマンドで回線を選択します。**ppp authentication** コマンドは、デフォルト方式リストをこの回線に適用します。

次に、PAP に同じ認証アルゴリズムを作成し、「default」ではなく「MIS-access」の方式リストを呼び出す例を示します。

```
aaa new-model
aaa authentication pap MIS-access if-needed group tacacs+ local
```

```
tacacs server server1
address IPv4 10.1.2.3
key goaway
exit

interface gigabitethernet 1/0/1
 switchport mode access
```

前述の設定例の回線は、次のように定義されます。

- **aaa new-model** コマンドは、AAA セキュリティ サービスをイネーブルにします。
- **aaa authentication** コマンドにより、PPP を実行するシリアル インターフェイスに使用する方式リスト「MIS-access」を定義します。方式リスト「MIS-access」は、PPP 認証がすべてのインターフェイスに適用されることを示します。**if-needed** キーワードは、ユーザが ASCII ログイン手順を介してすでに認証済みの場合には PPP 認証が不要なのでスキップできることを示します。認証が必要な場合、キーワード **group tacacs+** は、TACACS+ を介して認証が実行されることを示します。認証中に TACACS+ から何らかのエラーが返される場合、キーワード **local** は、ネットワーク アクセス サーバ上のローカル データベースを使用して認証が試行されることを示します。
- **tacacs server** コマンドにより TACACS+ デーモンが識別され、**address ipv4** コマンドにより 10.1.2.3 の IP アドレスが指定されます。**key** コマンドにより、共有暗号キーを「goaway」に定義します。
- **interface** コマンドで回線を選択します。**ppp authentication** コマンドは、デフォルト方式リストをこの回線に適用します。

次に、IP アドレスが 10.2.3.4 である TACACS+ デーモンと暗号キー「apple」の設定の例を示します。

```
aaa new-model
aaa authentication login default group tacacs+ local
tacacs server server1
address IPv4 10.2.3.4
key apple
```

前述の設定例の回線は、次のように定義されます。

- **aaa new-model** コマンドは、AAA セキュリティ サービスをイネーブルにします。
- **aaa authentication** コマンドで、デフォルトの方式リストを定義します。すべてのインターフェイスでの着信 ASCII ログイン（デフォルト）では、認証に TACACS+ を使用します。応答する TACACS+ サーバがない場合、ネットワーク アクセス サーバは、認証用のローカル ユーザ名データベースに含まれる情報を使用します。
- **tacacs server** コマンドにより TACACS+ デーモンが識別され、**address ipv4** コマンドにより 10.2.3.4 の IP アドレスが指定されます。**key** コマンドにより、共有暗号キーが「apple」になるように定義します。

TACACS+ に関する追加情報

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*

MIB

MIB	MIB のリンク
	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

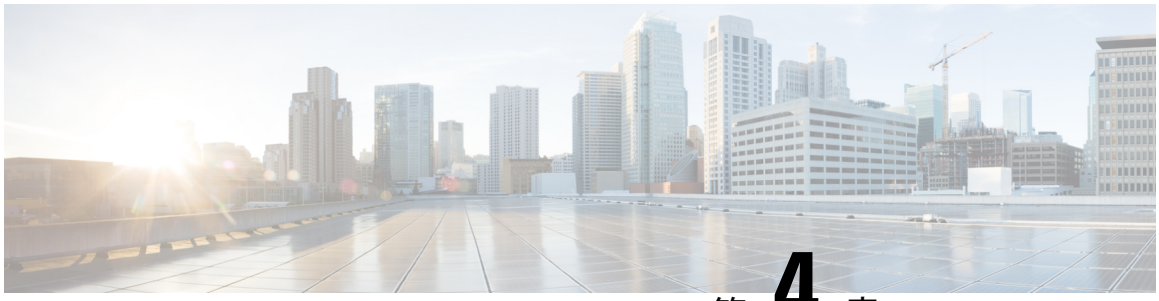
TACACS+ の機能の履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	TACACS+	TACACS+は、認証および認可プロセスについて詳細なアカウンティング情報と柔軟な管理コントロールを提供します。TACACS+ は、AAA を介して実装され、AAA コマンドを使用してのみイネーブルにできます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 4 章

RADIUS の設定

RADIUS セキュリティ システムは、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 クライアントはシスコ デバイス上で実行され、すべてのユーザ認証およびネットワーク サービス アクセス情報を持つ中央の RADIUS サーバに認証要求を送信します。

RADIUS を設定するための前提条件

ここでは、RADIUS によるdevice アクセスの制御の前提条件を示します。

全般：

- この章のいずれかのコンフィギュレーション コマンドを使用するには、RADIUS および認証、許可、ならびにアカウンティング（AAA）をイネーブルにする必要があります。
- RADIUS は、AAA を介して実装され、AAA コマンドを使用してのみイネーブルにできます。
- **aaa new-model** グローバル コンフィギュレーション コマンドを使用して、AAA をイネーブルにします。
- **aaa authentication** グローバル コンフィギュレーション コマンドを使用して、RADIUS 認証の方式リストを定義します。
- **line** および **interface** コマンドを使用して、使用する定義済みの方式リストをイネーブルにします。

- 最低限、RADIUS サーバ ソフトウェアが稼働するホスト（1 つまたは複数）を特定し、RADIUS認証の方式リストを定義する必要があります。また、任意でRADIUS許可およびアカウンティングの方式リストを定義できます。
- device上で RADIUS 機能の設定を行う前に、RADIUS サーバにアクセスし、サーバを設定する必要があります。
- RADIUS ホストは、通常、シスコ（Cisco Secure Access Control Server バージョン 3.0）、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 ソフトウェアが稼働しているマルチユーザ システムです。詳細については、RADIUSサーバのマニュアルを参照してください。
- Change-of-Authorization（CoA）インターフェイスを使用するには、スイッチにセッションがすでに存在している必要があります。CoAを使用すると、セッションの識別と接続解除要求を実行できます。アップデートは、指定されたセッションにだけ作用します。

RADIUS 操作の場合：

- ユーザは RADIUS 許可に進む前に、まず RADIUS 認証を正常に完了する必要があります（イネーブルに設定されている場合）。

RADIUS の設定に関する制約事項

ここでは、RADIUS による device アクセスの制御の制約事項について説明します。

全般：

- セキュリティの失効を防止するため、ネットワーク管理アプリケーションを使用して RADIUS を設定することはできません。

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、AppleTalk Remote Access（ARA）、NetBIOS Frame Control Protocol（NBFCP）、NetWare Asynchronous Services Interface（NASI）、または X.25 PAD 接続をサポートしません。
- スイッチ間またはルータ間状態。RADIUSは、双方向認証を行いません。RADIUSは、他社製のデバイスが認証を必要とする場合に、あるデバイスから他社製のデバイスへ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 に関する情報

RADIUS およびスイッチ アクセス

この項では、RADIUS をイネーブルにし、設定する方法について説明します。RADIUS を使用すると、アカウンティングの詳細を取得したり、認証および許可プロセスの柔軟な管理制御を実現できます。

RADIUS の概要

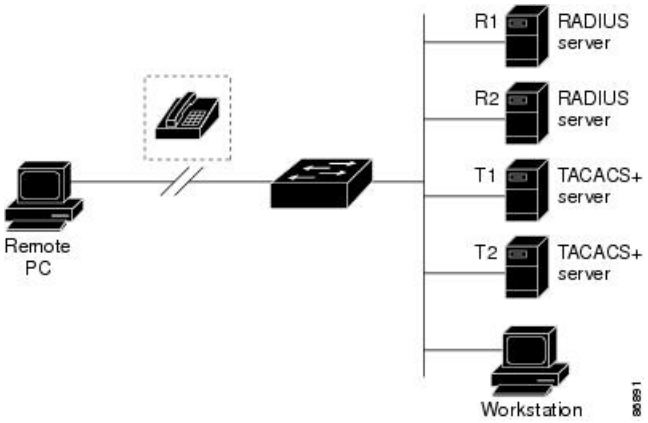
RADIUS は、不正なアクセスからネットワークのセキュリティを保護する分散クライアント/サーバ システムです。RADIUS クライアントは、サポート対象の Cisco ルータおよびスイッチ上で稼働します。クライアントは中央の RADIUS サーバに認証要求を送ります。中央の RADIUS サーバにはすべてのユーザ認証情報、ネットワーク サービス アクセス情報が登録されています。

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します。

- 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。あるケースでは、RADIUS は Enigma のセキュリティ カードとともに使用してユーザを確認し、ネットワーク リソースのアクセスを許可します。
- すでに RADIUS を使用中のネットワーク。RADIUS クライアント装備のシスコ deviceをネットワークに追加できます。これが TACACS+ サーバへの移行の最初のステップとなることもあります。下の図 「RADIUS サービスから TACACS+ サービスへの移行」を参照してください。
- ユーザが 1 つのサービスにしかアクセスできないネットワーク。RADIUS を使用すると、ユーザのアクセスを 1 つのホスト、Telnet などの 1 つのユーティリティ、または IEEE 802.1x などのプロトコルを使用するネットワークに制御できます。このプロトコルの詳細については、「IEEE 802.1x ポートベース認証の設定」の章を参照してください。
- リソース アカウンティングが必要なネットワーク。RADIUS 認証または許可とは別個に RADIUS アカウンティングを使用できます。RADIUS アカウンティング機能によって、サービスの開始および終了時点でデータを送信し、このセッション中に使用されるリソース（時間、パケット、バイトなど）の量を表示できます。インターネット サービス プロバイダーは、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、特殊なセキュリティおよび課金に対するニーズを満たすこともできます。

図 *2 : RADIUS* サービスから *TACACS+* サービスへの移行



RADIUS の動作

RADIUS サーバによってアクセス コントロールされるdeviceに、ユーザがログインおよび認証を試みると、次のイベントが発生します。

1. ユーザ名およびパスワードの入力を要求するプロンプトが表示されます。
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。
   - ACCEPT：ユーザは認証されます。
   - REJECT：ユーザの認証が失敗し、ユーザ名およびパスワードの再入力が要求されるか、またはアクセスが拒否されます。
   - CHALLENGE：ユーザに追加データを要求します。
   - CHALLENGE PASSWORD：ユーザは新しいパスワードを選択するように要求されます。

   ACCEPT または REJECT 応答には、特権 EXEC またはネットワーク許可に使用する追加データがバンドルされています。ACCEPT または REJECT パケットには次の追加データが含まれます。

- Telnet、SSH、rlogin、または特権 EXEC サービス
- 接続パラメータ（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、およびユーザ タイムアウトを含む）

RADIUS のデフォルト設定

RADIUS および AAA は、デフォルトではディセーブルに設定されています。

セキュリティの失効を防止するため、ネットワーク管理アプリケーションを使用して RADIUS を設定することはできません。RADIUS をイネーブルに設定した場合、CLI を通じてスイッチにアクセスするユーザを認証できます。

RADIUS サーバ ホスト

スイッチと RADIUS サーバの通信には、次の要素が関係します。

- 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
- キー文字列
- タイムアウト時間
- 再送信回数

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 UDP ポート番号、または IP アドレスと特定の UDP ポート番号によって特定します。IP アドレスと UDP ポート番号の組み合わせによって、一意の ID が作成され、特定の AAA サービスを提供する RADIUS ホストとして個々のポートを定義できます。この一意の ID を使用することによって、同じ IP アドレスにあるサーバ上の複数の UDP ポートに、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 番目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フェールオーバー バックアップとして動作します。この例では、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、スイッチは「%RADIUS-4-RADIUS_DEAD」メッセージを表示し、その後、同じデバイス上で 2 番めに設定されたホスト エントリでアカウンティング サービスを試みます（RADIUS ホスト エントリは、設定した順序に従って試行されます）。

RADIUS サーバとスイッチは、共有秘密テキスト文字列を使用して、パスワードの暗号化および応答の交換を行います。RADIUS で AAA セキュリティ コマンドを使用するように設定するには、RADIUS サーバデーモンが稼働するホストと、そのホストがスイッチと共有する秘密テキスト（キー）文字列を指定する必要があります。

タイムアウト、再送信回数、および暗号キーの値は、すべての RADIUS サーバに対してグローバルに設定することもできますし、サーバ単位で設定することもできます。また、グローバルな設定とサーバ単位での設定を組み合わせることもできます。

RADIUS ログイン認証

AAA 認証を設定するには、認証方式の名前付きリストを作成してから、各種ポートにそのリストを適用します。方式リストは実行される認証のタイプと実行順序を定義します。このリストを特定のポートに適用してから、定義済み認証方式を実行する必要があります。唯一の例外は、デフォルトの方式リストです。デフォルトの方式リストは、名前付き方式リストを明示的に定義されたインターフェイスを除いて、自動的にすべてのポートに適用されます。

方式リストは、ユーザ認証のためクエリ送信を行う手順と認証方式を記述したものです。認証に使用する 1 つまたは複数のセキュリティ プロトコルを指定できるので、最初の方式が失敗した場合のバックアップ システムが確保されます。ソフトウェアは、リスト内の最初の方式を使用してユーザを認証します。その方式で応答が得られなかった場合、ソフトウェアはそのリストから次の認証方式を選択します。このプロセスは、リスト内の認証方式による通信が成功するか、定義された方式をすべて試し終わるまで繰り返されます。このサイクルのある時点で認証が失敗した場合（つまり、セキュリティ サーバまたはローカルのユーザ名データベースがユーザアクセスを拒否すると応答した場合）、認証プロセスは停止し、それ以上認証方式が試行されることはありません。

AAA サーバグループ

既存のサーバ ホストを認証用にグループ化するため、AAA サーバ グループを使用するようにデバイスを設定できます。設定済みのサーバ ホストのサブセットを選択して、それを特定のサービスに使用します。サーバ グループは、選択されたサーバ ホストの IP アドレスのリストを含むグローバルなサーバ ホスト リストとともに使用されます。

サーバ グループには、同じサーバの複数のホスト エントリを含めることもできますが、各エントリが一意の ID（IP アドレスと UDP ポート番号の組み合わせ）を持っていることが条件です。この場合、個々のポートをそれぞれ特定の AAA サービスを提供する RADIUS ホストとして定義できます。この一意の ID を使用することによって、同じ IP アドレスにあるサーバ上の異なる UDP ポートに、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 エントリのフェールオーバー バックアップとして動作します。最初のホスト エントリがアカウンティング サービスの提供に失敗すると、ネットワーク アクセス サーバは同じデバイスに設定されている 2 番めのホスト エントリを使用してアカウンティング サービスを提供するように試行します。（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）。

AAA 許可

AAA 許可によってユーザが使用できるサービスが制限されます。AAA 許可が有効になっていると、デバイスはユーザのプロファイルから取得した情報を使用します。このプロファイルは、ローカルのユーザデータベースまたはセキュリティサーバ上にあり、ユーザのセッションを設定します。ユーザは、ユーザ プロファイル内の情報で認められている場合に限り、要求したサービスのアクセスが認可されます。

RADIUS アカウンティング

AAA アカウンティング機能は、ユーザが使用したサービスと、消費したネットワーク リソース量を追跡します。AAA アカウンティングをイネーブルにすると、スイッチはユーザの活動状況をアカウンティング レコードの形式で RADIUS セキュリティ サーバに報告します。各アカウンティング レコードにはアカウンティングの Attribute-Value（AV）ペアが含まれ、レコードはセキュリティ サーバに格納されます。これらのデータは、ネットワーク管理、クライアントへの課金、または監査のために後で分析できます。

ベンダー固有の RADIUS 属性

Internet Engineering Task Force（IETF）ドラフト規格に、ベンダー固有の属性（属性 26）を使用して、デバイスと RADIUS サーバ間でベンダー固有の情報を通信するための方式が定められています。各ベンダーは、Vendor-Specific Attribute（VSA）を使用することによって、一般的な用途には適さない独自の拡張属性をサポートできます。シスコが実装する RADIUS では、この仕様で推奨されるフォーマットを使用して、ベンダー固有のオプションを 1 つサポートしています。シスコのベンダー ID は 9 であり、サポート対象のオプションはベンダータイプ 1（名前は *cisco-avpair*）です。この値は、次のフォーマットのストリングです。

```
protocol : attribute sep value *
```

protocol は、特定の認証タイプに使用するシスコのプロトコル属性の値です。*attribute* および *value* は、シスコの TACACS+ 仕様で定義されている適切な属性値（AV）ペアです。*sep* は、必須の属性の場合は =、任意指定の属性の場合は * です。TACACS+ 認証で使用できるすべての機能は、RADIUS でも使用できます。

たとえば、次の AV ペアにより、IP 許可時（PPP のインターネットプロトコル制御プロトコル（IPCP）アドレス割り当て時）に、シスコの複数の名前付き IP アドレスプール機能がアクティブになります。

```
cisco-avpair= ”ip:addr-pool=first“
```

「*」を挿入すると、AV ペア「ip:addr-pool=first」は省略可能になります。任意の AV ペアを省略可能にすることができます。

```
cisco-avpair= ”ip:addr-pool*first“
```

次に、ネットワーク アクセス サーバからユーザがログインしたときに、すぐに EXEC コマンドを実行する方法の例を示します。

```
cisco-avpair= ”shell:priv-lvl=15“
```

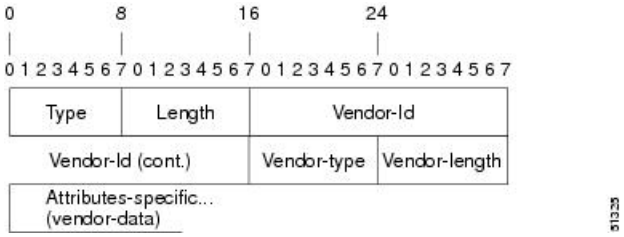
他のベンダーにも、それぞれ独自のベンダー ID、オプション、および対応する VSA があります。ベンダー ID および VSA の詳細については、RFC 2138『Remote Authentication Dial-In User Service (RADIUS)』を参照してください。

属性 26 には、次の 3 つの要素が含まれています。

- タイプ
- 長さ
- ストリング（またはデータ）
  - Vendor-Id
  - Vendor-Type
  - Vendor-Length
  - Vendor-Data

次の図は、属性 26 の「背後で」カプセル化される VSA のパケット形式を示します。

図 *3:* 属性 *26* の背後でカプセル化される *VSA*



（注） VSA の形式はベンダーが指定します。Attribute-Specific フィールド（Vendor-Data とも呼ばれる）は、ベンダーによるその属性の定義によって異なります。

次の表に、「ベンダー固有 RADIUS IETF 属性テーブル」（次の 2 番目の表）で表示される重要なフィールドを示します。これは、サポート対象のベンダー固有 RADIUS 属性（IETF 属性 26）を表示します。

表 *7:* ベンダー固有属性表のフィールドの説明

フィールド	説明
番号	次の表に示されるすべての属性は、IETF 属性 26 の拡張です。
ベンダー固有のコマンドコード	特定のベンダーの識別に使用する定義されたコード。コード 9 は Cisco VSA、311 は Microsoft VSA、529 は Ascend VSA を定義します。
サブタイプ番号	属性 ID 番号。この番号は、属性 26 の背後でカプセル化される「2 番めのレイヤ」の ID 番号であること以外、IETF 属性の ID 番号に似ています。
属性	属性の ASCII ストリング名。
説明	属性の説明。

表 *8* : ベンダー固有 *RADIUS IETF* 属性

番号	ベンダー固有の企業コード	サブタイプ番号	属性	説明
MS-CHAP 属性				
26	311	1	MSCHAP-Response	PPP MS-CHAP ユーザがチャレンジに対する応答で提供するレスポンス値が含まれます。Access-Request パケットでしか使用されません。この属性は、PPP CHAP ID と同じです（RFC 2548）
26	311	11	MSCHAP-Challenge	ネットワーク アクセス サーバが MS-CHAP ユーザに送信するチャレンジが含まれます。これは、Access-Request パケットと Access-Challenge パケットの両方で使用できます。（RFC 2548）
VPDN 属性				
26	9	1	l2tp-cm-local-window-size	L2TP 制御メッセージの最大受信ウィンドウサイズを指定します。この値は、トンネルの確立中にピアにアドバタイズされます。
26	9	1	l2tp-drop-out-of-order	正しくない順序で受信したデータパケットをドロップして、シーケンス番号を順守します。これは受信した場合の処理方法であって、データパケット上でシーケンス番号が送信されるわけではありません。
26	9	1	l2tp-hello-interval	hello キープアライブ インターバルの秒数を指定します。ここで指定した秒数、トンネルでデータが送信されないと、hello パケットが送信されます。

番号	ベンダー固有の企業コード	サブタイプ番号	属性	説明
26	9	1	l2tp-hidden-avp	イネーブルにすると、L2TP 制御メッセージで、大文字小文字を区別する AVP にスクランブルがかけられるか、または非表示になります。
26	9	1	l2tp-nosession-timeout	タイムアウトおよびシャットダウンまでに、セッションなしでトンネルがアクティブのままになる秒数を指定します。
26	9	1	tunnel-tos-reflect	LNS でトンネルに入るパケットに対して、IP ToS フィールドを各ペイロードパケットの IP ヘッダーからトンネル パケットの IP ヘッダーにコピーします。
26	9	1	l2tp-tunnel-authen	この属性を設定すると、L2TP トンネル認証が実行されます。
26	9	1	l2tp-tunnel-password	L2TP トンネル認証および AVP 隠蔽に使用される共有秘密。
26	9	1	l2tp-udp-checksum	これは認可属性で、L2TP がデータ パケットに対して UDP チェックサムを実行する必要があるかどうかを定義します。有効な値は「yes」と「no」です。デフォルトは「no」です。
Store and Forward Fax 属性				
26	9	3	Fax-Account-Id-Origin	**mmoip aaa receive-id** コマンドまたは **mmoip aaa send-id** コマンドについて、システム管理者によって定義されたものとしてアカウント ID の発信元を示します。

番号	ベンダー固有の企業コード	サブタイプ番号	属性	説明
26	9	4	Fax-Msg-Id=	Store and Forward Fax 機能によって割り当てられた一意のファクスメッセージ識別番号を示します。
26	9	5	Fax-Pages	このファクスセッション中に送信または受信したページ数を示します。このページ数には、カバーページも含まれます。
26	9	6	Fax-Coverpage-Flag	カバーページがこのファクスセッションのオフランプゲートウェイで生成されたかどうかを示します。true はカバーページが生成されたことを示します。false はカバーページが生成されなかったことを意味します。
26	9	7	Fax-Modem-Time	モデムがファクスデータを送信した時間（x）、およびファクス セッションの合計時間（y）を秒単位で示します。これには、fax-mail および PSTN 時間が x/y の形式で含まれます。たとえば、10/15 は送信時間が 10 秒で、合計ファクス セッションが 15 秒であったことを示します。
26	9	8	Fax-Connect-Speed	この fax-mail が最初に送信または受信された時点のモデム速度を示します。有効値は、1200、4800、9600、および 14400 です。
26	9	9	Fax-Recipient-Count	このファクス送信の受信者数を示します。E メール サーバがセッション モードをサポートするまで、この数字は 1 にする必要があります。

番号	ベンダー固有の企業コード	サブタイプ番号	属性	説明
26	9	10	Fax-Process-Abort-Flag	ファクスセッションが中断したこと、または正常に終了したことを示します。true はセッションが中断したことを示します。false はセッションが成功したことを示します。
26	9	11	Fax-Dsn-Address	DSN の送信先のアドレスを示します。
26	9	12	Fax-Dsn-Flag	DSN がイネーブルにされているかどうかを示します。true は DSN がイネーブルにされていることを示します。false は DSN がイネーブルにされていないことを示します。
26	9	13	Fax-Mdn-Address	MDN の送信先のアドレスを示します。
26	9	14	Fax-Mdn-Flag	メッセージ配信通知（MDN）がイネーブルにされているかどうかを示します。true は MDN がイネーブルにされていることを示します。false は MDN がイネーブルにされていないことを示します。
26	9	15	Fax-Auth-Status	このファクスセッションに対する認証が成功したかどうかを示します。このフィールドに対する有効値は、success、failed、bypassed、または unknown です。
26	9	16	Email-Server-Address	オンランプ fax-mail メッセージを処理する E メールサーバの IP アドレスを示します。
26	9	17	Email-Server-Ack-Flag	オンランプ ゲートウェイが fax-mail メッセージを受け入れる E メールサーバから肯定確認応答を受信したことを示します。

番号	ベンダー固有の企業コード	サブタイプ番号	属性	説明
26	9	18	Gateway-Id	ファクスセッションを処理したゲートウェイの名前を示します。名前は、hostname.domain-name という形式で表示されます。
26	9	19	Call-Type	ファクスのアクティビティのタイプを、fax receive または fax send のどちらかで記述します。
26	9	20	Port-Used	この fax-mail の送受信いずれかに使用される Cisco AS5300 のスロット/ポート番号を示します。
26	9	21	Abort-Cause	ファクスセッションが中断した場合、中断の信号を送信したシステムコンポーネントを示します。中断する可能性のあるシステムコンポーネントには、FAP（Fax Application Process）、TIFF（TIFF リーダーまたはTIFF ライター）、fax-mail クライアント、fax-mail サーバー、ESMTP クライアント、ESMTP サーバーなどがあります。
H323 属性				
26	9	23	Remote-Gateway-ID（h323-remote-address）	リモート ゲートウェイの IP アドレスを示します。
26	9	24	Connection-ID（h323-conf-id）	会議 ID を識別します。
26	9	25	Setup-Time（h323-setup-time）	以前、グリニッジ標準時（GMT）およびズールタイムと呼ばれていた協定世界時（UTC）でのこの接続のセットアップ時間を示します。

番号	ベンダー固有の企業コード	サブタイプ番号	属性	説明
26	9	26	Call-Origin（h323-call-origin）	ゲートウェイに対するコールの発行元を示します。有効値は、originating および terminating です（回答）。
26	9	27	Call-Type（h323-call-type）	コールのレグ タイプを示します。使用可能な値は **telephony** と **VoIP** です。
26	9	28	Connect-Time（h323-connect-time）	このコール レッグの UTC での接続時間を示します。
26	9	29	Disconnect-Time（h323-disconnect-time）	このコール レッグが UTC で接続解除された時間を示します。
26	9	30	Disconnect-Cause（h323-disconnect-cause）	Q.931 仕様によって、接続がオフラインにされた理由を示します。
26	9	31	Voice-Quality（h323-voice-quality）	コールの音声品質に影響する Impairment Factor（ICPIF）を指定します。
26	9	33	Gateway-ID（h323-gw-id）	下位のゲートウェイの名前を示します。
大規模のダイヤルアウト属性				
26	9	1	callback-dialstring	コールバックに使用するダイヤリング文字列を定義します。
26	9	1	data-service	説明はありません。
26	9	1	dial-number	ダイヤルする番号を定義します。
26	9	1	force-56	チャネルの 64 K すべてが使用可能に見える場合でも、ネットワーク アクセス サーバが 56 K の部分のみを使用するかどうかを指定します。

番号	ベンダー固有の企業コード	サブタイプ番号	属性	説明
26	9	1	map-class	ユーザプロファイルに、ダイヤルアウトするネットワークアクセスサーバ上で同じ名前のマップクラスで設定される情報の参照を許可します。
26	9	1	send-auth	CLID 認証に続く、username-password 認証で使用するプロトコル（PAP または CHAP）を定義します。

番号	ベンダー固有の企業コード	サブタイプ番号	属性	説明
26	9	1	send-name	PPP 名前認証。PAP に適用する場合、インターフェイスで **ppp pap sent-name password** コマンドは設定しないでください。PAP の場合、アウトバウンド認証の PAP ユーザ名および PAP パスワードとして、「preauth:send-name」および「preauth:send-secret」が使用されます。CHAP の場合、「preauth:send-name」は、アウトバウンド認証だけでなく、インバウンド認証にも使用されます。CHAP インバウンドの場合、NAS は発信元のボックスへのチャレンジ パケットに、「preauth:send-name」で定義された名前を使用します。 （注） send-name 属性は時間の経過とともに変わっています。最初は、現在 send-name および remote-name 属性の両方で提供されている機能を実行していました。remote-name 属性が追加されたため、send-name 属性は現在の動作に制限されています。

番号	ベンダー固有の企業コード	サブタイプ番号	属性	説明
26	9	1	send-secret	PPP パスワード認証。ベンダー固有属性（VSA）の場合、アウトバウンド認証の PAP ユーザ名および PAP パスワードとして、「preauth:send-name」および「preauth:send-secret」が使用されます。CHAP アウトバウンドの場合、「preauth:send-name」と「preauth:send-secret」の両方が応答パケットで使用されます。
26	9	1	remote-name	大規模のダイヤルアウトで使用するリモートホストの名前を提供します。ダイヤラは、大規模のダイヤルアウトのリモート名が認証された名前と一致することを確認し、偶発的なユーザ RADIUS 設定ミスから保護します（有効な電話番号にダイヤルしたが誤ったデバイスに接続されるなどのミスです）。
その他の属性				
26	9	2	Cisco-NAS-Port	NAS-Port アカウンティングに追加的なベンダー固有属性（VSA）を指定します。追加的な NAS-Port 情報を属性値ペア（AVPair）の形式で指定するには、**radius-server vsa send** 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。 （注） この VSA は、通常アカウンティングで使用されますが認証（Access-Request）パケットで使用される場合もあります。

番号	ベンダー固有の企業コード	サブタイプ番号	属性	説明
26	9	1	min-links	MLP に対するリンクの最小数を設定します。
26	9	1	proxyacl#<n>	ダウンロード可能なユーザプロファイル（ダイナミック ACL）を、認証プロキシを使用して設定でき、これにより設定されたインターフェイスのトラフィックの通過を許可するよう、認証を設定できます。
26	9	1	spi	登録中にホームエージェントがモバイルノードの認証で必要とする認証情報を伝送します。この情報は、**ip mobile secure host <addr>** コンフィギュレーションコマンドと同じ構文です。基本的に、この文字列に続く残りのコンフィギュレーションコマンドはそのまま含まれます。これにはセキュリティパラメータインデックス（SPI）、キー、認証アルゴリズム、認証モード、およびリプレイ保護タイムスタンプ範囲が含まれています。

RADIUS Disconnect-Cause 属性値

Disconnect-cause 属性値は、接続がオフラインにされた理由を指定します。属性値は、アカウンティング要求パケットで送信されます。セッションの認証が失敗しても、これらの値は、セッションの終了時に送信されます。セッションが認証されないと、属性が開始レコードを生成せずに終了レコードを発生させる可能性があります。

次の表に、Disconnect-Cause（195）属性の原因コード、値、および説明を示します。

（注） Disconnect-Cause は、RADIUS AVPairs で使用されるごとに 1000 ずつ増分されます。たとえば、disc-cause 4 は 1004 になります。

表 *9 : Disconnect-Cause* 属性値

原因コード	値	説明
0	No-Reason	接続解除の理由は提供されない。
1	No-Disconnect	イベントは接続解除されていない。
2	Unknown	理由は不明。
3	Call-Disconnect	コールが接続解除された。
4	CLID-Authentication-Failure	calling-party 数の認証の失敗。
9	No-Modem-Available	コールへの接続にモデムが使用できない。
10	No-Carrier	キャリアが検出されない。 （注） 最初のモデム接続中に接続解除があると、コード 10、11、および 12 が送信される場合があります。
11	Lost-Carrier	キャリアの喪失。
12	No-Detected-Result-Codes	モデム結果コード検出の失敗。
20	User-Ends-Session	ユーザがセッションを終了した。 （注） コード 20、22、23、24、25、26、27、および 28 は、EXEC セッションに適用されます。
21	Idle-Timeout	ユーザ入力待機中のタイムアウト。 コード 21、100、101、102、および 120 は、すべてのセッション タイプに適用されます。
22	Exit-Telnet-Session	既存の Telnet セッションによる接続解除。
23	No-Remote-IP-Addr	SLIP/PPP への切り替え不能。リモート エンドに IP アドレスがない。
24	Exit-Raw-TCP	既存の raw TCP による接続解除。
25	Password-Fail	間違ったパスワード。
26	Raw-TCP-Disabled	Raw TCP がディセーブルにされた。
27	Control-C-Detected	Control-C が検出された。
28	EXEC-Process-Destroyed	EXEC プロセスが破棄された。
29	Close-Virtual-Connection	ユーザが仮想接続を終了した。
30	End-Virtual-Connection	仮想接続が終了した。

原因コード	値	説明
31	Exit-Rlogin	ユーザが Rlogin を終了した。
32	Invalid-Rlogin-Option	無効な Rlogin オプションが選択された。
33	Insufficient-Resources	不十分なリソース。
40	Timeout-PPP-LCP	PPP LCP ネゴシエーションがタイムアウトした。 （注） コード 40 ～ 49 が PPP セッションに適用されます。
41	Failed-PPP-LCP-Negotiation	PPP LCP ネゴシエーションが失敗した。
42	Failed-PPP-PAP-Auth-Fail	PPP PAP 認証が失敗した。
43	Failed-PPP-CHAP-Auth	PPP CHAP 認証が失敗した。
44	Failed-PPP-Remote-Auth	PPP リモート認証が失敗した。
45	PPP-Remote-Terminate	PPP がリモート エンドから Terminate Request を受信した。
46	PPP-Closed-Event	上位層がセッションの終了を要求した。
47	NCP-Closed-PPP	開いている NCP がなかったため、PPP セッションが終了した。
48	MP-Error-PPP	MP エラーのため、PPP セッションが終了した。
49	PPP-Maximum-Channels	最大チャネルに達したため、PPP セッションが終了した。
50	Tables-Full	ターミナル サーバ テーブルがいっぱいになったため、接続解除された。
51	Resources-Full	内部リソースがいっぱいになったため、接続解除された。
52	Invalid-IP-Address	Telnet ホストに対する IP アドレスが有効でない。
53	Bad-Hostname	ホスト名が検証されていない。
54	Bad-Port	ポート番号が無効または欠落している。
60	Reset-TCP	TCP 接続がリセットされた。 （注） コード 60 ～ 67 は Telnet または raw TCP セッションに適用されます。
61	TCP-Connection-Refused	TCP 接続がホストによって拒否された。
62	Timeout-TCP	TCP 接続がタイムアウトした。
63	Foreign-Host-Close-TCP	TCP 接続が終了した。
64	TCP-Network-Unreachable	TCP ネットワークに到達できない。

原因コード	値	説明
65	TCP-Host-Unreachable	TCP ホストに到達できない。
66	TCP-Network-Admin Unreachable	管理上の理由により、TCP ネットワークに到達できない。
67	TCP-Port-Unreachable	TCP ポートに到達できない。
100	Session-Timeout	セッションがタイムアウトした。
101	Session-Failed-Security	セキュリティ上の理由から、セッションが失敗した。
102	Session-End-Callback	コールバックにより、セッションが終了した。
120	Invalid-Protocol	検出されたプロトコルがディセーブルにされていたため、コールが拒否された。
150	RADIUS-Disconnect	RADIUS 要求による接続解除。
151	Local-Admin-Disconnect	管理上の接続解除。
152	SNMP-Disconnect	SNMP 要求による接続解除。
160	V110-Retries	許可された V.110 リトライを超過した。
170	PPP-Authentication-Timeout	PPP 認証がタイムアウトした。
180	Local-Hangup	ローカルのハングアップによって接続解除された。
185	Remote-Hangup	リモート エンドのハングアップよって接続解除された。
190	T1-Quiesced	T1 回線が休止状態のため接続解除された。
195	Call-Duration	コールの最大継続時間を超過したため、接続解除された。
600	VPN-User-Disconnect	クライアントによってコールが接続解除された（PPP 経由）。 LNS がクライアントから PPP terminate request を受信するとコードが送信されます。
601	VPN-Carrier-Loss	キャリアの喪失。これは回線が物理的に普通になった結果である場合があります。 クライアントがダイヤラを使用してダイヤルアウトできない場合、コードが送信されます。
602	VPN-No-Resources	コールの処理に使用できるリソースがない。 クライアントがメモリを割り当てることができない場合、コードが送信されます（メモリの不足）。

原因コード	値	説明
603	VPN-Bad-Control-Packet	L2TP または L2F 制御パケットが間違っている。 このコードは、必須の属性値ペア（AVP）が欠落しているなど、ピアから受信した制御パケットが無効な場合に送信されます。L2TP を使用すると、コードは 6 回の再送信後に送信されます。L2F を使用すると、再送信の回数はユーザ設定が可能です。 （注） トンネルにアクティブなセッションがある場合は、VPN-Tunnel-Shut が送信されます。
604	VPN-Admin-Disconnect	管理上の接続解除。これは、VPN ソフト シャットダウンの結果である場合があります。これは、クライアントが最大セッション制限に達するか、最大ホップカウントを超過した場合に発生します。 トンネルが、**clear vpdn tunnel** コマンドの発行によってダウンした場合に、コードが送信されます。
605	VPN-Tunnel-Shut	トンネルのティアダウン、またはトンネルのセットアップが失敗した。 トンネルにアクティブなセッションがあり、トンネルがダウンした場合にコードが送信されます。 （注） このコードはトンネルの認証が失敗した場合は、送信されません。
606	VPN-Local-Disconnect	LNS PPP モジュールによって、コールが接続解除された。 LNS がクライアントに PPP terminate request を送信するとコードが送信されます。これは通常の PPP 接続解除が LNS によって開始されたことを示します。
607	VPN-Session-Limit	VPN ソフト シャットダウンがイネーブルになった。 前述したソフト シャットダウンの制約事項のいずれかによってコールが拒否されると、コードが送信されます。
608	VPN-Call-Redirect	VPN コール リダイレクトがイネーブルになった。

RADIUS 進捗コード

RADIUS 進捗コード機能は、進捗コードを通してコールが切断される前の接続状態を示す RADIUS 属性 196（Ascend-Connect-Progress）に新たな進捗コードを追加します。

属性 196 は、ネットワーク、EXEC、およびリソースアカウンティング開始および終了レコード内で送信されます。各進捗コードからコールの接続状態に関連するアカウンティング情報が特定されるため、この属性によってコール失敗のデバッグが容易になります。この属性はデフォルトでアクティブになります。アカウンティング開始または終了アカウンティングレコー

ドが要求されると、認証、許可、およびアカウンティング（AAA）が、属性196を標準属性リストの一部としてレコードに追加します。アカウンティング開始および終了レコード内で送信される進捗コードがコール失敗のデバッグを容易にするため、属性 196 は有用です。

（注） アカウンティング開始レコードでは、属性 196 に値はありません。

表 *10 :* 属性 *196* で新たにサポートされた進捗コード

コード	説明
10	モデム割り当てとネゴシエーションが完了しています。コールが作動しています。
30	モデムが動作中です。
33	モデムが結果コードを待機しています。
41	最大 TNT が、TCP クリア コールを設定することにより TCP 接続を確立しています。
60	リンク制御プロトコル（LCP）が、PPP および IP Control Protocol（IPCP）ネゴシエーションを伴ってオープンな状態にあります。LAN セッションが動作中です。
65	PPP ネゴシエーションが行われ、初めに、LCP ネゴシエーションが行われます。LCP がオープン状態にあります。
67	オープン状態の LCP を伴う PPP ネゴシエーションが行われた後、IPCP ネゴシエーションが開始されます。

（注） 進捗ステータス コード33、30、および67は、NAS でのデバッグを通して生成され表示されます。他のコードはすべて、RADIUS サーバでのデバッグとアカウンティング レコードを通して生成され表示されます。

ベンダー独自仕様の RADIUS サーバ通信

RADIUS に関する IETF ドラフト規格では、スイッチと RADIUS サーバ間でベンダー独自仕様の情報を通信する方式について定められていますが、RADIUS 属性セットを独自に機能拡張しているベンダーもあります。Cisco IOS ソフトウェアは、ベンダー独自仕様の RADIUS 属性のサブセットをサポートしています。

前述したように、RADIUS（ベンダーの独自仕様によるものか、IETF ドラフトに準拠するものかを問わず）を設定するには、RADIUS サーバ デーモンが稼働しているホストと、そのホストがスイッチと共有秘密テキスト文字列を指定する必要があります。RADIUS ホストおよび秘密テキスト文字列を指定するには、**radius server** 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。

拡張テスト コマンド

拡張テスト コマンド機能を使用すると、発呼回線 ID（CLID）または着信番号識別サービス（DNIS）属性値を持つ名前付きユーザ プロファイルを作成できます。RADIUS サーバがすべての着信コールの CLID または DNIS 属性情報にアクセスできるように、CLID または DNIS 属性値を、ユーザ プロファイルとともに送信される RADIUS レコードに関連付けることができます。

RADIUS の設定方法

RADIUS サーバ ホストの識別

デバイスと通信するすべてのRADIUSサーバにこのような設定をグローバルに適用するには、**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。

既存のサーバ ホストを認証用にグループ化するため、AAA サーバ グループを使用するようにデバイスを設定できます。詳細については、次の関連項目を参照してください。

RADIUS サーバ上でも、いくつかの値を設定する必要があります。deviceの IP アドレス、およびサーバとdeviceの双方で共有するキーストリングなどの設定値です。詳細については、RADIUS サーバのマニュアルを参照してください。

サーバ単位で RADIUS サーバとの通信を設定するには、次の手順を実行します。

始める前に

device上にグローバルな機能とサーバ単位での機能（タイムアウト、再送信回数、およびキー コマンド）を設定した場合、サーバ単位で設定したタイムアウト、再送信回数、およびキーに関するコマンドは、グローバルに設定したタイムアウト、再送信回数、およびキーに関するコマンドを上書きします。すべてのRADIUSサーバに対してこれらの値を設定する方法については、次の関連項目を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例：	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`Device# configure terminal`	
ステップ 3	**radius server** *server name* 例： `Device(config)# radius server server1`	RADIUS サーバーの名前を指定し、RADIUS サーバー コンフィギュレーション モードを開始します。
ステップ 4	**address** {**ipv4** | **ipv6**} *ip address* { **auth-port** *port number* | **acct-port** *port number*} 例： `Device(config-radius-server)# address ipv4 172.2.2.12 auth-port 1612`	RADIUS サーバのパラメータを指定します。 **auth-port** *port-number* には、認証要求の UDP 宛先ポートを指定します。デフォルトは 1645 です。指定できる範囲は 0 ～ 65536 です。 **acct-port** *port-number* には、認証要求の UDP 宛先ポートを指定します。デフォルトは 1646 です。
ステップ 5	**key** [**0** | **6** | **7**] *string* 例： `Device(config-radius-server)# key 0`	device と RADIUS サーバ上で動作する RADIUS デーモンとの間で使用される認証および暗号キーを指定します。

	コマンドまたはアクション	目的
	`cisco123`	（注） • キー 0、6、7 は、それぞれクリアテキストパスワード、タイプ 6 暗号化、タイプ 7 暗号化を示します。キーがタイプ 7 と設定されている場合は、タイプ 7 の有効な暗号化文字列も設定する必要があります。同様に、キータイプ 6 の後に続けてタイプ 6 の暗号化文字列を指定する必要があります。 • テキスト文字列は、RADIUS サーバーで使用する暗号化キーと一致させる必要があります。必ず **radius server** コマンドの最終項目としてキーを設定してください。先頭のスペースは無視されますが、キーの中間および末尾のスペースは使用されます。キーにスペースを使用する場合は、引用符がキーの一部分である場合を除き、引用符でキーを囲まないでください。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

すべての RADIUS サーバの設定

すべての RADIUS サーバを設定するには、特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

ステップ 3	**radius-server key** *string*
	例：
	`Device(config)# radius-server key your_server_key`
	`Device(config)# key your_server_key`
		スイッチとすべての RADIUS サーバ間で共有されるシークレット テキスト ストリングを指定します。
		（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。先頭のスペースは無視されますが、キーの中間および末尾のスペースは使用されます。キーにスペースを使用する場合は、引用符がキーの一部分である場合を除き、引用符でキーを囲まないでください。

ステップ 4	**radius-server retransmit** *retries* 例： `Device(config)# radius-server retransmit 5`	スイッチが RADIUS 要求をサーバに再送信する回数を指定します。デフォルトは 3 です。指定できる範囲は 1 ～ 1000 です。
ステップ 5	**radius-server timeout** *seconds* 例： `Device(config)# radius-server timeout 3`	スイッチが RADIUS 要求に対する応答を待って、要求を再送信するまでの時間（秒）を指定します。デフォルトは5秒です。指定できる範囲は 1 ～ 1000 です。
ステップ 6	**radius-server deadtime** *minutes* 例：	RADIUS サーバが認証要求に応答していない場合、このコマンドはそのサーバに対する要求を停止する時刻を指定しま

	コマンドまたはアクション	目的
	`Device(config)# radius-server deadtime 0`	す。これにより、要求がタイムアウトするまで待たずとも、次に設定されているサーバを試行することができます。デフォルトは 0 です。指定できる範囲は 1 ～ 1440 分です。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

RADIUS ログイン認証の設定

RADIUS ログイン認証を設定するには、次の手順を実行します。

始める前に

AAA 方式を使用して HTTP アクセスに対しデバイスのセキュリティを確保するには、**ip http authentication aaa** グローバル コンフィギュレーション コマンドでデバイスを設定する必要があります。AAA 認証を設定しても、AAA 方式を使用した HTTP アクセスに対しデバイスのセキュリティは確保されません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 4	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] 例： `Device(config)# aaa authentication login default local`	ログイン認証方式リストを作成します。 • **login authentication** コマンドにリストが指定されていない場合に使用するデフォルトのリストを作成するには、**default** キーワードの後ろにデフォルト状況で使用される方式を指定します。デフォルトの方式リストは、自動的にすべてのポートに適用されます。 • *list-name* には、作成するリストの名前として使用する文字列を指定します。 • *method1...* には、認証アルゴリズムが試行する実際の方式を指定します。追加の認証方式は、その前の方式でエラーが返された場合に限り使用されます。前の方式が失敗した場合は使用されません。 次のいずれかの方式を選択します。 • *enable*：イネーブルパスワードを認証に使用します。この認証方式を使用するには、あらかじめ **enable** *password* グローバル コンフィギュレーション コマンドを使用してイネーブルパスワードを定義しておく必要があります。 • *group radius*：RADIUS 認証を使用します。この認証方式を使用するには、あらかじめ RADIUS サーバを設定しておく必要があります。 • *line*：回線パスワードを認証に使用します。この認証方式を使用するには、あらかじめ回線パスワードを定義しておく必要があります。**password** *password* ライン コンフィギュレーション コマンドを使用します。

	コマンドまたはアクション	目的
		• *local*：ローカルユーザ名データベースを認証に使用します。データベースにユーザ名情報を入力しておく必要があります。**username** *name* **password** 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。 • *local-case*：大文字と小文字が区別されるローカルユーザ名データベースを認証に使用します。**username** *password* グローバル コンフィギュレーション コマンドを使用して、ユーザ名情報をデータベースに入力する必要があります。 • *none*：ログインに認証を使用しません。
ステップ 5	**line** [**console** | **tty** | **vty**] *line-number* [*ending-line-number*] 例： `Device(config)# line 1 4`	ライン コンフィギュレーション モードを開始し、認証リストを適用する回線を設定します。
ステップ 6	**login authentication** {**default** | *list-name*} 例： `Device(config)# login authentication default`	1 つの回線または複数回線に認証リストを適用します。 • **default** を指定する場合は、**aaa authentication login** コマンドで作成したデフォルトのリストを使用します。 • *list-name* には、**aaa authentication login** コマンドで作成したリストを指定します。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

AAA サーバ グループの定義

定義したグループサーバに特定のサーバを対応付けるには、**server** グループ サーバ コンフィギュレーション コマンドを使用します。サーバを IP アドレスで特定することもできますし、任意指定の **auth-port** および **acct-port** キーワードを使用して複数のホストインスタンスまたはエントリを特定することもできます。

AAA サーバ グループを定義するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** **例：** `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**radius server** *name* **例：** `Device(config)# radius server ISE`	RADIUS サーバの設定の名前を Protected Access Credential（PAC）のプロビジョニング用に指定し、RADIUS サーバ設定モードを開始します。 deviceは、IPv6 対応の RADIUS をサポートしています。
ステップ 4	**address** {**ipv4** | **ipv6**} {*ip-address* | *hostname*} **auth-port** *port-number* **acct-port** *port-number* **例：** `Device(config-radius-server)# address ipv4 10.1.1.1 auth-port 1645 acct-port 1646`	RADIUS サーバのアカウンティングおよび認証パラメータの IPv4 アドレスを設定します。
ステップ 5	**key** [**0** | **6** | **7**] *string* **例：** `Device(config-radius-server)# key 0`	デバイスと RADIUS サーバとの間におけるすべての RADIUS 通信用の認証および暗号キーを指定します。

	コマンドまたはアクション	目的
	cisco123	（注） • キー 0、6、7 は、それぞれクリアテキストパスワード、タイプ 6 暗号化、タイプ 7 暗号化を示します。キーがタイプ 7 と設定されている場合は、タイプ 7 の有効な暗号化文字列も設定する必要があります。同様に、キータイプ 6 の後に続けてタイプ 6 の暗号化文字列を指定する必要があります。 • テキスト文字列は、RADIUS サーバーで使用する暗号化キーと一致させる必要があります。必ず **radius server** コマンドの最終項目としてキーを設定してください。先頭のスペースは無視されますが、キーの中間および末尾のスペースは使用されます。キーにスペースを使用する場合は、引用符がキーの一部分である場合を除き、引用符でキーを囲まないでください。
ステップ 6	**end** 例： `Device(config-radius-server)# end`	RADIUS サーバ コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

ユーザ イネーブル アクセスおよびネットワーク サービスに関する RADIUS 許可の設定

（注） 許可が設定されていても、CLI を使用してログインし、認証されたユーザに対しては、許可は省略されます。

ユーザ特権アクセスおよびネットワーク サービスに関する RADIUS 許可を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa authorization network radius** 例： `Device(config)# aaa authorization network radius`	ネットワーク関連のすべてのサービス要求に対して、ユーザが RADIUS 許可を受けるようにデバイスを設定します。
ステップ 4	**aaa authorization exec radius** 例： `Device(config)# aaa authorization exec radius`	ユーザに特権 EXEC のアクセス権限がある場合、ユーザが RADIUS 許可を受けるように device を設定します。 **exec** キーワードを指定すると、ユーザ プロファイル情報（**autocommand** 情報など）が返される場合があります。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

次のタスク

aaa authorization グローバル コンフィギュレーション コマンドと **radius** キーワードを使用すると、ユーザのネットワーク アクセスを特権 EXEC モードに制限するパラメータを設定できます。

aaa authorization exec radius local コマンドは、次の許可パラメータを設定します。

- RADIUS を使用して認証を行った場合は、RADIUS を使用して特権 EXEC アクセスを許可します。
- 認証に RADIUS を使用しなかった場合は、ローカル データベースを使用します。

RADIUS アカウンティングの起動

RADIUS アカウンティングを開始するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa accounting network start-stop radius** 例： `Device(config)# aaa accounting network start-stop radius`	ネットワーク関連のあらゆるサービス要求に関して、RADIUS アカウンティングをイネーブルにします。
ステップ 4	**aaa accounting exec start-stop radius** 例： `Device(config)# aaa accounting exec start-stop radius`	RADIUS アカウンティングをイネーブルにして、特権 EXEC プロセスの最初に記録開始アカウンティング通知、最後に記録停止通知を送信します。
ステップ 5	**end** 例：	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config)# end`	

属性 196 の確認

RADIUS進捗コードには設定は必要ありません。アカウンティング開始および停止レコード内の属性 196 を確認するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**debug aaa accounting** 例： `Device# debug aaa accounting`	説明の義務があるイベントが発生したときに、その情報を表示します。
ステップ 3	**show radius statistics** 例： `Device# debug aaa authorization`	アカウンティングパケットと認証パケットについての RADIUS 統計情報を示します。

ベンダー固有の RADIUS 属性を使用するデバイスの設定

ベンダー固有のRADIUS属性を使用するようにデバイスを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例：	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`Device# configure terminal`	
ステップ 3	**radius-server vsa send** [**accounting** | **authentication**] 例： `Device(config)# radius-server vsa send accounting`	device が VSA（RADIUS IETF 属性 26 で定義）を認識して使用できるようにします。 • （任意）認識されるベンダー固有属性の集合をアカウンティング属性だけに限定するには、**accounting** キーワードを使用します。 • （任意）認識されるベンダー固有属性の集合を認証属性だけに限定するには、**authentication** キーワードを使用します。 キーワードを指定せずにこのコマンドを入力すると、アカウンティングおよび認証のベンダー固有属性の両方が使用されます。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

ベンダー独自仕様の RADIUS サーバ通信に関するデバイスの設定

ベンダー独自仕様の RADIUS サーバ通信を使用するようにデバイスを設定するには、次の手順に従います。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例：	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`Device# configure terminal`	
ステップ 3	**radius server** *server name* 例： `Device(config)# radius server server1`	RADIUS サーバ設定の名前を Protected Access Credential（PAC）のプロビジョニング用に指定し、RADIUS サーバ設定モードを開始します。
ステップ 4	**address** {**ipv4** | **ipv6**} *ip address* 例： `Device(config-radius-server)# address ipv4 172.2.2.12`	（任意）RADIUS サーバの IP アドレスを指定します。
ステップ 5	**non-standard** 例： `Device(config-radius-server)# non-standard`	RADIUS サーバが RADIUS ベンダー独自の実装を使用していることを示します。
ステップ 6	**key** [**0** | **6** | **7**] *string* 例： `Device(config-radius-server)# key 0 cisco123`	デバイスとベンダー独自仕様の RADIUS サーバーとの間で使用される共有秘密タイプと文字列を指定します。デバイスと RADIUS サーバーはこれを使用してパスワードを暗号化し、応答を交換します。

	コマンドまたはアクション	目的
		（注） • キー 0、6、7 は、それぞれクリアテキストパスワード、タイプ 6 暗号化、タイプ 7 暗号化を示します。キーがタイプ 7 と設定されている場合は、タイプ 7 の有効な暗号化文字列も設定する必要があります。同様に、キータイプ 6 の後に続けてタイプ 6 の暗号化文字列を指定する必要があります。 • テキスト文字列は、RADIUS サーバーで使用する暗号化キーと一致させる必要があります。必ず **radius server** コマンドの最終項目としてキーを設定してください。先頭のスペースは無視されますが、キーの中間および末尾のスペースは使用されます。キーにスペースを使用する場合は、引用符がキーの一部分である場合を除き、引用符でキーを囲まないでください。
ステップ 7	**end** 例： `Device(config-radius-server)# end`	特権 EXEC モードに戻ります。

ユーザ プロファイルの設定と RADIUS レコードへの関連付け

ここでは、CLID または DNIS 属性値を持つ名前付きユーザ プロファイルを作成し、RADIUS レコードに関連付ける方法について説明します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa user profile** *profile-name* 例： `Device(config)# aaa user profile profilename1`	ユーザ プロファイルを作成します。
ステップ 4	**aaa attribute** {**dnis** | **clid**} 例： `Device(config)# aaa attribute dnis`	DNIS または CLID 属性値をユーザ プロファイルに追加し、AAA ユーザ 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	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 6	**test aaa group** {*group-name* | **radius**} *username password* **new-code** [**profile** *profile-name*] 例： `Device# test aaa group radius secret new-code profile profilename1`	DNIS または CLID の名前付きユーザ プロファイルを、RADIUS サーバに送信するレコードに関連付けます。 （注） *profile-name* は **aaa user profile** コマンドで指定するプロファイル名と一致する必要があります。

拡張テスト コマンドの設定の確認

拡張テスト コマンドの設定を確認するには、特権 EXEC モードで次のコマンドを使用します。

コマンド	目的
`Device# debug radius`	RADIUS 関連の情報を表示します。

コマンド	目的
`Device# more system:running-config`	現在実行されているコンフィギュレーション ファイルの内容を表示します(コマンド**more system:running-config**が **show running-config** コマンドに置き換えられていることに注意してください)。

RADIUS の設定例

例：RADIUS サーバホストの識別

次に、1 つの RADIUS サーバを認証用に、もう 1 つの RADIUS サーバをアカウンティング用に設定する例を示します。

```
Device# configure terminal
Device(config)# radius server server1
Device(config-radius-server)# address ipv4 172.2.2.12 auth-port 1612
Device(config-radius-server)# key key1
Device(config-radius-server)# exit
Device(config)# radius server server2
Device(config-radius-server)# address ipv4 172.2.2.20 auth-port 1618
Device(config-radius-server)# key key2
Device(config-radius-server)# exit
```

例：AAA サーバグループ

次に、3 つの RADIUS サーバ メンバを持ち、各メンバがデフォルトの認証ポート（1645）とアカウンティング ポート（1646）を使用するサーバ グループ radgroup1 を作成する例を示します。

```
aaa group server radius radgroup1
 server 172.16.1.11
 server 172.17.1.21
 server 172.18.1.31
```

次に、3 つの RADIUS サーバ メンバを持ち、各メンバが IP アドレスは同じでも認証ポートとアカウンティング ポートはそれぞれ異なるサーバ グループ radgroup2 を作成する例を示します。

```
aaa group server radius radgroup2
 server 172.16.1.1 auth-port 1000 acct-port 1001
 server 172.16.1.1 auth-port 2000 acct-port 2001
 server 172.16.1.1 auth-port 3000 acct-port 3001
```

RADIUS 進捗コードに関するトラブルシューティングのヒント

次の例は、**debug ppp negotiation** コマンドからのデバッグ出力のサンプルです。このデバッグ出力を使用して、アカウンティング終了レコードが生成されていることと、属性 196（Ascend-Connect-Progress）に 65 の値が設定されていることを確認します。

```
Tue Aug 7 06:21:03 2001
        NAS-IP-Address = 10.0.58.62
        NAS-Port = 20018
        Vendor-Specific = ""
        NAS-Port-Type = ISDN
        User-Name = "peer_16a"
        Called-Station-Id = "5213124"
        Calling-Station-Id = "5212175"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed-User
        Acct-Session-Id = "00000014"
        Framed-Protocol = PPP
        Framed-IP-Address = 172.16.0.2
        Acct-Input-Octets = 3180
        Acct-Output-Octets = 3186
        Acct-Input-Packets = 40
        Acct-Output-Packets = 40
        Ascend-Connect-Pr = 65
        Acct-Session-Time = 49
        Acct-Delay-Time = 0
        Timestamp = 997190463
        Request-Authenticator = Unverified
```

例：ベンダー固有の RADIUS 属性を使用するデバイスの設定

たとえば、次の AV ペアを指定すると、IP 許可時（PPP の IPCP アドレスの割り当て時）に、シスコの複数の名前付き IP アドレス プール機能が有効になります。

```
cisco-avpair= "ip:addr-pool=first"
```

次に、デバイスから特権EXECコマンドへの即時アクセスが可能となるユーザログインを提供する例を示します。

```
cisco-avpair= "shell:priv-lvl=15"
```

次に、RADIUS サーバ データベース内の許可 VLAN を指定する例を示します。

```
cisco-avpair= "tunnel-type(#64)=VLAN(13)"
cisco-avpair= "tunnel-medium-type(#65)=802 media(6)"
cisco-avpair= "tunnel-private-group-id(#81)=vlanid"
```

次に、この接続中に ASCII 形式の入力 ACL をインターフェイスに適用する例を示します。

```
cisco-avpair= "ip:inacl#1=deny ip 10.10.10.10 0.0.255.255 20.20.20.20 255.255.0.0"
cisco-avpair= "ip:inacl#2=deny ip 10.10.10.10 0.0.255.255 any"
```

```
cisco-avpair= "mac:inacl#3=deny any any decnet-iv"
```

次に、この接続中に ASCII 形式の出力 ACL をインターフェイスに適用する例を示します。

```
cisco-avpair= "ip:outacl#2=deny ip 10.10.10.10 0.0.255.255 any"
```

例：ベンダー独自仕様の RADIUS サーバ通信に関するデバイスの設定

次に、ベンダー独自の RADIUS ホストを指定し、デバイスとサーバの間で *rad124* という秘密キーを使用する例を示します。

```
Device# configure terminal
Device(config)# radius server server1
Device(config-radius-server)# address ipv4 172.2.2.12
Device(config-radius-server)# nonstandard
Device(config-radius-server)# key rad124
Device(config-radius-server)# exit
```

例：test aaa group コマンドに関連付けるユーザ プロファイル

次に、dnis = dnisvalue ユーザプロファイル *prfl1* を設定し、**test aaa group** コマンドに関連付ける例を示します。この例では、**debug radius** コマンドが有効化され、設定の後に出力が続いています。

```
aaa user profile prfl1
  aaa attribute dnis
  aaa attribute dnis dnisvalue
  no aaa attribute clid
! Attribute not found.
  aaa attribute clid clidvalue
  no aaa attribute clid
  exit
!
! Associate the dnis user profile with the test aaa group command.
test aaa group radius user1 pass new-code profile prfl1
!
!
!
! debug radius output, which shows that the dnis value has been passed to the radius !
server.
*Dec 31 16:35:48: RADIUS: Sending packet for Unique id = 0
 *Dec 31 16:35:48: RADIUS: Initial Transmit unknown id 8 172.22.71.21:1645, Access-Request,
 len 68
 *Dec 31 16:35:48: RADIUS: code=Access-Request id=08 len=0068
         authenticator=1E CA 13 F2 E2 81 57 4C - 02 EA AF 9D 30 D9 97 90
         T=User-Password[2]                L=12 V=*
         T=User-Name[1]                    L=07 V="test"
         T=Called-Station-Id[30]           L=0B V="dnisvalue"
         T=Service-Type[6]                 L=06 V=Login                    [1]
         T=NAS-IP-Address[4]               L=06 V=10.0.1.81

 *Dec 31 16:35:48: RADIUS: Received from id 8 172.22.71.21:1645, Access-Accept, len 38
 *Dec 31 16:35:48: RADIUS: code=Access-Accept id=08 len=0038
```

RADIUS に関する追加情報

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*

標準および RFC

標準/RFC	タイトル
RFC 5176	RADIUS 認可変更（CoA）の拡張

MIB

MIB	MIB のリンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

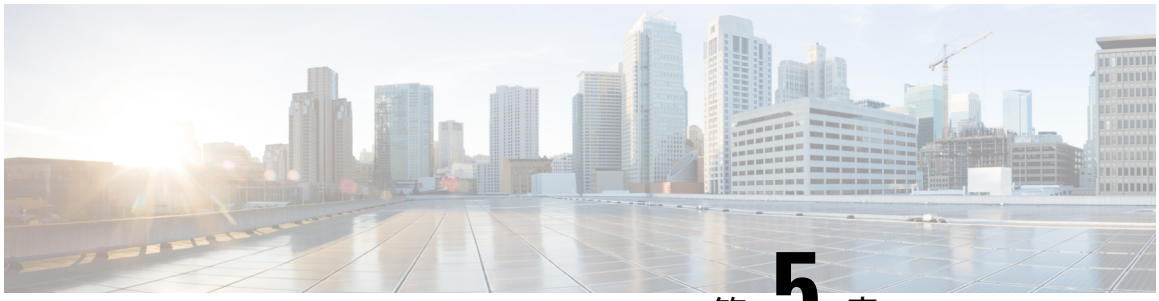
RADIUS の機能の履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	RADIUS	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。RADIUS クライアントは、サポート対象のシスコ デバイス上で稼働します。クライアントは中央の RADIUS サーバに認証要求を送ります。中央の RADIUS サーバにはすべてのユーザ認証情報、ネットワーク サービス アクセス情報が登録されています。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 5 章

アカウンティングの設定

AAA アカウンティング機能を使用すると、ユーザがアクセスするサービス、およびユーザが消費するネットワーク リソース量を追跡できます。AAA アカウンティングをイネーブルにすると、ネットワーク アクセス サーバから TACACS+ または RADIUS セキュリティ サーバ（実装しているセキュリティ手法によって異なります）に対して、アカウンティング レコードの形式でユーザ アクティビティがレポートされます。各アカウンティング レコードにはアカウンティングの Attribute-Value（AV）ペアが含まれ、レコードはセキュリティ サーバに格納されます。このデータを分析して、ネットワーク管理、クライアント課金、および監査に利用できます。

アカウンティングを設定するための前提条件

次のタスクを実行してから、名前付き方式リストを使用してアカウンティングを設定します。

- ネットワークアクセスサーバで AAA を有効にするには、グローバル コンフィギュレーション モードで **aaa new-model** コマンドを使用します。
- RADIUS または TACACS+ 認可が発行されている場合、RADIUS または TACACS+ セキュリティ サーバの特性を定義します。Cisco ネットワーク アクセス サーバを設定して RADIUS セキュリティ サーバと通信する方法の詳細については、「RADIUS の設定」モジュールを参照してください。Cisco ネットワークアクセスサーバを設定して TACACS+ セキュリティ サーバと通信する方法の詳細については、「TACACS+ の設定」モジュールを参照してください。

アカウンティングの設定の制約事項

- アカウンティング情報は、最大 4 台の AAA サーバにのみ同時送信できます。

アカウンティングの設定に関する情報

アカウンティングの名前付き方式リスト

認証および認可方式リストと同様に、アカウンティングの方式リストには、アカウンティングの実行方法とその方式を実行するシーケンスが定義されています。

アカウンティングの名前付き方式リストには、特定のセキュリティプロトコルを指定し、アカウンティングサービスの特定の行またはインターフェイスに使用できます。唯一の例外は、デフォルトの方式リスト（「default」という名前）です。デフォルトの方式リストは、明示的に定義された名前付きの方式リストを持つインターフェイスを除くすべてのインターフェイスに自動的に適用されます。定義済みの方式リストは、デフォルトの方式リストに優先します。

方式リストは、シーケンスで照会されるアカウンティング方式（RADIUS、TACACS+ など）を説明する単なる名前付きリストです。方式リストでは、アカウンティングに1つまたは複数のセキュリティプロトコルを指定できます。そのため、最初の方式が失敗した場合に備えてアカウンティングのバックアップ システムを確保できます。Cisco IOS ソフトウェアでは、リストされている最初の方式を使用して、アカウンティングをサポートします。その方式が応答しない場合、リストされている次のアカウンティング方式が選択されます。このプロセスは、リストのいずれかのアカウンティング方式と通信に成功するか、定義されているすべての方式が試行されるまで継続されます。

（注） Cisco IOS ソフトウェアでは、前の方式で応答が得られない場合にのみ、リストされている次のアカウンティング方式でアカウンティングが試行されます。このサイクルの任意の時点でアカウンティングが失敗した場合（つまり、セキュリティサーバからユーザアクセスの拒否応答が返される場合）、アカウンティングプロセスは停止し、その他のアカウンティング方式は試行されません。

アカウンティング方式リストは、要求されるアカウンティングの種類によって変わります。AAA は、次の 7 種類のアカウンティングをサポートしています。

- **Network** ：パケットやバイトカウントなど、すべての PPP、SLIP、または ARAP セッションに関する情報を提供します。
- **EXEC** ：ネットワークアクセスサーバのユーザ EXEC ターミナルセッションに関する情報を提供します。

- **Commands**：ユーザが発行するEXECモードコマンドに関する情報を提供します。コマンド アカウンティングは、特定の特権レベルに関連付けられた、グローバル コンフィギュレーション コマンドなどのすべてのEXEC モード コマンドについて、アカウンティング レコードを生成します。
- **Connection**：Telnet、ローカルエリア トランスポート（LAT）、TN3270、パケットアセンブラ/ディスアセンブラ（PAD）、rloginなどのネットワークアクセスサーバから行われたすべてのアウトバンド接続に関する情報を提供します。
- **System**：システムレベルのイベントに関する情報を提供します。
- **Resource**：ユーザ認証に成功したコールの「開始」および「終了」レコードを提供します。また、認証に失敗したコールの「終了」レコードを提供します。
- **VRRS**：Virtual Router Redundancy Service（VRRS）に関する情報を提供します。

（注） システム アカウンティングは、名前付きアカウンティング リストを使用しません。システム アカウンティングのデフォルト リストだけを定義できます。

この場合も、名前付き方式リストが作成されると、指定したアカウンティング タイプのアカウンティング方式のリストが定義されます。

アカウンティング方式リストを特定の回線またはインターフェイスに適用してから、定義済み方式のいずれかを実行する必要があります。唯一の例外は、デフォルトの方式リスト（「default」という名前）です。名前付き方式リストを指定せずに、特定のアカウンティングタイプに対して **aaa accounting** コマンドを発行すると、明示的に名前付き方式リストが定義されている場合を除き、すべてのインターフェイスまたは回線にデフォルトの方式リストが自動的に適用されます（定義した方式リストは、デフォルトの方式リストよりも優先されます）。デフォルトの方式リストが定義されていない場合、アカウンティングは実行されません。

ここでは、次の内容について説明します。

方式リストとサーバ グループ

サーバ グループは、方式リストに使用する既存のLDAP、RADIUS、またはTACACS+サーバ ホストをグループ化する方法の1つです。次の図に、4台のセキュリティ サーバ（R1とR2はRADIUSサーバ、T1とT2はTACACS+サーバ）が設置された一般的なAAAネットワーク設定を示します。R1とR2でRADIUSサーバのグループを構成します。T1とT2でTACACS+サーバのグループを構成します。

サーバ グループを使用して、設定したサーバ ホストのサブセットを指定し、特定のサービスに使用します。たとえば、サーバ グループを使用すると、R1およびR2を1つのサーバ グループとして定義し、T1およびT2を別のサーバ グループとして定義できます。R1とT1を方式リストに指定することや、R2とT2を方式リストに指定することができます。そのため、RADIUSおよびTACACS+のリソースを割り当てる場合の柔軟性が高くなります。

サーバ グループには、1台のサーバに対して複数のホスト エントリを含めることができます。エントリごとに固有の識別情報を設定します。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（たとえば許可）を設定した場合、2 番目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フェール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 番目のホスト 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）。

AAA アカウンティング方式

Cisco IOS ソフトウェアはアカウンティングについて次の 2 つの方式をサポートします。

- TACACS+：ネットワークアクセスサーバは、アカウンティングレコードの形式でTACACS+セキュリティサーバに対してユーザアクティビティを報告します。各アカウンティング レコードは、アカウンティング AV ペアが含まれ、セキュリティ サーバ上で保管されます。
- RADIUS：ネットワークアクセスサーバは、アカウンティングレコードの形式で RADIUS セキュリティサーバに対してユーザアクティビティを報告します。各アカウンティング レコードは、アカウンティング AV ペアが含まれ、セキュリティ サーバ上で保管されます。

（注） パスワードおよびアカウンティングログは、TACACS+ または RADIUS セキュリティサーバへ送信される前にマスクされます。マスクされていない情報を TACACS+ または RADIUS セキュリティサーバに送信するには、**aaa accounting commands visible-keys** コマンドを使用します。

アカウンティング レコードの種類

最小限のアカウンティングの場合、**stop-only** キーワードを使用します。このキーワードによって、要求されたユーザプロセスの終了時に、終了レコードアカウンティング通知を送信するよう、指定した方式（**RADIUS** または **TACACS+**）に指示します。詳細なアカウンティング情報が必要な場合、**start-stop** キーワードを使用して、要求されたイベントの開始時には開始アカウンティング通知、そのイベントの終了時には修理用アカウンティング通知を送信します。この回線またはインターフェイスですべてのアカウンティングアクティビティを終了するには、**none** キーワードを使用します。

AAA アカウンティング タイプ

この項では、さまざまな AAA アカウンティングタイプについて説明します。

ネットワーク アカウンティング

ネットワーク アカウンティングは、パケットやバイト カウントなど、すべての PPP、SLIP、または ARAP セッションに関する情報を提供します。

次に、EXEC セッションを介して着信する PPP ユーザの RADIUS ネットワーク アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 04:44:45 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 5
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “562”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = “0000000D”
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”

Wed Jun 27 04:45:00 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 5
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “562”
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = “0000000E”
        Framed-IP-Address = “10.1.1.2”
        Framed-Protocol = PPP
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
Wed Jun 27 04:47:46 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 5
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
        Caller-ID = “562”
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = “0000000E”
        Framed-IP-Address = “10.1.1.2”
        Framed-Protocol = PPP
        Acct-Input-Octets = 3075
        Acct-Output-Octets = 167
        Acct-Input-Packets = 39
        Acct-Output-Packets = 9
        Acct-Session-Time = 171
        Acct-Delay-Time = 0
        User-Id = “username1”
        NAS-Identifier = “172.16.25.15”
Wed Jun 27 04:48:45 2001
        NAS-IP-Address = “172.16.25.15”
        NAS-Port = 5
        User-Name = “username1”
        Client-Port-DNIS = “4327528”
```

```
        Caller-ID = "408"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "0000000D"
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

次に、最初にEXECセッションを開始したPPPユーザのTACACS+ネットワーク アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 04:00:35 2001 172.16.25.15    username1   tty4    562/4327528
starttask_id=28      service=shell
Wed Jun 27 04:00:46 2001 172.16.25.15    username1   tty4    562/4327528
starttask_id=30      addr=10.1.1.1   service=ppp
Wed Jun 27 04:00:49 2001 172.16.25.15    username1   tty4    408/4327528
updatetask_id=30     addr=10.1.1.1   service=ppp     protocol=ip     addr=10.1.1.1
Wed Jun 27 04:01:31 2001 172.16.25.15    username1   tty4    562/4327528     stoptask_id=30
       addr=10.1.1.1   service=ppp     protocol=ip     addr=10.1.1.1   bytes_in=2844
     bytes_out=1682  paks_in=36      paks_out=24     elapsed_time=51
Wed Jun 27 04:01:32 2001 172.16.25.15    username1   tty4    562/4327528     stoptask_id=28
       service=shell   elapsed_time=57
```

（注） アカウンティング パケット レコードの正確なフォーマットは、セキュリティ サーバ デーモンに応じて変わります。

次に、autoselectを介して着信するPPPユーザのRADIUSネットワーク アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 04:30:52 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 3
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000B"
        Framed-Protocol = PPP
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:36:49 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 3
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000B"
        Framed-Protocol = PPP
        Framed-IP-Address = "10.1.1.1"
        Acct-Input-Octets = 8630
```

```
        Acct-Output-Octets = 5722
        Acct-Input-Packets = 94
        Acct-Output-Packets = 64
        Acct-Session-Time = 357
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

次に、autoselect を介して着信する PPP ユーザの TACACS+ ネットワーク アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 04:02:19 2001 172.16.25.15    username1   Async5  562/4327528
starttask_id=35      service=ppp
Wed Jun 27 04:02:25 2001 172.16.25.15    username1   Async5  562/4327528
updatetask_id=35     service=ppp     protocol=ip     addr=10.1.1.2
Wed Jun 27 04:05:03 2001 172.16.25.15   username1   Async5  562/4327528    stoptask_id=35
       service=ppp     protocol=ip     addr=10.1.1.2   bytes_in=3366   bytes_out=2149
     paks_in=42      paks_out=28     elapsed_time=164
```

EXEC アカウンティング

EXEC アカウンティングは、ネットワーク アクセス サーバ上にあるユーザ EXEC ターミナル セッション（ユーザ シェル）に関する情報を提供します。たとえば、ユーザ名、日付、開始時刻と終了時刻、アクセス サーバの IP アドレス、および（ダイヤルイン ユーザの場合）発信元の電話番号などです。

次に、ダイヤルイン ユーザの RADIUS EXEC アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 04:26:23 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 1
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329483"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000006"
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
Wed Jun 27 04:27:25 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 1
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329483"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000006"
        Acct-Session-Time = 62
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

次に、ダイヤルイン ユーザの TACACS+ EXEC アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 03:46:21 2001        172.16.25.15    username1   tty3    5622329430/4327528
 start    task_id=2       service=shell
Wed Jun 27 04:08:55 2001        172.16.25.15    username1   tty3    5622329430/4327528
 stop     task_id=2       service=shell   elapsed_time=1354
```

次に、Telnetユーザの RADIUS EXEC アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 04:48:32 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 26
        User-Name = "username1"
        Caller-ID = "10.68.202.158"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000010"
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:48:46 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 26
        User-Name = "username1"
        Caller-ID = "10.68.202.158"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000010"
        Acct-Session-Time = 14
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

次に、Telnet ユーザの TACACS+ EXEC アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 04:06:53 2001        172.16.25.15    username1   tty26   10.68.202.158
starttask_id=41      service=shell
Wed Jun 27 04:07:02 2001        172.16.25.15    username1   tty26   10.68.202.158
stoptask_id=41       service=shell   elapsed_time=9
```

コマンド アカウンティング

コマンド アカウンティングは、ネットワーク アクセス サーバで実行される各特権レベルの EXEC シェル コマンドに関する情報を提供します。各コマンド アカウンティング レコードには、その特権レベルで実行されるコマンド、各コマンドが実行された日時、および実行したユーザのリストが含まれます。

次に、特権レベル 1 の TACACS+ コマンド アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 03:46:47 2001        172.16.25.15    username1   tty3    5622329430/4327528
 stop     task_id=3       service=shell   priv-lvl=1      cmd=show version <cr>
Wed Jun 27 03:46:58 2001        172.16.25.15    username1   tty3    5622329430/4327528
 stop     task_id=4       service=shell   priv-lvl=1      cmd=show interfaces Ethernet
0 <cr>
```

```
Wed Jun 27 03:47:03 2001        172.16.25.15    username1   tty3    5622329430/4327528
 stop     task_id=5       service=shell   priv-lvl=1      cmd=show ip route <cr>
```

次に、特権レベル 15 の TACACS+ コマンド アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 03:47:17 2001        172.16.25.15    username1   tty3    5622329430/4327528
 stop     task_id=6       service=shell   priv-lvl=15     cmd=configure terminal <cr>
Wed Jun 27 03:47:21 2001        172.16.25.15    username1   tty3    5622329430/4327528
 stop     task_id=7       service=shell   priv-lvl=15     cmd=interface Serial 0 <cr>
Wed Jun 27 03:47:29 2001        172.16.25.15    username1   tty3    5622329430/4327528
 stop     task_id=8       service=shell   priv-lvl=15     cmd=ip address 10.1.1.1
255.255.255.0 <cr>
```

（注） Cisco の RADIUS 実装は、コマンド アカウンティングをサポートしていません。

接続アカウンティング

接続アカウンティングは、Telnet、LAT、TN3270、PAD、rlogin などのネットワーク アクセス サーバから行われるすべての発信接続に関する情報を提供します。

次に、発信 Telnet 接続の RADIUS 接続アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 04:28:00 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "00000008"
        Login-Service = Telnet
        Login-IP-Host = "10.68.202.158"
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:28:39 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "00000008"
        Login-Service = Telnet
        Login-IP-Host = "10.68.202.158"
        Acct-Input-Octets = 10774
        Acct-Output-Octets = 112
        Acct-Input-Packets = 91
        Acct-Output-Packets = 99
        Acct-Session-Time = 39
```

```
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

次に、発信 Telnet 接続の TACACS+ 接続アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 03:47:43 2001        172.16.25.15    username1   tty3    5622329430/4327528
 start    task_id=10      service=connection      protocol=telnet addr=10.68.202.158
cmd=telnet username1-sun
Wed Jun 27 03:48:38 2001        172.16.25.15    username1   tty3    5622329430/4327528
 stop     task_id=10      service=connection      protocol=telnet addr=10.68.202.158
cmd=telnet username1-sun    bytes_in=4467   bytes_out=96    paks_in=61      paks_out=72
 elapsed_time=55
```

次に、発信 rlogin 接続の RADIUS 接続アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 04:29:48 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "0000000A"
        Login-Service = Rlogin
        Login-IP-Host = "10.68.202.158"
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:30:09 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "username1"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "0000000A"
        Login-Service = Rlogin
        Login-IP-Host = "10.68.202.158"
        Acct-Input-Octets = 18686
        Acct-Output-Octets = 86
        Acct-Input-Packets = 90
        Acct-Output-Packets = 68
        Acct-Session-Time = 22
        Acct-Delay-Time = 0
        User-Id = "username1"
        NAS-Identifier = "172.16.25.15"
```

次に、発信 rlogin 接続の TACACS+ 接続アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 03:48:46 2001        172.16.25.15    username1   tty3    5622329430/4327528
 start    task_id=12      service=connection      protocol=rlogin addr=10.68.202.158
cmd=rlogin username1-sun /user username1
```

```
Wed Jun 27 03:51:37 2001        172.16.25.15    username1   tty3    5622329430/4327528
 stop     task_id=12      service=connection      protocol=rlogin addr=10.68.202.158
cmd=rlogin username1-sun /user username1 bytes_in=659926 bytes_out=138   paks_in=2378
  paks_
out=1251        elapsed_time=171
```

次に、発信 LAT 接続の TACACS+ 接続アカウンティング レコードに含まれる情報の例を示します。

```
Wed Jun 27 03:53:06 2001        172.16.25.15    username1   tty3    5622329430/4327528
 start    task_id=18      service=connection      protocol=lat    addr=VAX        cmd=lat
 VAX
Wed Jun 27 03:54:15 2001        172.16.25.15    username1   tty3    5622329430/4327528
 stop     task_id=18      service=connection      protocol=lat    addr=VAX        cmd=lat
 VAX  bytes_in=0      bytes_out=0     paks_in=0      paks_out=0      elapsed_time=6
```

システム アカウンティング

システム アカウンティングは、すべてのシステムレベル イベント（たとえば、システムのリブート時やアカウンティングのオン/オフ時）に関する情報を提供します。

次のアカウンティング レコードは、AAA アカウンティングがオフになったことを示す一般的な TACACS+ システム アカウンティング レコード サーバを示します。

```
Wed Jun 27 03:55:32 2001       172.16.25.15    unknown unknown unknown start   task_id=25

service=system  event=sys_acct  reason=reconfigure
```

（注） アカウンティング パケット レコードの正確なフォーマットは、TACACS+デーモンに応じて変わります。

次のアカウンティング レコードは、AAAアカウンティングがオンになったことを示すTACACS+システム アカウンティング レコードを示します。

```
Wed Jun 27 03:55:22 2001       172.16.25.15    unknown unknown unknown stop    task_id=23

service=system  event=sys_acct  reason=reconfigure
```

リソース アカウンティング

Cisco IOS が採用している AAA アカウンティングでは、ユーザ認証を通過したコールに対する開始レコードと終了レコードがサポートされます。ユーザ認証の一部として認証に失敗したコールの終了レコードを生成する追加機能もサポートされます。このようなレコードは、ネットワークを管理およびモニタするアカウンティング レコードを採用する場合に必要です。

ここでは、次の内容について説明します。

AAA ブロードキャスト アカウンティング

AAA ブロードキャスト アカウンティングを有効にすると、アカウンティング情報を複数の AAA サーバに同時に送信できます。つまり、アカウンティング情報を 1 つまた複数の AAA サーバに同時にブロードキャストすることが可能です。この機能を使用すると、サービスプロバイダーは自社使用のプライベート AAA サーバやエンドユーザの AAA サーバにアカウンティング情報を送信できるようになります。この機能では、音声アプリケーションによる課金情報も提供されます。

ブロードキャストは、RADIUS または TACACS+ サーバのグループに使用できます。また、各サーバグループは、他のグループとは関係なく、フェールオーバーの場合のバックアップ サーバを定義できます。

したがって、サービス プロバイダーとそのエンド ユーザは、アカウンティング サーバに異なるプロトコル（RADIUS または TACACS+）を使用できます。また、サービス プロバイダーとそのエンド ユーザは、それぞれ単独でバックアップ サーバを指定することもできます。音声アプリケーションについては、独自のフェールオーバー シーケンスを持つ個別のグループを介して、冗長的なアカウンティング情報を単独で管理できます。

AAA セッション MIB

ユーザが AAA セッション MIB 機能を使用すると、簡易ネットワーク管理プロトコル（SNMP）を使用して自身の認証済みクライアント接続をモニタおよび終了できます。そのクライアントのデータが提示されるため、RADIUS または TACACS+ サーバから報告される AAA アカウンティング情報に直接関連付けることができます。AAA セッション MIB は、次の情報を提供します。

- 各 AAA 機能の統計情報（**show radius statistics** コマンドと併用する場合）
- AAA 機能を提供するサーバのステータス
- 外部 AAA サーバの ID
- （アイドル時間などの）リアルタイム情報（アクティブ コールを終了するかどうかを評価する SNMP ネットワークが使用する追加基準を提供します）

次の表に、認証済みクライアントと AAA セッション MIB 機能との接続をモニタおよび終了するために使用できる SNMP ユーザエンド データ オブジェクトを示します。

表 *11 : SNMP* エンドユーザ データ オブジェクト

SessionId	AAA アカウンティング プロトコルに使用されるセッション ID（RADIUS 属性 44（Acct-Session-ID）から報告される値と同じ）
UserId	ユーザ ログイン ID または（ログインが使用できない場合）長さがゼロの文字列
IpAddr	セッションの IP アドレスまたは（IP アドレスが適用されない場合、または使用できない場合）0.0.0.0

IdleTime	セッションがアイドルになってからの経過時間
Disconnect	そのクライアントとの接続を解除するために使用されるセッション終了オブジェクト
CallId	コール トラッカー レコードが保存した、このアカウンティング セッションに対応するエントリ インデックス

次の表に、システム別に SNMP を使用する AAA セッション MIB 機能から提供される AAA の概要情報を示します。

表 *12 : SNMP AAA* セッションの概要

ActiveTableEntries	現在アクティブなセッションの数
ActiveTableHighWaterMark	システムが最後に再インストールされてからの同時接続セッションの最大数
TotalSessions	システムが最後に再インストールされてからのセッションの合計数
DisconnectedSessions	システムが最後に再インストールされてから接続解除されたセッションの合計数

アカウンティング属性と値のペア

ネットワーク アクセス サーバは、TACACS+ AV のペアまたは RADIUS 属性（実装しているセキュリティ方式によって異なります）に定義されたアカウンティング機能をモニタします。

アカウンティングの設定方法

名前付き方式リストによる AAA アカウンティングの設定

名前付き方式リストを使用して AAA アカウンティングを設定するには、次の手順を実行します。

（注） システム アカウンティングは、名前付き方式リストを使用しません。システム アカウンティングの場合、デフォルトの方式リストだけを定義します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa accounting** {**system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} [*method1* [*method2...*]] 例： `Device(config)# aaa accounting system default start-stop`	アカウンティング方式リストを作成し、アカウンティングを有効にします。引数 *list-name* は、作成したリストに名前を付けるときに使用される文字列です。
ステップ 4	次のいずれかを実行します。 • **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*] • **interface** *interface-type interface-number* 例： `Device(config)# line aux line1`	アカウンティング方式リストを適用する回線について、ライン コンフィギュレーション モードを開始します。 または アカウンティング方式リストを適用するインターフェイスについて、インターフェイス コンフィギュレーション モードを開始します。
ステップ 5	次のいずれかを実行します。 • **accounting** {**arap** | **commands** *level* | **connection** | **exec**} {**default** | *list-name*} • **ppp accounting**{**default** | *list-name*} 例： `Device(config-line)# accounting arap default`	1 つの回線または複数回線にアカウンティング方式リストを適用します。 または 1 つのインターフェイスまたは複数インターフェイスにアカウンティング方式リストを適用します。
ステップ 6	**end** 例： `Device(config-line)# end`	（任意）ライン コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

RADIUS システム アカウンティングの設定

このタスクを実行して、グローバル RADIUS サーバで RADIUS システム アカウンティングを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA ネットワーク セキュリティ サービスをイネーブルにします。
ステップ 4	**radius-server accounting system host-config** 例： `Device(config)# radius-server accounting system host-config`	RADIUS サーバの追加および削除のために、デバイスからシステムアカウンティング レコードを送信できるようにします。
ステップ 5	**aaa group server radius** *server-name* 例： `Device(config)# aaa group server radius radgroup1`	RADIUS サーバを追加し、server-group コンフィギュレーション モードを開始します。 • *server-name* 引数には、RADIUS サーバ グループ名を指定します。
ステップ 6	**server-private** {*host-name* | *ip-address*} **key** {[**0** *server-key* | **7** *server-key*] *server-key* 例： `Device(config-sg-radius)# server-private 172.16.1.11 key cisco`	RADIUS サーバのホスト名または IP アドレスと、非表示のサーバ キーを入力します。 • （任意） **0** と *server-key* 引数により、暗号化されていない（クリアテキストの）非表示のサーバキーが後に続くことを指定します。 • （任意）**7** と *server-key* 引数により、暗号化されている非表示のサー

	コマンドまたはアクション	目的
		バキーが後に続くことを指定します。 • *server-key* 引数は、非表示のサーバキーを指定します。*server-key*引数の前に **0** も **7**も付いていない場合、サーバキーは暗号化されません。 （注） **server-private** コマンドが設定されると、RADIUS システム アカウンティングが有効になります。
ステップ 7	**accounting system host-config** 例： `Device(config-sg-radius)# accounting system host-config`	プライベート サーバ ホストの追加または削除時に、システム アカウンティング レコードの生成をイネーブルにします。
ステップ 8	**end** 例： `Device(config-sg-radius)# end`	サーバ グループ コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

ヌル ユーザ名セッション時のアカウンティング レコード生成の抑制

AAA アカウンティングをアクティブにすると、Cisco IOS ソフトウェアは、システム上のすべてのユーザにアカウンティング レコードを発行します。このとき、プロトコル変換のためユーザ名文字列がヌルになっているユーザも含まれます。この例では、**aaa authentication login** *method-list* **none** コマンドが適用される回線で着信するユーザがそれに該当します。関連付けられているユーザ名がないセッションについて、アカウンティング レコードが生成されないようにするには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
`Device(config)# aaa accounting suppress null-username`	ユーザ名文字列がヌルのユーザについて、アカウンティング レコードが生成されないようにします。

中間アカウンティング レコードの生成

アカウンティング サーバに定期的な中間アカウンティング レコードを送信できるようにするには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
`Device(config)# aaa accounting update [newinfo] [periodic] number`	アカウンティング サーバに送信される定期的中間アカウンティング 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、Cisco IOS ソフトウェアによってシステム上のすべてのユーザの中間アカウンティングレコードが発行されます。**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ティング情報が発生するたびに、中間アカウンティングレコードがアカウンティ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。中間アカウンティング 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。

キーワード **periodic** と一緒に使用した場合は、*number* 引数による定義に基づいて、中間アカウンティングレコードが定期的に送信されます。中間アカウンティング レコードには、中間アカウンティング 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ティング情報が含まれます。

⚠

注意 多数のユーザがネットワークにログインしている場合には、**aaa accounting update periodic** コマンドを使用すると、重度の輻輳が発生する可能性があります。

失敗したログインまたはセッションに対するアカウンティングレコードの生成

AAA アカウンティングをアクティブにすると、Cisco IOS ソフトウェアは、ログイン認証に失敗したシステム ユーザや、ログイン認証には成功しても何らかの理由で PPP ネゴシエーションに失敗したシステム ユーザには、アカウンティング レコードを生成しません。

ログイン時またはセッション ネゴシエーション中の認証に失敗したユーザについて、アカウンティング終了レコードを生成するように指定するには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
`Device(config)# aaa accounting send stop-record authentication failure`	ログイン時またはセッション ネゴシエーション中の認証に失敗したユーザについて、「終了」レコードを生成します。
`Device(config)# aaa accounting send stop-record always`	開始レコードが送信済みかどうかに関係なく、AAA 終了レコードを送信します。

EXEC-Stop レコードよりも前のアカウンティング NETWORK-Stop レコードの指定

PPP ユーザが EXEC ターミナル セッションを開始する場合、EXEC 終了レコードの前に生成する NETWORK レコードを指定できます。特定のサービスについて顧客に課金する場合など、状況によっては、ネットワークの開始レコードと終了レコードを一緒に保持する方が望ましいことがあります。その際、基本的に、EXEC の開始メッセージと終了メッセージのフレームワーク内に「ネスト」にします。たとえば、PPP を使用するユーザ ダイヤルインによって、EXEC-start、NETWORK-start、EXEC-stop、NETWORK-stop というレコードを作成できます。アカウンティング レコードをネストにすることで、NETWORK-stop レコードは NETWORK-start メッセージ（EXEC-start、NETWORK-start、NETWORK-stop、EXEC-stop）に従います。

ユーザ セッションのアカウンティング レコードをネストするには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
`Device(config)# aaa accounting nested`	ネットワーク アカウンティング レコードをネストします。

AAA リソース失敗終了アカウンティングの設定

リソース失敗終了アカウンティングを有効にするには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
`Device(config)# aaa accounting resource` *method-list* `stop-failure group` *server-group*	ユーザ認証に到達しないコールについて、終了レコードを生成します。

開始 - 終了レコードの AAA リソース アカウンティングの設定

開始 - 終了レコードのフル リソース アカウンティングをイネーブルにするには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
`Device(config)# aaa accounting resource` *method-list* **start-stop group** *server-group*	各コール設定時に開始レコードを送信し、コールの接続解除時に対応する終了レコードを送信する機能をサポートします。

AAA ブロードキャスト アカウンティングの設定

AAA ブロードキャスト アカウンティングを設定するには、グローバル コンフィギュレーション モードで **aaa accounting** コマンドを使用します。

コマンド	目的
Device(config)# **aaa accounting** {**system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} [**broadcast**] *method1* [*method2*...]	

コマンド	目的
	複数の **A** サバに対するアカウンティングレコードの送信をイネブルにします。各グループの最初のサバ

コマンド	目的
	に対しアカウンティングレコードを同時に送信します最初のサーバが使用できない場合はセカンダリオーバが

コマンド	目的
	発生しそのグループ内に定義されているバックアップ サーバが使用されます。

DNIS による AAA ブロードキャスト アカウンティングの設定

AAA ブロードキャスト アカウンティングを設定するには、グローバル コンフィギュレーション モードで **aaa dnis map accounting network** コマンドを使用します。

コマンド	目的
`Device(config)# aaa dnis map dnis-number accounting network [start-stop | stop-only | none] [broadcast] method1 [method2...]`	DNIS によるアカウンティングの設定を許可します。このコマンドは、グローバルの **aaa accounting** コマンドよりも優先されます。 複数の AAA サーバに対するアカウンティング レコードの送信をイネーブルにします。各グループの最初のサーバに対し、アカウンティング レコードを同時に送信します。最初のサーバが使用できない場合はフェールオーバーが発生し、そのグループ内に定義されているバックアップ サーバが使用されます。

AAA セッション MIB の設定

次のタスクは、次の AAA セッション MIB 機能の設定よりも前に実行する必要があります。

- SNMP を設定します。
- AAA を設定します。
- RADIUS または TACACS+ サーバの特性を定義します。

（注） SNMP を多用すると、全体のシステム パフォーマンスに影響が出る可能性があります。そのため、この機能を使用するときに、通常のネットワーク管理パフォーマンスを考慮する必要があります。

AAA セッション MIB を設定するには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	Device (config)# **aaa session-mib disconnect**	SNMP を使用して、認証済みクライアント接続をモニタおよび終了します。 コールを終了するには、**disconnect** キーワードを使用する必要があります。

AAA サーバが到達不能な場合のデバイスとのセッションの確立

AAA サーバが到達不能の場合に、デバイスとの間にコンソールまたは Telnet セッションを確立するには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
`Device(config)# no aaa accounting system guarantee-first`	最初のレコードとしてシステムアカウンティングを保証します（これがデフォルトの条件です）。 状況によっては、システムの再ロードが完了するまで（3 分よりも長くかかる可能性があります）、ユーザがコンソールまたは Telnet 接続でセッションを開始できない可能性があります。この問題を解決するために、**no aaa accounting system guarantee-first** コマンドを使用できます。

（注） **no aaa accounting system guarantee-first** コマンドの入力は、コンソールセッションまたは Telnet セッションを起動可能にするための唯一の条件ではありません。たとえば、特権 EXEC セッションが TACACS+ によって認証され、TACACS+ サーバが到達不能の場合、セッションは開始できません。

アカウンティングのモニタリング

RADIUS または TACACS+ アカウンティングの場合、特定の **show** コマンドは存在しません。現在ログインしているユーザに関する情報を表示するアカウンティング レコードを取得するには、特権 EXEC モードで次のコマンドを使用します。

コマンド	目的
`Device# show accounting`	ネットワークでアクティブなアカウント可能なイベントの表示を許可し、アカウンティング サーバでデータが損失した場合に情報を収集できます。

アカウンティングのトラブルシューティング

アカウンティング情報の問題を解決するには、特権 EXEC モードで次のコマンドを使用します。

コマンド	目的
`Device# debug aaa accounting`	説明の義務があるイベントが発生したときに、その情報を表示します。

アカウンティングの設定例

例：名前付き方式リストの設定

次に、RADIUS サーバから AAA サービスを提供するためにデバイス（AAA および RADIUS セキュリティサーバとの通信で有効）を設定する例を示します。RADIUS サーバが応答に失敗すると、認証情報と許可情報についてローカルデータベースへの照会が行われ、アカウンティングサービスは TACACS+ サーバによって処理されます。

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication login admins local
Device(config)# aaa authentication ppp dialins group radius local
Device(config)# aaa authorization network blue1 group radius local
Device(config)# aaa accounting network red1 start-stop group radius group tacacs+
Device(config)# username root password ALongPassword
Device(config)# tacacs server server1
Device(config-tacacs-server)# address IPv4 172.31.255.0
Device(config-tacacs-server)# key goaway
Device(config-tacacs-server)# exit
Device(config)# radius server server2
Device(config-radius-server)# address IPv4 172.16.2.7
Device(config-radius-server)# key myRaDiUSpassWoRd
Device(config-radius-server)# exit
Device(config)# interface group-async 1
Device(config-if)# group-range 1 16
Device(config-if)# encapsulation ppp
Device(config-if)# ppp authentication chap dialins
Device(config-if)# ppp authorization blue1
Device(config-if)# ppp accounting red1
Device(config-if)# exit
Device(config)# line 1 16
Device(config-line)# autoselect ppp
Device(config-line)# autoselect during-login
Device(config-line)# login authentication admins
Device(config-line)# modem dialin
Device(config-line)# end
```

この RADIUS AAA 設定のサンプル行は、次のように定義されます。

- **aaa new-model** コマンドは、AAA ネットワーク セキュリティ サービスをイネーブルにします。
- **aaa authentication login admins local** コマンドは、ログイン認証に方式リスト「admins」を定義します。
- **aaa authentication ppp dialins group radius local** コマンドで、認証方式リスト「dialins」を定義します。このリストは、最初にRADIUS認証を指定して、次に（RADIUSサーバが応答しない場合）PPP を使用してシリアル回線上でローカル認証が使用されます。
- **aaa authorization network blue1 group radius local** コマンドで、「blue1」というネットワーク許可方式リストを定義します。これにより、PPP を使用してシリアル回線上で RADIUS

許可を使用するよう指定されます。RADIUS サーバが応答に失敗すると、ローカル ネットワークの許可が実行されます。

- **aaa accounting network red1 start-stop group radius group tacacs+** コマンドで、「red1」というネットワーク アカウンティング方式リストを定義します。これにより、PPP を使用してシリアル回線上で RADIUS アカウンティング サービス（この場合、特定のイベントに対する開始レコードと終了レコード）を使用するよう指定されます。RADIUS サーバが応答に失敗すると、アカウンティングサービスは TACACS+ サーバによって処理されます。
- **username** コマンドはユーザ名とパスワードを定義します。これらの情報は、PPP パスワード認証プロトコル（PAP）の発信元身元確認に使用されます。
- **tacacs server** コマンドは TACACS+ サーバーホスト名を定義し、**key** コマンドは、ネットワーク アクセス サーバーと TACACS+ サーバーホストの間の共有秘密テキスト文字列を定義します。
- **radius server** コマンドは RADIUS サーバーホスト名を定義し、**key** コマンドは、ネットワーク アクセス サーバーと RADIUS サーバーホストの間の共有秘密テキスト文字列を定義します。
- **interface group-async** コマンドは、非同期インターフェイス グループを選択して定義します。
- **group-range** コマンドは、インターフェイス グループ内のメンバ非同期インターフェイスを定義します。
- **encapsulation ppp** コマンドは、指定のインターフェイスに使用されるカプセル化方式として PPP を設定します。
- **ppp authentication chap dialins** コマンドは、PPP 認証方式としてチャレンジ ハンドシェイク認証プロトコル（CHAP）を選択し、特定のインターフェイスに「dialins」方式リストを適用します。
- **ppp authorization blue1** コマンドによって、blue1 ネットワーク許可方式リストが、指定したインターフェイスに適用されます。
- **ppp accounting red1** コマンドによって、red1 ネットワーク アカウンティング方式リストが、指定したインターフェイスに適用されます。
- **line** コマンドはコンフィギュレーション モードをグローバル コンフィギュレーションからライン コンフィギュレーションに切り替え、設定対象の回線を指定します。
- **autoselect ppp** コマンドは、選択した回線上で PPP セッションを自動的に開始できるように Cisco IOS ソフトウェアを設定します。
- **autoselect during-login** コマンドを使用すると、Return キーを押さずにユーザ名およびパスワードのプロンプトが表示されます。ユーザがログインすると、autoselect機能（この場合は PPP）が開始します。
- **login authentication admins** コマンドは、ログイン認証に admins 方式リストを適用します。

- **modem dialin** コマンドは、選択した回線に接続されているモデムを設定し、着信コールだけを受け入れるようにします。

show accounting コマンドを使用すると、前述の設定に関する出力が次のように生成されます。

```
Device# show accounting

Active Accounted actions on tty1, User username2 Priv 1
 Task ID 5, Network Accounting record, 00:00:52 Elapsed
 task_id=5 service=ppp protocol=ip address=10.0.0.98
```

次の表に、前述の出力に含まれるフィールドについて説明します。

表 *13 : show accounting* のフィールドの説明

フィールド	説明
Active Accounted actions on	ユーザがログインに使用する端末回線またはインターフェイス名
User	ユーザの ID。
Priv	ユーザの特権レベル。
Task ID	各アカウンティング セッションの固有識別情報
Accounting record	アカウンティング セッション タイプ
Elapsed	このセッション タイプの期間（hh:mm:ss）
attribute=value	このアカウンティング セッションに関連付けられている AV ペア

例：AAA リソース アカウンティングの設定

次に、リソース失敗終了アカウンティング、および 開始 - 終了レコード機能のリソース アカウンティングを設定する例を示します。

```
!Enable AAA on your network access server.
aaa new-model
!Enable authentication at login and list the AOL string name to use for login
authentication.
aaa authentication login AOL group radius local
!Enable authentication for ppp and list the default method to use for PPP authentication.
aaa authentication ppp default group radius local
!Enable authorization for all exec sessions and list the AOL string name to use for
authorization.
aaa authorization exec AOL group radius if-authenticated
!Enable authorization for all network-related service requests and list the default
method to use for all network-related authorizations.
aaa authorization network default group radius if-authenticated
!Enable accounting for all exec sessions and list the default method to use for all
start-stop accounting services.
aaa accounting exec default start-stop group radius
!Enable accounting for all network-related service requests and list the default method
 to use for all start-stop accounting services.
aaa accounting network default start-stop group radius
```

```
!Enable failure stop accounting.
aaa accounting resource default stop-failure group radius
!Enable resource accounting for start-stop records.
aaa accounting resource default start-stop group radius
```

例：AAA ブロードキャスト アカウンティングの設定

次に、グローバル **aaa accounting** コマンドを使用して、ブロードキャストアカウンティングを有効にする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# aaa group server radius isp
Device(config-sg-radius)# server 10.0.0.1
Device(config-sg-radius)# server 10.0.0.2
Device(config-sg-radius)# exit
Device(config)# aaa group server tacacs+ isp_customer
Device(config-sg-tacacs)# server 172.0.0.1
Device(config-sg-tacacs)# exit
Device(config)# aaa accounting network default start-stop broadcast group isp group
isp_customer
Device(config)# radius server server
Device(config-radius-server)# address IPv4 10.0.0.1
Device(config-radius-server)# key key_1
Device(config-radius-server)# exit
Device(config)# radius server server
Device(config-radius-server)# address IPv4 10.0.0.2
Device(config-radius-server)# key key_1
Device(config-radius-server)# exit
Device(config)# tacacs server server1
Device(config-tacacs-server)# address IPv4 172.0.0.1
Device(config-tacacs-server)# key key2
Device(config-tacacs-server)# end
```

broadcast キーワードによって、ネットワーク接続に関する開始および終了アカウンティングレコードが、グループ isp ではサーバ 10.0.0.1 に、グループ isp_customer ではサーバ 172.0.0.1 に同時送信されます。サーバ 10.0.0.1 が使用できなくなると、サーバ 10.0.0.2 へのフェールオーバーが行われます。サーバ 172.0.0.1 が使用できなくなっても、グループ isp_customer にはバックアップ サーバが設定されていないため、フェールオーバーは行われません。

例：DNIS による AAA ブロードキャスト アカウンティングの設定

次に、グローバル **aaa dnis map accounting network** コマンドを使用して、DNIS 単位のブロードキャスト アカウンティングを有効にする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# aaa group server radius isp
Device(config-sg-radius)# server 10.0.0.1
Device(config-sg-radius)# server 10.0.0.2
Device(config-sg-radius)# exit
Device(config)# aaa group server tacacs+ isp_customer
Device(config-sg-radius)# server 172.0.0.1
Device(config-sg-radius)# exit
Device(config)# aaa dnis map enable
Device(config)# aaa dnis map 7777 accounting network start-stop broadcast group isp group
 isp_customer
```

```
Device(config)# radius server server
Device(config-radius-server)# address IPv4 10.0.0.1
Device(config-radius-server)# key key_1
Device(config-radius-server)# exit
Device(config)# radius server server
Device(config-radius-server)# address IPv4 10.0.0.2
Device(config-radius-server)# key key_1
Device(config-radius-server)# exit
Device(config)# tacacs server server
Device(config-tacacs-server)# address IPv4 172.0.0.1
Device(config-tacacs-server)# key key_2
Device(config-tacacs-server)# end
s
```

broadcast キーワードによって、DNIS 番号 7777 のネットワーク接続コールに関する開始および終了アカウンティングレコードが、グループ isp ではサーバ 10.0.0.1 に、グループ isp_customer ではサーバ 172.0.0.1 に同時送信されます。サーバ 10.0.0.1 が使用できなくなると、サーバ 10.0.0.2 へのフェールオーバーが行われます。サーバ 172.0.0.1 が使用できなくなっても、グループ isp_customer にはバックアップ サーバが設定されていないため、フェールオーバーは行われません。

例：AAA セッション MIB

次に、AAA セッション MIB 機能を設定して、PPP ユーザの認証済みクライアント接続を解除する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication ppp default group radius
Device(config)# aaa authorization network default group radius
Device(config)# aaa accounting network default start-stop group radius
Device(config)# aaa session-mib disconnect
Device(config)# end
```

アカウンティングの設定に関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*

RFC

RFC	タイトル
RFC 2903	「*Generic AAA Architecture*」
RFC 2904	「*AAA Authorization Framework*」

RFC	タイトル
RFC 2906	「*AAA Authorization Requirements*」
RFC 2989	「*Criteria for Evaluating AAA Protocols for Network Access*」

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカル サポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

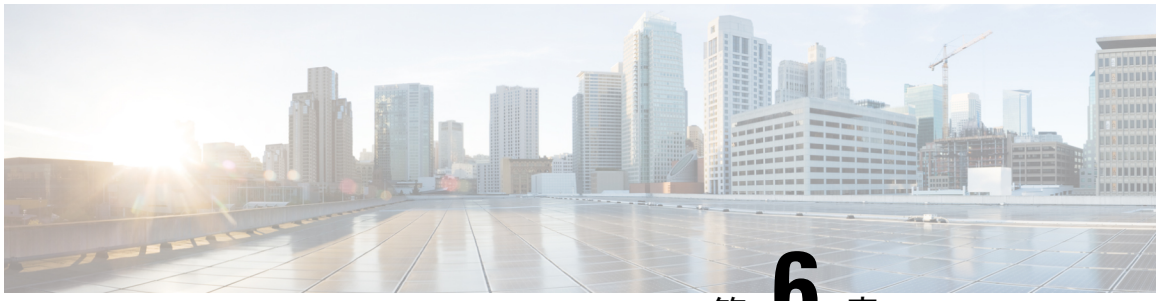
アカウンティングの設定の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	アカウンティング（Accounting）	AAA ブロードキャスト アカウンティングを有効にすると、アカウンティング情報を複数の AAA サーバに同時に送信できます。つまり、アカウンティング情報を 1 つまた複数の AAA サーバに同時にブロードキャストすることが可能です。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 6 章

ローカル認証および許可の設定

スイッチのローカル認証および許可の設定方法

ローカルモードで認証、許可、およびアカウンティング（AAA）を実装するようにスイッチを設定すると、サーバがなくても動作するように AAA を設定できます。この場合、スイッチは認証および許可の処理を行います。この設定ではアカウンティング機能は使用できません。

（注） AAA 方式を使用して HTTP アクセスに対しスイッチのセキュリティを確保するには、グローバル コンフィギュレーション モードで **ip http authentication aaa** コマンドを使用してスイッチを設定する必要があります。AAA 認証を設定しても、AAA 方式を使用した HTTP アクセスに対しスイッチのセキュリティは確保しません。

ローカルモードで AAA を実装するようにスイッチを設定して、サーバがなくても動作するように AAA を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をイネーブルにします。
ステップ 4	**aaa authentication login default local** 例： `Device(config)# aaa authentication login default local`	ローカルユーザ名データベースを使用するログイン認証を設定します。**default** キーワードにより、ローカルユーザデータベース認証がすべてのポートに適用されます。
ステップ 5	**aaa authorization exec default local** 例： `Device(config)# aaa authorization exec default local`	ユーザの AAA 許可を設定し、ローカルデータベースを確認して、そのユーザに EXEC シェルの実行を許可します。
ステップ 6	**aaa authorization network default local** 例： `Device(config)# aaa authorization network default local`	ネットワーク関連のすべてのサービス要求に対してユーザ AAA 許可を設定します。
ステップ 7	**username** *name* [**privilege** *level*] { **password** *encryption-type password*} 例： `Device(config)# username your_user_name privilege 1 password 7 secret567`	ローカルデータベースを入力し、ユーザ名ベースの認証システムを設定します。 ユーザごとにコマンドを繰り返し入力します。 • *name*：ユーザ ID を 1 ワードで指定します。スペースと引用符は使用できません。 • *level*：（任意）ユーザがアクセス権を取得した後に持つ特権レベルを指定します。指定できる範囲は 0 ～ 15 です。レベル 15 では特権 EXEC モードでのアクセスが可能です。レベル 0 では、ユーザ EXEC モードでのアクセスとなります。 • *encryption-type*：暗号化されていないパスワードを指定する場合は 0 を入力します。非表示のパスワードを指定する場合は 7 を入力します。 • *password*：スイッチにアクセスするためにユーザが入力しなければ

	コマンドまたはアクション	目的
		ならないパスワードを指定します。パスワードは1～25文字で、埋め込みスペースを使用でき、**username** コマンドの最後のオプションとして指定します。
ステップ8	**end** 例： `Device(config-line)# end`	特権 EXEC モードに戻ります。
ステップ9	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ10	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

ローカル認証および許可のモニタリング

表 *14 :* ローカル認証および許可を表示するためのコマンド

コマンド	目的
show running-config	ローカル認証および許可の設定を表示します。

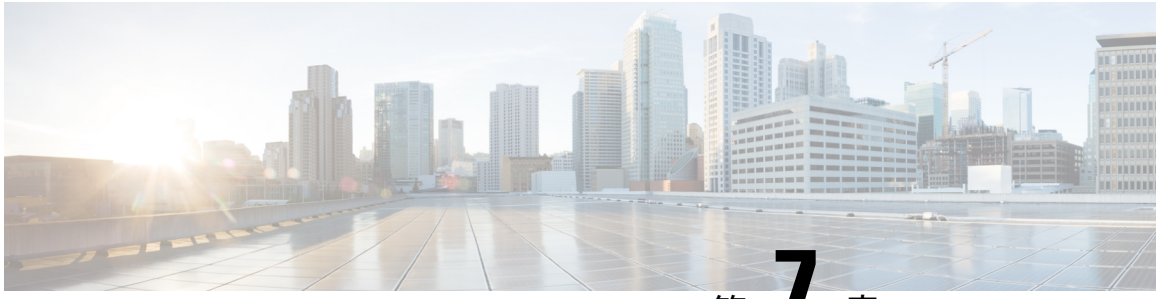
ローカル認証および許可の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	ローカル認証および許可	ローカルモードでAAAを実装するようにデバイスを設定すると、サーバがなくても動作するようにAAAを設定できます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 7 章

MAC 認証バイパス

MAC 認証バイパス機能とは、クライアントの MAC アドレスを使用してネットワークのクライアントを Cisco Identity Based Networking Services（IBNS）およびネットワーク アドミッション コントロール（NAC）の戦略と統合できる、MAC アドレス ベースの認証メカニズムです。MAC 認証バイパス機能は、次のネットワーク環境に適用できます。

- 特定のクライアント プラットフォームにサプリカント コードを使用できないネットワーク環境。
- エンド クライアント設定が管理コントロールを受けていないネットワーク環境、つまり IEEE 802.1X 要求がサポートされていないネットワーク環境。

MAC 認証バイパス設定の前提条件

IEEE 802.1x：ポートベースのネットワーク アクセス コントロール

ポートベースのネットワーク アクセス コントロールの概念とシスコのプラットフォーム上のポートベースのネットワーク アクセス コントロールの設定方法を理解しておく必要があります。

RADIUS および ACL

RADIUS プロトコルの概念とアクセス コントロール リスト（ACL）の作成および適用方法を理解しておく必要があります。詳細については、シスコのプラットフォームのマニュアル、および『Securing User Services Configuration Guide Library』を参照してください。

デバイスが RADIUS 設定されていること、および Cisco Secure アクセス コントロール サーバ（ACS）に接続されていることが必要です。詳細については、『User Guide for Secure ACS Appliance 3.2』を参照してください。

MAC 認証バイパスに関する情報

Cisco IOS Auth Manager の概要

指定されたネットワークに接続するデバイスの機能は異なっている可能性があるため、ネットワークはさまざまな認証方式および許可ポリシーをサポートする必要があります。Cisco IOS Auth Manager は、認証方法に関係なく、ネットワーク認証要求を処理し、許可ポリシーを強制します。Auth Manager は、すべてのポートベースのネットワーク接続試行、認証、許可、および接続解除に対する運用データを維持することで、セッションマネージャとして機能します。

Auth Manager セッションには、次のような状態が考えられます。

- Idle：idle 状態では、認証セッションは初期化されていますが、実行されている方式はありません。これは中間の状態です。
- Running：現在、方式が実行されています。これは中間の状態です。
- Authc Success：認証方式の実行に成功しました。これは中間の状態です。
- Authc Failed：認証方式が失敗しました。これは中間の状態です。
- Authz Success：このセッションに対するすべての機能の適用に成功しました。これは最終的な状態です。
- Authz Failed：このセッションに対して、少なくとも 1 つの機能の適用に失敗しました。これは最終的な状態です。
- 方法なし：このセッションに関する結果はありませんでした。これは最終的な状態です。

設定可能 MAB ユーザ名およびパスワードの概要

MAC 認証バイパス（MAB）動作には、ユーザ名とパスワードの両方の属性を持つ RADIUS Access-Request パケットを使用した認証が含まれます。デフォルトでは、ユーザ名とパスワードの値は同じであり、MAC アドレスを含んでいます。設定可能 MAB ユーザ名およびパスワード機能により、次のシナリオで、ユーザ名とパスワードの両方の属性を設定することができます。

- フォーマットされたユーザ名属性を使用する既存の大規模データベース向けに MAB を有効化するには、クライアント MAC のユーザ名形式を設定する必要があります。ユーザ名形式を設定するには **mab request format attribute 1** コマンドを使用します。
- 一部のデータベースは、ユーザ名とパスワードの値が同じである場合には、認証を受け入れません。そのような場合は、ユーザ名とは確実に異なる値になるようパスワードを設定

する必要があります。パスワードを設定するには **mab request format attribute 2** コマンドを使用します。

設定可能 MAB ユーザ名およびパスワード機能では、Cisco IOS 認証マネージャと既存の MAC データベースおよび RADIUS サーバ間での相互運用が可能です。パスワードはグローバル パスワードなので、すべての MAB 認証およびインターフェイスで共通です。また、このパスワードはすべてのスーパーバイザデバイス間で同期され、それにより高可用性を実現します。

パスワードが提供または設定されていない場合、パスワードはユーザ名と同じ値になります。次の表に、ユーザ名とパスワードの形式を示します。

MAC アドレス	ユーザ名形式（グループのサイズ、区切り記号）	ユーザ名	設定されたパスワード	作成されたパスワード
08002b8619de	（1、:） （1、-） （1、.）	0:8:0:0:2:b:8:6:1:9:d:e 0-8-0-0-2-b-8-6-1-9-d-e 0.8.0.0.2.b.8.6.1.9.d.e	なし	0:8:0:0:2:b:8:6:1:9:d:e 0-8-0-0-2-b-8-6-1-9-d-e 0.8.0.0.2.b.8.6.1.9.d.e
08002b8619de	（1、:） （1、-） （1、.）	0:8:0:0:2:b:8:6:1:9:d:e 0-8-0-0-2-b-8-6-1-9-d-e 0.8.0.0.2.b.8.6.1.9.d.e	Password	Password
08002b8619de	（2、:） （2、-） （2、.）	08:00:2b:86:19:de 08-00-2b-86-19-de 08.00.2b.86.19.de	なし	08:00:2b:86:19:de 08-00-2b-86-19-de 08.00.2b.86.19.de
08002b8619de	（2、:） （2、-） （2、.）	08:00:2b:86:19:de 08-00-2b-86-19-de 08.00.2b.86.19.de	Password	Password
08002b8619de	（4、:） （4、-） （4、.）	0800:2b86:19de 0800-2b86-19de 0800.2b86.19de	なし	0800:2b86:19de 0800-2b86-19de 0800.2b86.19de
08002b8619de	（4、:） （4、-） （4、.）	0800:2b86:19de 0800-2b86-19de 0800.2b86.19de	Password	Password
08002b8619de	（12、<該当なし>）	08002b8619de	なし	08002b8619de

MAC アドレス	ユーザ名形式（グループのサイズ、区切り記号）	ユーザ名	設定されたパスワード	作成されたパスワード
08002b8619de	（12、<該当なし>）	08002b8619de	Password	Password

MAC 認証バイパスの設定方法

MAC 認証バイパスのイネーブル化

802.1X ポートで MAC 認証バイパス機能を有効にするには、次の作業を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type slot / port* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**mab** 例： `Device(config-if)# mab`	MAB をイネーブルにします。
ステップ 5	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 6	**show authentication sessions interface** *type slot* / *port* **details** 例： `Device# show authentication sessions interface gigabitethernet 1/0/1`	インターフェイスの設定と、インターフェイス上のオーセンティケータ インスタンスを表示します。

ポート上の再認証のイネーブル化

デフォルトでは、ポートは自動的に再認証されません。自動再認証をイネーブルにし、再認証の頻度を指定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type slot* / *port* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**switchport** 例： `Device(config-if)# switchport`	インターフェイスをレイヤ 2 スイッチド モードに設定します。
ステップ 5	**switchport mode access** 例： `Device(config-if)# switchport mode access`	インターフェイスのタイプを、非トランキングで非タグ付きのシングル VLAN レイヤ 2 インターフェイスに設定します。
ステップ 6	**authentication port-control auto** 例：	ポートの認証ステートを設定します。

	コマンドまたはアクション	目的
	`Device(config-if)# authentication port-control auto`	
ステップ 7	**mab** [**eap**] 例： `Device(config-if)# mab`	MAB をイネーブルにします。
ステップ 8	**authentication periodic** 例： `Device(config-if)# authentication periodic`	再認証をイネーブルにします。
ステップ 9	**authentication timer reauthenticate** {*seconds* | **server**} 例： `Device(config-if)# authentication timer reauthenticate 900`	再認証の間隔（秒単位）を設定します。
ステップ 10	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

セキュリティ違反モードの指定

ポート上でセキュリティ違反がある場合、ポートをシャットダウンするか、トラフィックを制限できます。デフォルトでは、ポートはシャットダウンされます。ポートをシャットダウンする一定の時間を設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**interface** *type slot* / *port* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**switchport** 例： `Device(config-if)# switchport`	インターフェイスをレイヤ 2 スイッチド モードに設定します。
ステップ 5	**switchport mode access** 例： `Device(config-if)# switchport mode access`	インターフェイスのタイプを、非トランキングで非タグ付きのシングル VLAN レイヤ 2 インターフェイスに設定します。
ステップ 6	**authentication port-control auto** 例： `Device(config-if)# authentication port-control auto`	ポートの認証ステートを設定します。
ステップ 7	**mab** [**eap**] 例： `Device(config-if)# mab`	MAB をイネーブルにします。
ステップ 8	**authentication violation** {**protect** | **replace** | **restrict** | **shutdown**} 例： `Device(config-if)# authentication violation shutdown`	ポート上でセキュリティ違反が生じた場合に取るアクションを設定します。
ステップ 9	**authentication timer restart** *seconds* 例： `Device(config-if)# authentication timer restart 30`	無許可ポートの認証の間隔（秒単位）を設定します。
ステップ 10	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

設定可能 MAB ユーザ名およびパスワードのイネーブル化

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**mab request format attribute 1 groupsize** {**1** | **2** | **4** | **12**} **separator** {**-** | **:** | **.**} [**lowercase** | **uppercase**] 例： `Device(config)# mab request format attribute 1 groupsize 2 separator :`	MAB 要求のユーザ名形式を設定します。
ステップ 4	**mab request format attribute 2** [**0** | **7**] *password* 例： `Device(config)# mab request format attribute 2 password1`	すべての MAB 要求に適用されるグローバル パスワードを設定します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

MAC 認証バイパスの設定例

例：MAC 認証バイパスの設定

次の例では、指定したインターフェイスでMAC認証バイパス（MAB）機能をイネーブルにするために、**mab** コマンドが設定されています。オプションとして、インターフェイス コンフィギュレーションおよびインターフェイス上の認証インスタンスを表示するための **show authentication sessions** コマンドがイネーブル化されています。

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# mab
```

```
Device(config-if)# end
Device# show authentication sessions interface gigabitethernet 1/0/1 details
```

例：設定可能 MAB ユーザ名およびパスワードのイネーブル化

次の例は、MAC 認証バイパス（MAB）のユーザ名形式とパスワードを設定する方法を示しています。この例では、ユーザ名形式は区切り記号のない 12 桁の 16 進数のグループとして設定され、グローバル パスワードは **password1** と設定されます。

```
Device> enable
Device# configure terminal
Device(config)# mab request format attribute 1 groupsize 2 separator :
Device(config)# mab request format attribute 2 password1
Device(config)# end
```

MAC 認証バイパスに関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*

MIB

MIB	MIB のリンク
• CISCO-AUTH-FRAMEWORK-MIB • CISCO-MAC-AUTH-BYPASS-MIB • CISCO-PAE-MIB • IEEE8021-PAE-MIB	選択したプラットフォーム、Cisco IOS ソフトウェア リリース、およびフィーチャセットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	タイトル
RFC 3580	『*IEEE 802.1x Remote Authentication Dial In User Service (RADIUS)*』

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカル サポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

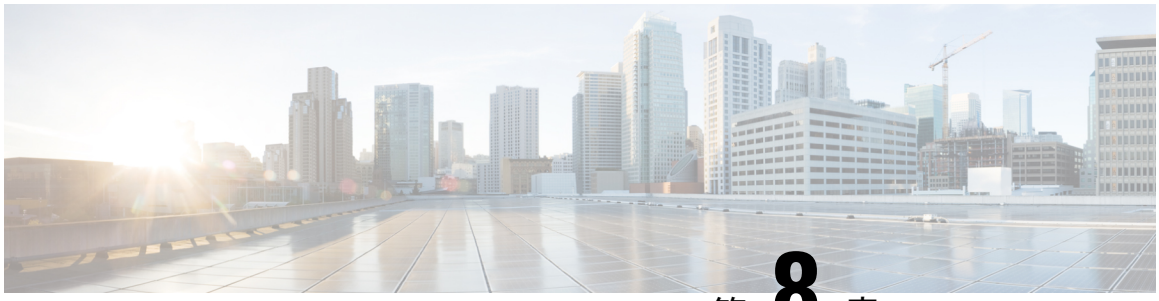
MAC 認証バイパスの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	MAC 認証バイパス	MAC認証バイパス機能とは、クライアントのMACアドレスを使用してネットワークのクライアントを IBNS および NAC の戦略と統合できる、MACアドレスベースの認証メカニズムです。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 8 章

コモン クライテリアに準拠したパスワードの強度と管理

コモン クライテリアに準拠したパスワードの強度と管理機能は、ユーザ パスワードを指定するルールの保存、検索、提供のためのパスワード ポリシーおよびセキュリティ メカニズムを指定するために使用されます。

ローカルユーザについては、ユーザのプロファイルとパスワード情報が重要なパラメータとともにシスコデバイスに保存され、このプロファイルを使用して、ユーザのローカル認証が行われます。このユーザになり得るのは、管理者（ターミナルアクセス）またはネットワークユーザ（たとえば、ネットワーク アクセスのために認証された PPP ユーザ）です。

リモート ユーザについては、ユーザ プロファイル情報がリモート サーバに保存されている場合、管理アクセスとネットワーク アクセスの双方にサードパーティの認証、許可、およびアカウンティング（AAA）サーバを使って AAA サービスが提供される可能性があります。

コモンクライテリアに準拠したパスワードの強度と管理の制約事項

vty を使用して同時にシステムにログインできるユーザは 4 人までです。

コモンクライテリアに準拠したパスワードの強度と管理に関する情報

次の各項では、パスワードの強度と管理について説明します。

パスワード構成ポリシー

パスワード構成ポリシーでは、パスワードを作成するために、英字の大文字小文字、数字、特殊文字（「!」、「@」、「#」、「$」、「%」、「^」、「&」、「*」、「(」、「)」など）を自由に組み合わせて使用できます。

パスワード長ポリシー

パスワードの最小長と最大長は、管理者により柔軟に設定することが可能です。推奨されるパスワードの最小長は 8 文字です。管理者は、パスワードの最小長（1）も最大長（64）も指定できます。

パスワード ライフタイム ポリシー

セキュリティ管理者は、パスワードのライフタイムを最大限にするための設定可能オプションを提供できます。ライフタイムパラメータが設定されていない場合、設定済みのパスワードは無限に有効です。最大ライフタイムは、設定可能な値を年、月、日、時間、分、および秒単位で入力することにより設定できます。ライフタイム設定は設定の一部であるためリロード後も有効ですが、パスワード作成時刻はシステムがリブートするたびに新しい時刻に更新されます。たとえば、パスワードに1ヵ月のライフタイムが設定されており、29日目にシステムがリブートした場合、そのパスワードはシステム リブート後 1 ヵ月間有効になります。

パスワード有効期限ポリシー

ユーザがログインを試みたときにこのユーザのパスワードクレデンシャルが期限切れになっていた場合、次の処理が行われます。

1. ユーザは、期限切れのパスワードの入力に成功した後、新しいパスワードを設定するよう求められます。
2. ユーザが新しいパスワードを入力すると、パスワード セキュリティ ポリシーに照らしてそのパスワードが検証されます。
3. 新しいパスワードがパスワード セキュリティ ポリシーに適合していれば、認証、許可、およびアカウンティング（AAA）データベースが更新され、ユーザは新しいパスワードで認証されます。

4. 新しいパスワードがパスワード セキュリティ ポリシーに適合していない場合、ユーザは再度パスワードの入力を求められます。再試行数は、AAA では制限されていません。認証失敗の場合のパスワード プロンプトの再試行数は、それぞれのターミナル アクセス インタラクティブ モジュールによって制御されます。たとえば Telnet では、3 回失敗するとセッションが終了します。

パスワードのライフタイムを設定されていないユーザがすでにログインしているときに、セキュリティ管理者がそのユーザのライフタイムを設定すると、ライフタイムがデータベースに設定されます。同じユーザが次回に認証されるときに、システムがパスワードの期限を確認します。パスワード期限がチェックされるのは認証フェーズの間のみです。

すでに認証済みかつシステムにログイン中のユーザのパスワードが期限切れになっても、何のアクションも起こりません。同じユーザが次に認証されるときに初めて、ユーザにパスワード変更が求められます。

パスワード変更ポリシー

新しいパスワードは、前のパスワードから 4 文字以上変更されている必要があります。パスワード変更のきっかけとなるシナリオとしては、次のようなものが考えられます。

- セキュリティ管理者がパスワードの変更を求める場合。
- ユーザがプロファイル使用による認証を試みたが、そのプロファイルのパスワードが期限切れになっている場合。

セキュリティ管理者がパスワード セキュリティ ポリシーを変更し、既存のプロファイルがそのパスワード セキュリティ ポリシー ルールに適合しなくなっても、ユーザがすでにシステムにログインしている場合には、何のアクションも起こりません。ユーザは、パスワードセキュリティ制限に適合しないプロファイルを使用して認証を試みたときに初めて、パスワードを変更するよう求められます。

ユーザがパスワードを変更すると、セキュリティ管理者によって古いプロファイルに設定されているライフタイム パラメータが、新しいパスワードのライフタイム パラメータとして引き継がれます。

dot1x などの非インタラクティブ クライアントでは、パスワードの期限が切れると、適切なエラー メッセージがクライアントに送られます。クライアントは、セキュリティ管理者に連絡してパスワードを更新する必要があります。

ユーザ再認証ポリシー

ユーザがパスワードを変更すると、ユーザの再認証が行われます。

期限満了時にパスワードを変更すると、新しいパスワードに対してユーザ認証が行われます。このような場合、実際には、以前のクレデンシャルに基づいて認証が行われ、データベースで新しいパスワードが更新されます。

（注） ユーザがパスワードを変更できるのは、ログイン中かつ古いパスワードの期限が切れた後のみです。ただし、セキュリティ管理者はこのユーザのパスワードをいつでも変更できます。

フレームド（非インタラクティブ）セッションのサポート

dot1x などのクライアントがローカルデータベースを使用して認証を行うときには、コモン クライテリアに準拠したパスワードの強度と管理機能が適用されます。ただし、パスワードの期限が切れると、クライアントによるパスワード変更はできなくなります。そのようなクライアントには適切なエラー メッセージが送られます。そのユーザは、セキュリティ管理者にパスワードの変更を要求する必要があります。

コモンクライテリアに準拠したパスワードの強度と管理の設定方法

次の各項では、パスワードの強度と管理の設定について説明します。

パスワード セキュリティ ポリシーの設定

パスワード セキュリティ ポリシーを作成し、そのポリシーを特定のユーザプロファイルに適用するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をグローバルに有効にします。

	コマンドまたはアクション	目的
ステップ 4	**aaa common-criteria policy** *policy-name* 例： `Device(config)# aaa common-criteria policy policy1`	AAAセキュリティパスワードポリシーを作成し、コモンクライテリア設定ポリシー モードを開始します。
ステップ 5	**char-changes** *number* 例： `Device(config-cc-policy)# char-changes 4`	（任意）古いパスワードから新規のパスワードへの変更文字数を指定します。
ステップ 6	**max-length** *number* 例： `Device(config-cc-policy)# max-length 25`	（任意）パスワードの最大長を指定します。
ステップ 7	**min-length** *number* 例： `Device(config-cc-policy)# min-length 8`	（任意）パスワードの最小長を指定します。
ステップ 8	**numeric-count** *number* 例： `Device(config-cc-policy)# numeric-count 4`	（任意）パスワード内の数字の数を指定します。
ステップ 9	**special-case** *number* 例： `Device(config-cc-policy)# special-case 3`	（任意）パスワード内の特殊文字の数を指定します。
ステップ 10	**exit** 例： `Device(config-cc-policy)# exit`	（任意）コモンクライテリア設定ポリシーモードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 11	**username** *username* **common-criteria-policy** *policy-name* **password** *password*	（任意）ユーザプロファイルに特定のポリシーとパスワードを適用します。

	コマンドまたはアクション	目的

	例：
	`Device(config)# username user1 common-criteria-policy policy1 password password1`
		（注） 単一の数字はパスワードとして使用できません。単一の数字でパスワードを設定しようとすると、次のコンソールメッセージが表示されます。
		`username user2 common-criteria-policy Hay_passwd_policy_2 password 3`
		`% Incomplete command.`

ステップ 12	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

コモン クライテリア ポリシーの確認

すべてのコモンクライテリア セキュリティ ポリシーを確認するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。

ステップ 2	**show aaa common-criteria policy name** *policy-name*
	例：
	`Device# show aaa common-criteria policy name policy1`
	
	`Policy name: policy1`
	`Minimum length: 1`
	`Maximum length: 64`
	`Upper Count: 20`
	`Lower Count: 20`
	`Numeric Count: 5`
	`Special Count: 2`
	`Number of character changes 4`
	`Valid forever. User tied to this policy will not expire.`
		特定のポリシーのパスワード セキュリティ ポリシー情報を表示します。

	コマンドまたはアクション	目的

ステップ 3	**show aaa common-criteria policy all**
	例：
	`Device# show aaa common-criteria policy all`
	`==========`
	`Policy name: policy1`
	`Minimum length: 1`
	`Maximum length: 64`
	`Upper Count: 20`
	`Lower Count: 20`
	`Numeric Count: 5`
	`Special Count: 2`
	`Number of character changes 4`
	`Valid forever. User tied to this policy will not expire.`
	`==========`
	`Policy name: policy2`
	`Minimum length: 1`
	`Maximum length: 34`
	`Upper Count: 10`
	`Lower Count: 5`
	`Numeric Count: 4`
	`Special Count: 2`
	`Number of character changes 2`
	`Valid forever. User tied to this policy will not expire.`
	`==========`
		設定されたすべてのポリシーのパスワード セキュリティ ポリシー情報を表示します。

コモンクライテリアに準拠したパスワードの強度と管理の設定例

次の項では、コモンクライテリアに準拠したパスワードの強度と管理の設定例を示します。

例：コモン クライテリアに準拠したパスワードの強度と管理

次の例は、コモン クライテリア セキュリティ ポリシーを作成し、特定のポリシーをユーザ プロファイルに適用する方法を示しています。

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa common-criteria policy policy1
Device(config-cc-policy)# char-changes 4
Device(config-cc-policy)# max-length 20
Device(config-cc-policy)# min-length 6
Device(config-cc-policy)# numeric-count 2
Device(config-cc-policy)# special-case 2
Device(config-cc-policy)# exit
Device(config)# username user1 common-criteria-policy policy1 password password1
Device(config)# end
```

コモンクライテリアに準拠したパスワードの強度と管理に関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Cisco IOS* リリース *15.2(7)E*（*Catalyst* マイクロスイッチ） 統合プラットフォーム コマンド リファレンス

RFC

RFC	タイトル
RFC 2865	『*Remote Authentication Dial-in User Service*』
RFC 3576	『*Dynamic Authorization Extensions to RADIUS*』

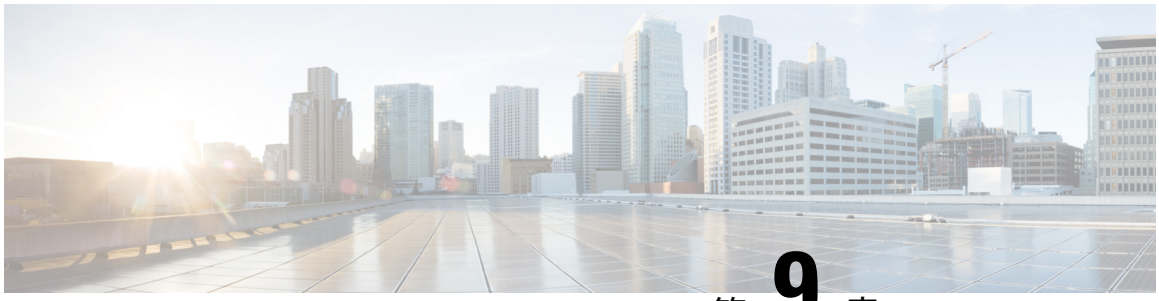
コモンクライテリアに準拠したパスワードの強度と管理の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	コモン クライテリアに準拠したパスワードの強度と管理	コモンクライテリアに準拠したパスワードの強度と管理機能は、ユーザパスワードを指定するルールの保存、検索、提供のためのパスワード ポリシーおよびセキュリティ メカニズムを指定するために使用されます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 9 章

AAA-SERVER-MIB Set Operation

AAA-SERVER-MIB Set Operation 機能により、認証、許可、およびアカウンティング（AAA）サーバ設定を拡張できます。CISCO-AAA-SERVER-MIB を使用して、新規 AAA サーバの作成や追加、CISCO-AAA-SERVER-MIB での「キー」の変更、AAA サーバ設定の削除などを実行できます。

AAA-SERVER-MIB Set Operation の前提条件

AAA がルータで有効になっている必要があります。つまり、**aaa new-model** コマンドが設定されている必要があります。この設定が行われていない場合、SET 操作は失敗します。

AAA-SERVER-MIB Set Operation の制約事項

現時点では、CISCO SNMP SET 操作は RADIUS プロトコルに対してのみサポートされています。このため、追加、修正、削除できるのはグローバル コンフィギュレーション モードの RADIUS サーバだけです。

AAA-SERVER-MIB Set Operation に関する情報

AAA-SERVER-MIB Set Operation 機能により、認証、許可、およびアカウンティング（AAA）サーバ設定を拡張できます。CISCO-AAA-SERVER-MIB を使用して、新規 AAA サーバの作成や追加、CISCO-AAA-SERVER-MIB での「キー」の変更、AAA サーバ設定の削除などを実行できます。

CISCO-AAA-SERVER-MIB

CISCO-AAA-SERVER-MIB により、サーバ自体と AAA サーバの動作、および外部サーバとの AAA 通信の両方の状態が統計情報に反映されます。CISCO-AAA-SERVER-MIB からは次の情報が得られます。

- 各 AAA 動作の統計情報
- AAA 機能を使用できるようになっているサーバのステータス
- 外部 AAA サーバの ID

CISCO-AAA-SERVER-MIB Set Operation

SET 操作を使用すると、次の作業を行うことができます。

- 新しい AAA サーバを作成または追加する。
- CISCO-AAA-SERVER-MIB でキーを修正する。この「秘密キー」は、ネットワーク アクセス サーバ（NAS）および AAA サーバに存在する AAA サーバへの接続をセキュリティ保護するために使用されます。
- AAA サーバの設定を削除する。

Configure AAA-SERVER-MIB Set Operation の設定方法

次の各項では、AAA-SERVER-MIB Set Operation を設定する方法について説明します。

AAA-SERVER-MIB Set Operation の設定

この機能を使用するに当たって、特別な設定は必要ありません。簡易ネットワーク管理プロトコル（SNMP）フレームワークを使用して MIB を管理できます。SNMP の設定については、「その他の参考資料」のセクションを参照してください。

SNMP 値の確認

SNMP 値は次の手順で確認できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを開始します。 プロンプトが表示されたらパスワードを入力します。

	コマンドまたはアクション	目的
ステップ 2	**show running-config aaa** 例： `Device# show running-config aaa`	グローバル コンフィギュレーション モードで設定されているすべての認証、許可、およびアカウンティング（AAA）サーバーを表示します。
ステップ 3	**show aaa servers** 例： `Device# show aaa servers`	認証、許可、およびアカウンティング（AAA）サーバとの間で送受信された要求の数に関するデータを表示します。

AAA-SERVER-MIB Set Operation の設定例

この項では、AAA-SERVER-MIB Set Operation の設定例について説明します。

RADIUS サーバの設定およびサーバの統計情報の例

次の出力例は、SET 操作の前と後の RADIUS サーバの設定およびサーバの統計情報を示しています。

SET 操作の前

```
Device# show aaa servers

RADIUS: id 2, priority 1, host 172.19.192.238, auth-port 2095, acct-port 2096
State: current UP, duration 25s, previous duration 0s
     Dead: total time 0s, count 7
Authen: request 8, timeouts 8
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 2
Author: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Account: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Elapsed time since counters last cleared: 5m
RADIUS: id 3, priority 2, host 172.19.192.238, auth-port 1645, acct-port 1646
State: current UP, duration 5s, previous duration 0s
     Dead: total time 0s, count 2
Authen: request 8, timeouts 8
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 4
Author: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Account: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Elapsed time since counters last cleared: 3m
```

RADIUS サーバの設定と統計情報をチェックする SNMP GET 操作

```
aaa-server5:/users/smetri> getmany 10.0.1.42 casConfigTable
casAddress.2.2 = 172.19.192.238
casAddress.2.3 = 172.19.192.238
casAuthenPort.2.2 = 2095
casAuthenPort.2.3 = 1645
casAcctPort.2.2 = 2096
casAcctPort.2.3 = 1646
casKey.2.2 =
casKey.2.3 =
! The following line shows priority for server 1.
casPriority.2.2 = 1
! The following line shows priority for server 2.
casPriority.2.3 = 2
casConfigRowStatus.2.2 = active(1)
casConfigRowStatus.2.3 = active(1)
aaa-server5:/users/smetri>
```

SNMP SET 操作

RADIUS サーバのキーが変更されています。また、インデックス「1」が使用されています。このインデックスは、エントリの追加、削除、修正に使用されるワイルドカードとして機能します。

```
Change the key for server 1:=>
aaa-server5:/users/smetri>  setany -v2c 10.0.1.42 public casAddress.2.1 -a 172.19.192.238
 casAuthenPort.2.1 -i 2095 casAcctPort.2.1 -i 2096 casKey.2.1 -o king
casAddress.2.1 = 172.19.192.238
casAuthenPort.2.1 = 2095
casAcctPort.2.1 = 2096
casKey.2.1 = king
aaa-server5:/users/smetri>
```

SET 操作の後

上記の SNMP SET 操作後、デバイスの設定が変更されます。SET 操作後の出力を次に示します。

```
Device# show aaa servers

RADIUS: id 3, priority 1, host 172.19.192.238, auth-port 1645, acct-port 1646
State: current UP, duration 189s, previous duration 0s
     Dead: total time 0s, count 2
Authen: request 8, timeouts 8
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 4
Author: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Account: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Elapsed time since counters last cleared: 6m

! The following line shows a new server with new statistics.
RADIUS: id 4, priority 2, host 172.19.192.238, auth-port 2095, acct-port 2096
State: current UP, duration 209s, previous duration 0s
     Dead: total time 0s, count 7
```

```
Authen: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Author: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
     Transaction: success 0, failure 0
Account: request 0, timeouts 0
     Response: unexpected 0, server error 0, incorrect 0, time 0ms
```

AAA-SERVER-MIB Set Operationに関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*

MIB

MIB	MIB のリンク
AAA-SERVER-MIB	選択したプラットフォーム、Cisco IOS XE ソフトウェア リリース、および機能セットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

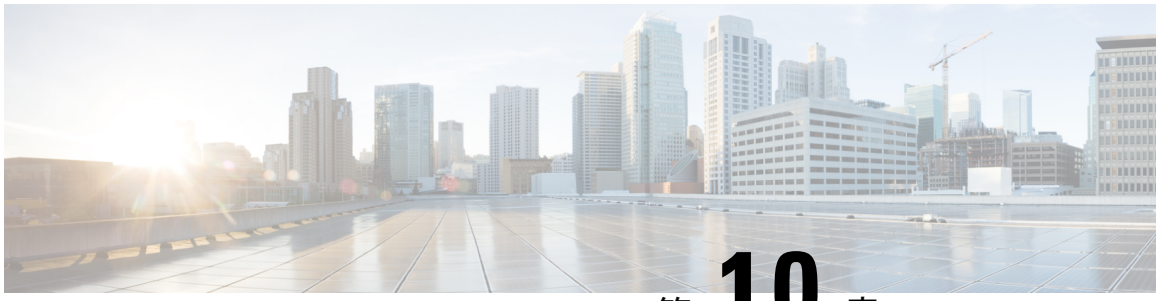
AAA-SERVER-MIB Set Operation の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	AAA-SERVER-MIB Set Operation	AAA-SERVER-MIB Set Operation 機能により、認証、許可、およびアカウンティング（AAA）サーバ設定を拡張できます。CISCO-AAA-SERVER-MIB を使用して、新規 AAA サーバの作成や追加、CISCO-AAA-SERVER-MIB での「キー」の変更、AAA サーバ設定の削除などを実行できます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 10 章

セキュア シェルの設定

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。2 つのバージョンの SSH（SSH バージョン 1 と SSH バージョン 2）を使用できます。

セキュア シェルを設定するための前提条件

セキュア シェル（SSH）用にスイッチを設定するための前提条件は、次のとおりです。

- SSH を動作させるには、スイッチに Rivest、Shamir、および Adleman（RSA）の公開キーと秘密キーのペアが必要です。これは SSH が必要なセキュア コピー プロトコル（SCP）も同様で、セキュアな転送を実現させるには、これらのキーのペアが必要です。
- SCP をイネーブルにする前に、スイッチの SSH、認証、許可、およびアカウンティングを適切に設定してください。
- SCP は SSH を使用してセキュアな転送を実行するため、ルータには RSA キーのペアが必要です。
- SCP はセキュリティについて SSH に依存します。
- SCP の設定には認証、許可、およびアカウンティング（AAA）の許可も必要なため、ルータはユーザが正しい権限レベルを保有しているか確認する必要があります。
- ユーザが SCP を使用するには適切な許可が必要です。

- 適切な許可を得ているユーザは、SCP を使用して Cisco IOS File System（IFS）のファイルをスイッチに（またはスイッチから）自由にコピーできます。コピーには **copy** コマンドを使用します。また、許可されている管理者もこの作業をワークステーションから実行できます。
- セキュア シェル（SSH）サーバは、IPsec（データ暗号規格（DES）または 3DES）の暗号化ソフトウェアイメージを必要とします。SSH クライアントは、IPsec（DES または 3DES）の暗号化ソフトウェア イメージが必要です。
- グローバル コンフィギュレーション モードで **hostname** および **ip domain-name** コマンドを使用して、デバイスのホスト名とホスト ドメインを設定します。

セキュア シェルの設定に関する制約事項

セキュア シェル用にデバイスを設定するための制約事項は、次のとおりです。

- スイッチは、Rivest, Shamir, and Adelman（RSA）認証をサポートします。
- SSH は、実行シェル アプリケーションだけをサポートします。
- SSH サーバおよび SSH クライアントは、データ暗号規格（DES）（56 ビット）および 3DES（168 ビット）データ暗号化ソフトウェアでのみサポートされます。DES ソフトウェア イメージの場合、使用できる暗号化アルゴリズムは DES だけです。3DES ソフトウェア イメージの場合、DES と 3DES の両方の暗号化アルゴリズムを使用できます。
- device は、128 ビット キー、192 ビット キー、または 256 ビット キーの Advanced Encryption Standard（AES）暗号化アルゴリズムをサポートします。ただし、キーを暗号化する対称暗号化 AES はサポートされません。
- SCP を使用する場合、**copy** コマンドにパスワードを入力することはできません。プロンプトが表示されたときに、入力する必要があります。
- ログイン バナーはセキュア シェル バージョン 1 ではサポートされません。セキュア シェル バージョン 2 ではサポートされています。
- リバース SSH の代替手段をコンソール アクセス用に設定する場合、-l キーワード、userid :{number} {ip-address} デリミタ、および引数が必須です。
- FreeRADIUS over RADSEC でクライアントを認証するには、1024 ビットよりも長い RSA キーを生成する必要があります。その場合は、**crypto key generate rsa general-keys exportable label** *label-name* コマンドを使用します。

セキュア シェルの設定について

セキュア シェル（SSH）は、デバイスに対する安全なリモート接続を可能にするプロトコルです。SSH は、デバイスの認証時に強力な暗号化を行うことで、リモート接続について Telnet 以

上のセキュリティを実現します。このソフトウェア リリースは、SSH バージョン 1（SSHv1）および SSH バージョン 2（SSHv2）をサポートしています。

SSH およびスイッチ アクセス

セキュア シェル（SSH）は、デバイスに対する安全なリモート接続を可能にするプロトコルです。SSH は、デバイスの認証時に強力な暗号化を行うことで、リモート接続について Telnet 以上のセキュリティを実現します。このソフトウェアリリースは、SSH バージョン1（SSHv1）および SSH バージョン2（SSHv2）をサポートしています。

IPv6 の SSH 機能は IPv4 における機能と同じです。IPv6 の場合、SSH は IPv6 アドレスをサポートし、IPv6 トランスポート上において、リモート IPv6 ノードとのセキュリティ保護および暗号化された接続を有効化します。

SSH サーバ、統合クライアント、およびサポートされているバージョン

セキュア シェル（SSH）統合クライアント機能は、SSH プロトコル上で動作し、デバイスの認証および暗号化を実現するアプリケーションです。SSH クライアントによって、シスコ デバイスは別のシスコ デバイスなど SSH サーバを実行するデバイスに対して、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、接続が暗号化される点を除いて Telnet のアウトバウンド接続と同様の機能を提供します。SSH クライアントは、認証および暗号化により、保護されていないネットワーク上でもセキュアな通信ができます。

SSH サーバおよび SSH 統合クライアントは、スイッチ上で実行されるアプリケーションです。SSH サーバは、このリリースでサポートされている SSH クライアントおよび、他社製の SSH クライアントと使用します。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、Data Encryption Standard（DES）、3DES、およびパスワード認証の暗号をサポートします。

スイッチは、SSHv1 または SSHv2 サーバをサポートします。

スイッチは、SSHv1 クライアントをサポートします。

（注） SSH クライアント機能を使用できるのは、SSH サーバがイネーブルの場合だけです。

ユーザ認証は、デバイスに対する Telnet セッションの認証と同様に実行されます。SSH は、次のユーザ認証方式もサポートします。

- TACACS+
- RADIUS
- ローカル認証および許可

RSA 認証のサポート

セキュア シェル（SSH）クライアントで使用できる Rivest、Shamir、Adleman（RSA）認証は、シスコソフトウェアの SSH サーバではデフォルトでサポートされていません。

SSL の設定時の注意事項

SSL をスイッチ クラスタで使用すると、SSL セッションがクラスタ コマンダで終了します。クラスタ メンバのスイッチは標準の HTTP で動作させる必要があります。

CA のトラストポイントを設定する前に、システム クロックが設定されていることを確認してください。クロックが設定されていないと、不正な日付により証明書が拒否されます。

Secure Copy Protocol の概要

Secure Copy Protocol（SCP）機能は、スイッチの設定やイメージ ファイルのコピーにセキュアな認証方式を提供します。SCP にはセキュア シェル（SSH）が必要です（Berkeley の r-tool に代わるセキュリティの高いアプリケーションおよびプロトコルです）。

SSH を動作させるには、スイッチに RSA の公開キーと秘密キーのペアが必要です。これは SSH が必要な SCP も同様で、セキュアな転送を実現させるには、これらのキーのペアが必要です。

また、SSH には AAA 許可が必要のため、適切に設定するには、SCP にも AAA 認証が必要になります。

- SCP をイネーブルにする前に、スイッチの SSH、認証、許可、およびアカウンティングを適切に設定してください。
- SCP は SSH を使用してセキュアな転送を実行するため、ルータには RSA キーのペアが必要です。

（注） SCP を使用する場合、**copy** コマンドにパスワードを入力することはできません。プロンプトが表示されたときに、入力する必要があります。

Secure Copy Protocol

Secure Copy Protocol（SCP）機能は、デバイスの設定やスイッチイメージファイルのコピーにセキュアな認証方式を提供します。SCP は一連の Berkeley の r-tools に基づいて設計されているため、その動作内容は、SCP が SSH のセキュリティに対応している点を除けば、Remote Copy Protocol（RCP）と類似しています。また、SCP では認証、許可、およびアカウンティング（AAA）の許可が必要なため、デバイスはユーザが正しい権限レベルを保有しているかどうかを特定できます。セキュア コピー機能を設定するには、SCP の概念を理解する必要があります。

Secure Copy の動作方法

Secure Copy（SCP）は一連の Berkeley の r-tools（Berkeley 大学独自のネットワーキング アプリケーション セット）に基づいて設計されているため、その動作内容は Remote Copy Protocol（RCP）と類似しています。ただし、SCP はセキュア シェル（SSH）のセキュリティに対応している点は除きます。加えて、SCP では、ユーザが正しい権限レベルを持っていることをデバイス上で判断できるように、認証、許可、およびアカウンティング（AAA）の許可を設定する必要があります。

SCP を使用すると、**copy** コマンドを使用して Cisco IOS ファイルシステム（IFS）内の任意のファイルのコピーをデバイスとの間で実行できるのは、特権レベルが 15 のユーザのみになります。許可された管理者はワークステーションからこの操作を実行することもできます。

（注） シスコソフトウェアと一緒に pscp.exe ファイルを使用している場合は、SCP オプションを有効にします。

リバース Telnet

リバース Telnet を使用すると、特定のポート範囲に Telnet を実行したり、端末または補助回線に接続することができます。リバース Telnet は、他のシスコ デバイスのコンソールへの端末回線を複数内蔵したシスコ デバイスとの接続によく使用されていました。Telnet を使用すると、特定の回線上のターミナル サーバに Telnet することによって、どの場所からでも簡単にデバイス コンソールに到達できます。この Telnet アプローチは、デバイスへのすべてのネットワーク接続が切断されている場合でも、そのデバイスの設定に使用できます。また、リバース Telnet は、シスコ デバイスに接続されたモデムをダイヤルアウトに使用することもできます（通常は、ロータリー デバイスと一緒に使用します）。

リバース SSH

リバース Telnet は SSH を使用して実現できます。リバース Telnet と違って、SSH はセキュアな接続を提供します。リバース SSH 拡張機能は、SSH の設定を容易にします。この機能を使用すれば、SSH を有効にする端末または補助回線ごとに別々の回線を設定する必要がなくなります。以前のリバース SSH 設定方法では、アクセスできるポートの数が 100 に制限されていました。リバース SSH 拡張機能では、ポートの数に制限がありません。

セキュア シェルの設定方法

SSH を実行するためのデバイスの設定

SSH を実行するようにデバイスをセットアップするには、次の手順を実行してください。

始める前に

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。この手順は必須です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**hostname** *hostname* 例： `Device(config)# hostname your_hostname`	device のホスト名および IP ドメイン名を設定します。 （注） この手順を実行するのは、device を SSH サーバとして設定する場合だけです。
ステップ 4	**ip domain-name** *domain_name* 例： `Device(config)# ip domain-name your_domain`	device のホストドメインを設定します。
ステップ 5	**crypto key generate rsa** 例： `Device(config)# crypto key generate rsa`	device 上でローカルおよびリモート認証用に SSH サーバをイネーブルにし、RSA キー ペアを生成します。device の RSA キー ペアを生成すると、SSH が自動的にイネーブルになります。 最小モジュラス サイズは、1024 ビットにすることを推奨します。 RSA キーのペアを生成する場合に、モジュラスの長さの入力を求められます。モジュラスが長くなるほど安全ですが、生成と使用に時間がかかります。

	コマンドまたはアクション	目的
		（注） この手順を実行するのは、device を SSH サーバとして設定する場合だけです。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

SSH サーバの設定

SSH サーバを設定するには、次の手順を実行します。

（注） デバイスを SSH サーバとして設定する場合にのみ、この手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip ssh version** [**1** | **2**] 例： `Device(config)# ip ssh version 1`	（任意）SSH バージョン 1 または SSH バージョン 2 を実行するようにデバイスを設定します。 • **1**：SSH バージョン 1 を実行するようにデバイスを設定します。 • **2**：SSH バージョン 2 を実行するようにデバイスを設定します。 このコマンドを入力しない場合、またはキーワードを指定しない場合、SSH サー

	コマンドまたはアクション	目的
		バは、SSH クライアントでサポートされている最新バージョンの SSH を選択します。たとえば、SSH クライアントが SSHv1 および SSHv2 をサポートする場合、SSH サーバは SSHv2 を選択します。
ステップ 4	**ip ssh** {**timeout** *seconds* | **authentication-retries** *number*} 例： `Device(config)# ip ssh timeout 90 authentication-retries 2`	SSH 制御パラメータを設定します。 • タイムアウト値は秒単位で指定します（デフォルト値は 120 秒）。指定できる範囲は 0 ～ 120 秒です。このパラメータは、SSH ネゴシエーション フェーズに適用されます。接続が確立されると、デバイスは CLI ベース セッションのデフォルトのタイムアウト値を使用します。 デフォルトでは、ネットワーク上の複数の CLI ベース セッション（セッション 0 ～ 4）に対して、最大 5 つの暗号化同時 SSH 接続を使用できます。実行シェルが起動すると、CLI ベース セッションのタイムアウト値はデフォルトの 10 分に戻ります。 • クライアントをサーバへ再認証できる回数を指定します。デフォルトは 3 です。指定できる範囲は 0 ～ 5 です。 両方のパラメータを設定する場合はこの手順を繰り返します。
ステップ 5	次のいずれかまたは両方を使用します。 • `line vty` *line_number*[*ending_line_number*] • **transport input ssh** 例： `Device(config)# line vty 1 10` または	（任意）仮想端末回線設定を設定します。 • ライン コンフィギュレーション モードを開始して、仮想端末回線設定を設定します。*line_number* および *ending_line_number* には、回線のペアを指定します。指定できる範囲は 0 ～ 15 です。 • 非 SSH Telnet によるデバイスへの接続を許可しない設定です。これに

	コマンドまたはアクション	目的
	`Device(config-line)# transport input ssh`	より、ルータは SSH 接続に限定されます。
ステップ 6	**end** 例： `Device(config-line)# end`	特権 EXEC モードに戻ります。

トラブルシューティングのヒント

- セキュア シェル（SSH）コンフィギュレーション コマンドが不正なコマンドとして拒否された場合は、デバイスの Rivest、Shamir、Adleman（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 サーバを有効にします。

- RSA キー ペアを設定すると、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 グローバル コンフィギュレーション コマンドを使用して、デバイスのホスト名を設定する必要があります。

  - No domain specified

    ip domain-name グローバル コンフィギュレーション コマンドを使用して、デバイス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、デバイスに設定されている vty の最大数までに制限されます。各 SSH 接続は vty リソースを使用します。

- SSH では、デバイスで 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。認証、許可、およびアカウンティング（AAA）を設定する場合、コンソールで AAA のユーザ認証を無効にする必要があります。デフォルトでコンソールの AAA 認可はディセーブルです。コンソールで AAA 許可が有効になっている場合は、AAA の設定段階で **no aaa authorization console** コマンドを設定して無効にします。

コンソール アクセス用のリバース SSH の設定

SSH サーバ上でリバース SSH コンソール アクセスを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**line** *line-number ending-line-number* 例： `Device# line 1 3`	設定用の回線を特定して、ライン コンフィギュレーションモードに入ります。
ステップ 4	**no exec** 例： `Device(config-line)# no exec`	回線上の EXEC 処理を無効にします。
ステップ 5	**login authentication** *listname* 例： `Device(config-line)# login authentication default`	回線のログイン認証メカニズムを定義します。 （注） 認証方式はユーザ名とパスワードを使用する必要があります。
ステップ 6	**transport input ssh** 例： `Device(config-line)# transport input ssh`	デバイスの特定の回線への接続に使用されるプロトコルを定義します。 • リバース SSH 拡張機能の場合は、**ssh** キーワードを使用する必要があります。
ステップ 7	**exit** 例： `Device(config-line)# exit`	ライン コンフィギュレーション モードを終了します。
ステップ 8	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的
ステップ 9	**ssh -l** *userid* **:** {*number*} {*ip-address*} 例： `Device# ssh -l lab:1 router.example.com`	SSH サーバを実行しているリモートネットワーキング デバイスにログインするときに使用されるユーザ ID を指定します。 • *userid* ：ユーザ ID。 • **:**：ポート番号と端末 IP アドレスが userid 引数に続くことを示します。 • *number*：端末番号または補助回線番号。 • *ip-address* ：ターミナルサーバの IP アドレス。 （注） リバース SSH の代替手段をモデム アクセス用に設定する場合は、*userid* 引数、**:rotary** {*number*} {*ip-address*} デリミタ、および引数が必須です。

モデム アクセス用のリバース SSH の設定

この設定では、リバース SSH がダイヤルアウト回線に使用されるモデム上で設定されます。ダイヤルアウトモデムのいずれかに到達するには、任意の SSH クライアントを使用して SSH セッションを開始し、ロータリーデバイスから次に使用可能なモデムに到達します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**line** *line-number* *ending-line-number* 例： `Device# line 1 200`	設定用の回線を特定して、ラインコンフィギュレーション モードに入ります。
ステップ 4	**no exec** 例： `Device(config-line)# no exec`	回線上の EXEC 処理を無効にします。
ステップ 5	**login authentication** *listname* 例： `Device(config-line)# login authentication default`	回線のログイン認証メカニズムを定義します。 （注） 認証方式はユーザ名とパスワードを使用する必要があります。
ステップ 6	**rotary** *group* 例： `Device(config-line)# rotary 1`	1 つ以上の仮想端末回線または 1 つの補助ポート回線からなる回線グループを定義します。
ステップ 7	**transport input ssh** 例： `Device(config-line)# transport input ssh`	デバイスの特定の回線への接続に使用されるプロトコルを定義します。 • リバース SSH 拡張機能の場合は、**ssh** キーワードを使用する必要があります。
ステップ 8	**exit** 例： `Device(config-line)# exit`	ライン コンフィギュレーション モードを終了します。
ステップ 9	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 10	**ssh -l** *userid* **:rotary** {*number*} {*ip-address*} 例： `Device# ssh -l lab:rotary1 router.example.com`	SSH サーバを実行しているリモート ネットワーキング デバイスにログインするときに使用されるユーザ ID を指定します。 • *userid* ：ユーザ ID。

	コマンドまたはアクション	目的
		• **:** ：ポート番号と端末 IP アドレスが *userid* 引数に続くことを示します。 • *number* ：端末番号または補助回線番号。 • *ip-address* ：ターミナルサーバの IP アドレス。 （注） リバース SSH の代替手段をモデム アクセス用に設定する場合は、*userid* 引数、**:rotary**{*number*}{*ip-address*} デリミタ、および引数が必須です。

クライアント上でのリバース SSH のトラブルシューティング

クライアント（リモート デバイス）上でリバース SSH 設定の問題を解決するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**debug ip ssh client** 例： `Device# debug ip ssh client`	SSH クライアントに関するデバッギング メッセージを表示します。

サーバ上でのリバース SSH のトラブルシューティング

ターミナル サーバ上でリバース SSH 設定の問題を解決するには、次の手順を実行します。各ステップは、互いに独立しているため、任意の順序で設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**debug ip ssh** 例： `Device# debug ip ssh`	SSH サーバに関するデバッギング メッセージを表示します。
ステップ 3	**show ssh** 例： `Device# show ssh`	SSH サーバ接続のステータスを表示します。
ステップ 4	**show line** 例： `Device# show line`	端末回線のパラメータを表示します。

SSH の設定およびステータスのモニタリング

次の表に、SSH サーバの設定およびステータスを示します。

表 *15 : SSH* サーバの設定およびステータスを表示するコマンド

コマンド	目的
show ip ssh	SSHサーバのバージョンおよび設定情報を表示します。
show ssh	SSH サーバのステータスを表示します。

セキュアコピーの設定

シスコ デバイスにSecure Copy（SCP）サーバ側機能の設定をするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa new-model** 例： `Device(config)# aaa new-model`	ログイン時の AAA 認証を設定します。
ステップ 4	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] 例： `Device(config)# aaa authentication login default group tacacs+`	AAA アクセス コントロール システムをイネーブルにします。
ステップ 5	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [*method1* [*method2...*]] 例： `Device(config)# aaa authorization exec default group tacacs+`	ネットワークへのユーザ アクセスを制限するパラメータを設定します。 （注） **exec** キーワードは、許可を実行して、ユーザが EXEC シェルの実行を許可されているかどうかを判断します。したがって、SCP を設定するときに **exec** キーワードを使用する必要があります。
ステップ 6	**username** *name* [**privilege** *level*] **password** *encryption-type encrypted-password* 例： `Device(config)# username superuser privilege 2 password 0 superpassword`	ユーザ名をベースとした認証システムを構築します。 （注） TACACS+ や RADIUS などのネットワークベースの認証メカニズムが設定されている場合は、この手順を省略できます。
ステップ 7	**ip scp server enable** 例：	SCP サーバ側機能を有効にします。

	コマンドまたはアクション	目的
	`Device(config)# ip scp server enable`	
ステップ 8	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 9	**debug ip scp** 例： `Device# debug ip scp`	（任意）SCP 認証問題を解決します。

セキュア シェルの設定例

例：ローカル認証を使用したセキュア コピーの設定

次の例は、Secure Copy（SCP）のサーバ側機能の設定方法を示しています。この例では、ローカルに定義されたユーザ名とパスワードを使用します。

```
! AAA authentication and authorization must be configured properly in order for SCP to
work.
aaa new-model
aaa authentication login default local
aaa authorization exec default local
username user1 privilege 15 password 0 lab
! SSH must be configured and functioning properly.
ip scp server enable
```

例：ネットワークベース認証を使用した SCP サーバ側の設定

次の例は、ネットワークベースの認証メカニズムを使用した SCP のサーバ側機能の設定方法を示しています。

```
! AAA authentication and authorization must be configured properly for SCP to work.
aaa new-model
aaa authentication login default group tacacs+
aaa authorization exec default group tacacs+
! SSH must be configured and functioning properly.
ip ssh time-out 120
ip ssh authentication-retries 3
ip scp server enable
```

リバース SSH コンソール アクセスの例

ターミナル サーバの設定

次の設定例は、リバース SSH が端末回線 1 ～ 3 のコンソール アクセス用に設定されていることを示しています。

```
line 1 3
   no exec
   login authentication default
   transport input ssh
```

クライアント設定

SSHクライアント上で設定された次のコマンドは、それぞれ、回線1、2、および3とのリバース SSH セッションを形成します。

```
ssh -l lab:1 router.example.com
ssh -l lab:2 router.example.com
ssh -l lab:3 router.example.com
```

リバース SSH モデム アクセスの例

次の設定例では、ダイヤルアウト回線の 1 ～ 200 がモデム アクセス用のロータリー グループ 1 にグループ分けされています。

```
line 1 200
   no exec
   login authentication default
   rotary 1
   transport input ssh
   exit
```

次のコマンドは、リバース SSH がロータリー グループの最初の空き回線に接続されることを表示します。

```
ssh -l lab:rotary1 router.example.com
```

例：SSH の設定およびステータスのモニタリング

セキュア シェル（SSH）サーバが有効であることを確認し、SSH接続のバージョンおよび設定データを表示するには、**show ip ssh** コマンドを使用します。次に、SSH がイネーブルの例を示します。

```
Device# show ip ssh

SSH Enabled - version 1.5
Authentication timeout: 120 secs; Authentication retries: 3
```

次に、SSH がディセーブルの例を示します。

```
Device# show ip ssh

%SSH has not been enabled
```

SSH サーバ接続のステータスを確認するには、**show ssh** コマンドを使用します。次に、SSH を有効にしたときのデバイス上の SSH サーバ接続の例を示します。

```
Device# show ssh

Connection      Version     Encryption State Username
 0 1.5 3DES Session Started  guest
```

次に、SSH がディセーブルの例を示します。

```
Device# show ssh

%No SSH server connections running.
```

セキュア シェルに関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

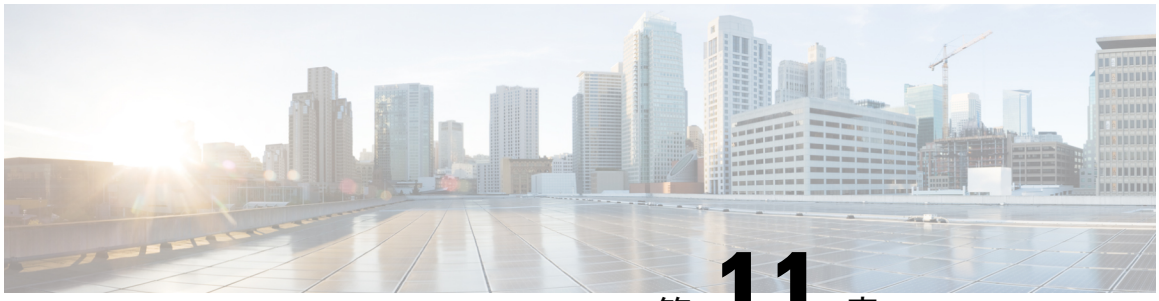
セキュアシェルの設定の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	セキュア シェル	SSHは、デバイスに対する安全なリモート接続を可能にするプロトコルです。SSH は、デバイスの認証時に強力な暗号化を行うことで、リモート接続について Telnet 以上のセキュリティを実現します。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 11 章

セキュア シェル バージョン 2 サポート

セキュア シェル バージョン 2 サポート機能で、セキュア シェル（SSH）バージョン 2 を設定できます（SSH バージョン 1 サポートは、以前のシスコ ソフトウェア リリースに実装されていました）。SSH は、信頼性の高いトランスポート層の上部で実行され、強力な認証機能と暗号化機能を提供します。SSH では、信頼できる転送として定義されているのは TCP のみです。SSH で、ネットワーク上の他のコンピュータに安全にアクセスしたり、コマンドを安全に実行できます。SSH とともに提供されるセキュア コピー プロトコル（SCP）機能で、ファイルを安全に転送できます。

セキュア シェル バージョン 2 サポートに関する情報

SSH バージョン 2

セキュア シェル バージョン 2 サポート機能で、SSH バージョン 2 を設定できます。

SSH バージョン 2 サーバの設定は、SSH バージョン 1 の設定と同様です。**ip ssh version** コマンドは、設定する SSH バージョンを定義します。このコマンドを設定しない場合、デフォルトで SSH は互換モードで実行されます。バージョン 1 とバージョン 2 両方の接続が利用できます。

（注） SSH バージョン 1 は、標準として定義されていないプロトコルです。未定義のプロトコル（バージョン 1）にデバイスがフォールバックしないようにするには、**ip ssh version** コマンドを使用してバージョン 2 を指定する必要があります。

ip ssh rsa keypair-name コマンドを使用すると、設定した Rivest、Shamir、および Adleman（RSA）キーを使用して SSH 接続を実行できます。すでに、SSH は生成済みの最初の RSA キーにリンクされています（つまり、最初の RSA キー ペアが生成された時点で SSH はイネーブルになっています）。この動作は存在していますが、**ip ssh rsa keypair-name** コマンドを使用してこの動作を行わないようにすることができます。 **ip ssh rsa keypair-name** コマンドをキーペアの名前を指定して設定すると、SSH は、キー ペアが存在する場合に有効になるか、キーペアを後で作成する場合は後から有効になります。このコマンドを使用して SSH をイネーブルにする場合、Cisco ソフトウェアの SSH バージョン 1 では必要な、ホスト名とドメイン名を設定を設定する必要はありません。

（注） ログイン バナーは SSH バージョン 2 でサポートされますが、セキュア シェル バージョン 1 ではサポートされません。

セキュア シェル バージョン 2 の RSA キーに関する機能拡張

Cisco SSH バージョン 2 は、キーボード インタラクティブ認証方式およびパスワード ベースの認証方式をサポートしています。RSA キーの SSH バージョン 2 拡張機能は、クライアントとサーバ向けの RSA ベースの公開キー認証もサポートしています。

ユーザ認証：RSA ベースのユーザ認証では、各ユーザに関連付けられている秘密キー/公開キーのペアを認証に使用します。ユーザは秘密キー/公開キーのペアをクライアントで生成し、公開キーを Cisco SSH サーバで設定して、認証を完了します。

クレデンシャルの確立を試行する SSH ユーザは、秘密キーを使用して暗号化された署名を提示します。署名とユーザの公開キーは、認証のために SSH サーバに送信されます。SSH サーバでは、ユーザから提示された公開キーに対してハッシュを計算します。ハッシュは、サーバに一致するエントリがあるかどうかを判断するために使用されます。一致が見つかった場合、RSA ベースのメッセージ検証が公開キーを使用して実行されます。その結果、暗号化されたシグニチャに基づいて、ユーザのアクセスは認証されるか拒否されます。

サーバ認証：SSH セッションの確立中に、Cisco SSH クライアントは、キー交換フェーズ中に使用できるサーバ ホスト キーを使用して、SSH サーバを認証します。SSH サーバ キーは、SSH サーバの識別に使用されます。これらのキーは SSH がイネーブルになるときに作成され、クライアント側で設定する必要があります。

サーバ認証の場合、Cisco SSH クライアントが各サーバにホスト キーを割り当てる必要があります。クライアントがサーバとの間で SSH セッションを確立しようとすると、クライアントはキー交換メッセージの一部として、サーバの署名を受信します。厳密なホスト キーのチェック フラグがクライアント側でイネーブルの場合、そのサーバに対応するホスト キー エントリがあるかどうかがクライアントで確認されます。一致が見つかると、クライアントはサーバ ホスト キーを使用して署名の検証を試行します。サーバの認証に成功すると、セッションの確立処理は続行します。失敗すると、処理は終了し、「Server Authentication Failed」というメッセージが表示されます。

（注） 公開キーをサーバで格納する際、メモリを使用します。したがって、SSHサーバで設定できる公開キーの数は、1 ユーザに最大 2 つの公開キーを作成した場合 10 ユーザ分に限られます。

（注） シスコ サーバはRSA ベースのユーザ認証をサポートしていますが、シスコ クライアントは認証方式として公開キーを提案できません。RSA ベースの認証に対するオープンな SSH クライアントからの要求を Cisco サーバが受信した場合、サーバは認証要求を受け入れます。

（注） サーバ認証の場合、サーバの RSA 公開キーを手動で設定し、Cisco SSH クライアント側で **ip ssh stricthostkeycheck** コマンドを設定します。

SNMP トラップ生成

ご使用のリリースに応じて、簡易ネットワーク管理プロトコル（SNMP）トラップは、トラップが有効で SNMP デバッグがオンになっている場合、SSH セッションが終了した際に自動的に生成されます。SNMP トラップの有効化に関する情報については、『SNMP Configuration Guide』の「Configuring SNMP Support」モジュールを参照してください。

（注） **snmp-server host** コマンドを設定する場合、IP アドレスは、SSH（telnet）クライアントがあり、SSH サーバへの IP 接続が可能な PC のアドレスにする必要があります。

また、**debug snmp packet** コマンドを使用して SNMP デバッグを有効にし、トラップを表示する必要があります。トラップ情報には、送信バイト数や SSH セッションで使用されたプロトコルなどの情報が含まれます。

次の例では、設定済みの SNMP トラップを示します。トラップ通知は、SSH セッションが終了すると自動的に生成されます。この例の a、b、c、d は SSH クライアントの IP アドレスです。

```
snmp-server
snmp-server host a.b.c.d public tty
```

次に、**debug snmp packet** コマンドの出力例を示します。出力には、SSH セッションの SNMP トラップ情報が含まれます。

```
Switch# debug snmp packet

SNMP packet debugging is on
Device1# ssh -l lab 10.0.0.2
Password:
```

```
Switch# exit

[Connection to 10.0.0.2 closed by foreign host]
Device1#
*Jul 18 10:18:42.619: SNMP: Queuing packet to 10.0.0.2
*Jul 18 10:18:42.619: SNMP: V1 Trap, ent cisco, addr 10.0.0.1, gentrap 6, spectrap 1
local.9.3.1.1.2.1 = 6
tcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 4
ltcpConnEntry.5.10.0.0.1.22.10.0.0.2.55246 = 1015
ltcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 1056
ltcpConnEntry.2.10.0.0.1.22.10.0.0.2.55246 = 1392
local.9.2.1.18.2 = lab
*Jul 18 10:18:42.879: SNMP: Packet sent via UDP to 10.0.0.2

Switch#
```

SSH キーボード インタラクティブ認証

SSH キーボード インタラクティブ認証機能は、SSH での汎用メッセージ認証とも呼ばれ、異なる種類の認証メカニズムを実装するために使用できる方式です。基本的に、現在サポートされている、ユーザの入力のみが必要な認証方式はすべて、この機能で実行することができます。この機能は自動的にイネーブルになります。

次の方式がサポートされています。

- Password
- サーバが送信するチャレンジに応答する番号またはストリングを印刷する SecurID およびハードウェア トークン
- プラグイン可能な認証モジュール（PAM）
- S/KEY（およびその他の使い捨てキー）

例：クライアント側のデバッグの有効化

次の例では、クライアント側のデバッグがオンになっており、プロンプトの最大数が 6（SSH キーボード インタラクティブ認証方式のために 3 つ、パスワード認証方式のために 3 つ）になっています。

```
Password:
Password:
Password:
Password:
Password:
Password: cisco123
Last login: Tue Dec 6 13:15:21 2005 from 10.76.248.213
user1@courier:~> exit
logout
[Connection to 10.76.248.200 closed by foreign host]
Device1# debug ip ssh client

SSH Client debugging is on

Device1# ssh -l lab 10.1.1.3
```

```
Password:
*Nov 17 12:50:53.199: SSH0: sent protocol version id SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.199: SSH CLIENT0: protocol version id is - SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.199: SSH CLIENT0: sent protocol version id SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.199: SSH CLIENT0: protocol version exchange successful
*Nov 17 12:50:53.203: SSH0: protocol version id is - SSH-1.99-Cisco-1.25
*Nov 17 12:50:53.335: SSH CLIENT0: key exchange successful and encryption on
*Nov 17 12:50:53.335: SSH2 CLIENT 0: using method keyboard-interactive
Password:
Password:
Password:
*Nov 17 12:51:01.887: SSH2 CLIENT 0: using method password authentication
Password:
Password: lab
Device2>

*Nov 17 12:51:11.407: SSH2 CLIENT 0: SSH2_MSG_USERAUTH_SUCCESS message received
*Nov 17 12:51:11.407: SSH CLIENT0: user authenticated
*Nov 17 12:51:11.407: SSH2 CLIENT 0: pty-req request sent
*Nov 17 12:51:11.411: SSH2 CLIENT 0: shell request sent
*Nov 17 12:51:11.411: SSH CLIENT0: session open
```

例：ブランク パスワードの変更による ChPass の有効化

次の例では、ChPass 機能が有効になっており、SSH キーボード インタラクティブ認証方式を使用してブランク パスワードが変更されています。TACACS+ アクセス コントロール サーバ（ACS）は、バックエンド AAA サーバとして使用されています。

```
Device1# ssh -l cisco 10.1.1.3

Password:
Old Password: cisco
New Password: cisco123
Re-enter New password: cisco123

Device2> exit

[Connection to 10.1.1.3 closed by foreign host]
```

例：ChPass の有効化および初回ログインでのパスワード変更

次の例では、ChPass 機能が有効になっており、TACACS+ ACS はバックエンド サーバとして使用されています。パスワードは、SSH キーボード インタラクティブ 認証方式を使用して最初のログインで変更されています。

```
Device1# ssh -l cisco 10.1.1.3

Password: cisco
Your password has expired.
Enter a new one now.
New Password: cisco123
Re-enter New password: cisco123

Device2> exit

[Connection to 10.1.1.3 closed by foreign host]

Device1# ssh -l cisco 10.1.1.3
```

```
Password:cisco1
Your password has expired.
Enter a new one now.
New Password: cisco
Re-enter New password: cisco12
The New and Re-entered passwords have to be the same.
Try again.
New Password: cisco
Re-enter New password: cisco

Device2>
```

例：ChPass の有効化および 3 回ログインした後のパスワードの失効

次の例では、ChPass 機能が有効になっており、TACACS+ ACS はバックエンド AAA サーバとして使用されています。パスワードは、SSH キーボード インタラクティブ認証方式を使用して 3 回ログインした後に期限切れになります。

```
Device# ssh -l cisco. 10.1.1.3

Password: cisco

Device2> exit

[Connection to 10.1.1.3 closed by foreign host]

Device1# ssh -l cisco 10.1.1.3

Password: cisco

Device2> exit

Device1# ssh -l cisco 10.1.1.3

Password: cisco

Device2> exit

[Connection to 10.1.1.3 closed by foreign host]

Device1# ssh -l cisco 10.1.1.3

Password: cisco
Your password has expired.
Enter a new one now.
New Password: cisco123
Re-enter New password: cisco123

Device2>
```

セキュア シェル バージョン 2 サポートの設定方法

ホスト名およびドメイン名を使用した SSH バージョン 2 のデバイス設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**hostname** *name* 例： `Device(config)# hostname cisco7200`	デバイスのホスト名を設定します。
ステップ 4	**ip domain-name** *name* 例： `cisco7200(config)# ip domain-name example.com`	デバイスのドメイン名を設定します。
ステップ 5	**crypto key generate rsa** 例： `cisco7200(config)# crypto key generate rsa`	ローカルおよびリモート認証用に SSH サーバをイネーブルにします。
ステップ 6	**ip ssh** [**time-out** *seconds* | **authentication-retries** *integer*] 例： `cisco7200(config)# ip ssh time-out 120`	（任意）デバイス上で SSH 制御変数を設定します。
ステップ 7	**ip ssh version** [**1** | **2**] 例：	（任意）デバイスで実行する SSH のバージョンを指定します。

	コマンドまたはアクション	目的
	`cisco7200(config)# ip ssh version 1`	
ステップ 8	**exit** 例： `cisco7200(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。 • デフォルト ホストに戻るには、 **no hostname** コマンドを使用します。

RSA キー ペアを使用した SSH バージョン 2 のデバイス設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip ssh rsa keypair-name** *keypair-name* 例： `Device(config)# ip ssh rsa keypair-name sshkeys`	SSH に使用する RSA キー ペアを指定します。 （注） シスコ デバイスには複数の RSA キー ペアを設定できます。
ステップ 4	**crypto key generate rsa usage-keys label** *key-label* **modulus** *modulus-size* 例： `Device(config)# crypto key generate rsa usage-keys label sshkeys modulus 768`	デバイスでローカルおよびリモート認証を行う SSH サーバを有効にします。 • SSH バージョン 2 では、絶対サイズは 768 ビット以上である必要があります。 （注） 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。RSA キー ペアを削除すると、SSH サーバは自動的に無効になります。

	コマンドまたはアクション	目的
ステップ 5	**ip ssh** [**time-out** *seconds* | **authentication-retries** *integer*] 例： `Device(config)# ip ssh time-out 12`	デバイス上で SSH 制御変数を設定します。
ステップ 6	**ip ssh version 2** 例： `Device(config)# ip ssh version 2`	デバイスで実行する SSH のバージョンを指定します。
ステップ 7	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。

RSA ベースのユーザ認証を実行するための Cisco SSH サーバの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**hostname** *name* 例： `Device(config)# hostname host1`	ホスト名を指定します。
ステップ 4	**ip domain-name** *name* 例： `host1(config)# ip domain-name name1`	Cisco 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 5	**crypto key generate rsa** 例：	RSA キー ペアを生成します。

	コマンドまたはアクション	目的
	`host1(config)# crypto key generate rsa`	
ステップ 6	**ip ssh pubkey-chain** 例： `host1(config)# ip ssh pubkey-chain`	SSH サーバ上のユーザおよびサーバ認証用に SSH-RSA キーを設定し、公開キー コンフィギュレーション モードを開始します。 • サーバに保存されている RSA 公開キーが、クライアントに保存されている公開キーと秘密キーのペアを使用して検証されると、ユーザ認証は成功です。
ステップ 7	**username** *username* 例： `host1(conf-ssh-pubkey)# username user1`	SSH ユーザ名を設定し、公開キー ユーザ コンフィギュレーション モードを開始します。
ステップ 8	**key-string** 例： `host1(conf-ssh-pubkey-user)# key-string`	リモート ピアの RSA 公開キーを指定し、公開キー データ コンフィギュレーション モードを開始します。 （注） オープン SSH クライアントから（言い換えると .ssh/id_rsa.pub ファイルから）公開キー値を取得できます。
ステップ 9	**key-hash** *key-type* *key-name* 例： `host1(conf-ssh-pubkey-data)# key-hash ssh-rsa key1`	（任意）SSH キータイプとバージョンを指定します。 • 秘密キー/公開キー ペアの設定では、キー タイプを ssh-rsa にする必要があります。 • **key-string** コマンドが設定されている場合に限りこの手順は任意です。 • **key-string** コマンドと **key-hash** コマンドのいずれかを設定する必要があります。

	コマンドまたはアクション	目的
		（注） 公開キー ストリングのハッシュを計算するには、ハッシュ処理ソフトウェアを使用します。また、別のシスコデバイスからハッシュ値をコピーすることもできます。初めて公開キーデータを入力する場合、**key-string** コマンドを使用して公開キーデータを入力することを推奨します。
ステップ 10	**end** 例： `host1(conf-ssh-pubkey-data)# end`	公開キーデータ コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。 • デフォルト ホストに戻るには、**no hostname** コマンドを使用します。

RSA ベースのサーバ認証を実行するための Cisco IOS SSH サーバの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**hostname** *name* 例： `Device(config)# hostname host1`	ホスト名を指定します。
ステップ 4	**ip domain-name** *name* 例： `host1(config)# ip domain-name name1`	Cisco 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。

	コマンドまたはアクション	目的
ステップ 5	**crypto key generate rsa** 例： `host1(config)# crypto key generate rsa`	RSA キー ペアを生成します。
ステップ 6	**ip ssh pubkey-chain** 例： `host1(config)# ip ssh pubkey-chain`	SSH サーバ上のユーザおよびサーバ認証用に SSH-RSA キーを設定し、公開キー コンフィギュレーション モードを開始します。
ステップ 7	**server** *server-name* 例： `host1(conf-ssh-pubkey)# server server1`	デバイスでの公開キー認証について SSH サーバを有効にし、公開キー サーバ コンフィギュレーション モードを開始します。
ステップ 8	**key-string** 例： `host1(conf-ssh-pubkey-server)# key-string`	リモート ピアの RSA 公開キーを指定し、公開キー データ コンフィギュレーション モードを開始します。 （注） オープン SSH クライアントから（言い換えると .ssh/id_rsa.pub ファイルから）公開キー値を取得できます。
ステップ 9	**exit** 例： `host1(conf-ssh-pubkey-data)# exit`	公開キー データ コンフィギュレーション モードを終了し、公開キー サーバ コンフィギュレーション モードを開始します。
ステップ 10	**key-hash** *key-type* *key-name* 例： `host1(conf-ssh-pubkey-server)# key-hash ssh-rsa key1`	（任意）SSH キー タイプとバージョンを指定します。 • 秘密キー/公開キー ペアの設定では、キー タイプを ssh-rsa にする必要があります。 • **key-string** コマンドが設定されている場合に限りこの手順は任意です。 • **key-string** コマンドと **key-hash** コマンドのいずれかを設定する必要があります。

	コマンドまたはアクション	目的
		（注） 公開キー ストリングのハッシュを計算するには、ハッシュ処理ソフトウェアを使用します。また、別のシスコデバイスからハッシュ値をコピーすることもできます。初めて公開キーデータを入力する場合、**key-string** コマンドを使用して公開キーデータを入力することを推奨します。
ステップ 11	**end** 例： `host1(conf-ssh-pubkey-server)# end`	公開キー サーバ コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 12	**configure terminal** 例： `host1# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 13	**ip ssh stricthostkeycheck** 例： `host1(config)# ip ssh stricthostkeycheck`	サーバ認証が実行されることを確認します。 • 障害が発生すると、接続は終了します。 • デフォルト ホストに戻るには、**no hostname** コマンドを使用します。

リモート デバイスとの暗号化セッションの開始

（注） 接続するデバイスは、シスコ ソフトウェアでサポートされる暗号化アルゴリズムを備えたセキュア シェル（SSH）サーバをサポートしている必要があります。また、デバイスを有効にする必要はありません。SSH はディセーブル モードで実行できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**ssh** [**-v** {**1** | **2**} | **-c** {**aes128-ctr** | **aes192-ctr** | **aes256-ctr** | **aes128-cbc** | **3des** | **aes192-cbc** | **aes256-cbc**} | **-l** *user-id* | **-l** *user-id*:*vrf-name number ip-address ip-address* | **-l** *user-id*:*rotary number ip-address* | **-m** {**hmac-md5-128** | **hmac-md5-96** | **hmac-sha1-160** | **hmac-sha1-96**} | **-o numberofpasswordprompts** *n* | **-p** *port-num*] {*ip-addr* | *hostname*} [**command** | **-vrf**] 例： `Device# ssh -v 2 -c aes256-ctr -m hmac-sha1-96 -l user2 10.76.82.24`	リモート ネットワーク デバイスとの暗号化されたセッションを開始します。

SSH サーバでの Secure Copy Protocol のイネーブル化

（注） 次のタスクでは、SCP のサーバ側機能を設定します。このタスクは、デバイスでリモートのワークステーションからファイルを安全にコピーできる一般的な設定を示しています。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA アクセス コントロール モデルをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 4	**aaa authentication login default local** 例： `Device(config)# aaa authentication login default local`	認証時にローカルのユーザ名データベースを使用するように、ログイン時の AAA 認証を設定します。
ステップ 5	**aaa authorization exec defaultlocal** 例： `Device(config)# aaa authorization exec default local`	ユーザアクセスを制限するパラメータをネットワークに設定します。許可を実行し、ユーザ ID で EXEC シェルの実行を許可するかどうかを定義します。その後、システムで許可にローカルデータベースを使用する必要があることを指定します。
ステップ 6	**username***name* **privilege** *privilege-level* **password** *password* 例： `Device(config)# username samplename privilege 15 password password1`	ユーザ名ベースの認証システムを確立し、ユーザ名、権限レベル、および非暗号化パスワードを指定します。 （注） *privilege-level* 引数の最小値は 15 です。権限レベルが 15 未満の場合、接続が切断されます。
ステップ 7	**ip ssh time-out***seconds* 例： `Device(config)# ip ssh time-out 120`	デバイスが SSH クライアントの応答を待つ時間間隔を、秒単位で設定します。
ステップ 8	**ip ssh authentication-retries** 整数 例： `Device(config)# ip ssh authentication-retries 3`	インターフェイスのリセット後、認証を試行する回数を設定します。
ステップ 9	**ip scpserverenable** 例： `Device(config)# ip scp server enable`	デバイスで、リモート ワークステーションから安全にファイルをコピーできるようにします。
ステップ 10	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 11	**debug ip scp** 例：	（任意）SCP 認証の問題に関する診断情報を提供します。

	コマンドまたはアクション	目的
	`Device# debug ip scp`	

セキュア シェル接続のステータスの確認

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**show ssh** 例： `Device# show ssh`	SSH サーバ接続のステータスを表示します。
ステップ 3	**exit** 例： `Device# exit`	特権 EXEC モードを終了し、ユーザ EXEC モードに戻ります。

次の **show ssh** コマンドの出力例には、バージョン 1 およびバージョン 2 接続の複数の SSH バージョン 1 およびバージョン 2 接続のステータスが表示されています。

```
---------------------------------------------------------------------
Device# show ssh

Connection      Version Encryption      State                   Username
 0              1.5     3DES            Session started         lab
Connection Version Mode Encryption  Hmac                State
Username
1          2.0     IN   aes128-cbc  hmac-md5     Session started       lab
1          2.0     OUT  aes128-cbc  hmac-md5     Session started       lab
---------------------------------------------------------------------
```

次の **show ssh** コマンドの出力例には、バージョン 2 接続（バージョン 1 接続なし）の複数の SSH バージョン 2 およびバージョン 1 接続のステータスが表示されています。

```
---------------------------------------------------------------------
Device# show ssh

Connection Version Mode Encryption  Hmac                State
Username
1          2.0     IN   aes128-cbc  hmac-md5     Session started       lab
1          2.0     OUT  aes128-cbc  hmac-md5     Session started       lab
```

```
%No SSHv1 server connections running.
-----------------------------------------------------------------------------
```

次の **show ssh** コマンドの出力例には、バージョン 2 接続（バージョン 1 接続なし）の複数の SSH バージョン 1 およびバージョン 2 接続のステータスが表示されています。

```
-----------------------------------------------------------------------------
Device# show ssh

Connection      Version Encryption      State                   Username
 0              1.5     3DES            Session started         lab
%No SSHv2 server connections running.
-----------------------------------------------------------------------------
```

セキュア シェル ステータスの確認

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**show ip ssh** **例：** `Device# show ip ssh`	SSH のバージョンおよび設定データを表示します。
ステップ 3	**exit** **例：** `Device# exit`	特権 EXEC モードを終了し、ユーザ EXEC モードに戻ります。

例

次の **show ip ssh** コマンドの出力例には、有効な SSH のバージョン、認証タイムアウト値、およびバージョン 1 およびバージョン 2 接続の認証の再試行回数が表示されています。

```
-----------------------------------------------------------------------------
Device# show ip ssh

SSH Enabled - version 1.99
Authentication timeout: 120 secs; Authentication retries: 3
-----------------------------------------------------------------------------
```

次の **show ip ssh** コマンドの出力例には、有効な SSH のバージョン、認証タイムアウト値、およびバージョン 2 接続（バージョン 1 接続なし）の認証の再試行回数が表示されています。

```
----------------------------------------------------------------------------
Device# show ip ssh

SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
----------------------------------------------------------------------------
```

次の **show ip ssh** コマンドの出力例には、有効な SSH のバージョン、認証タイムアウト値、およびバージョン 1 接続（バージョン 2 接続なし）の認証の再試行回数が表示されています。

```
----------------------------------------------------------------------------
Device# show ip ssh

3d06h: %SYS-5-CONFIG_I: Configured from console by console
SSH Enabled - version 1.5
Authentication timeout: 120 secs; Authentication retries: 3
----------------------------------------------------------------------------
```

セキュア シェル バージョン 2 のモニタリングと維持

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**debug ip ssh** 例： `Device# debug ip ssh`	SSH のデバッグを有効にします。
ステップ 3	**debug snmp packet** 例： `Device# debug snmp packet`	デバイスによって送受信されたすべての SNMP パケットのデバッグを有効にします。

例

次の **debug ip ssh** コマンドの出力例は、接続が SSH バージョン 2 接続であることを示します。

```
Device# debug ip ssh

00:33:55: SSH1: starting SSH control process
00:33:55: SSH1: sent protocol version id SSH-1.99-Cisco-1.25
00:33:55: SSH1: protocol version id is - SSH-2.0-OpenSSH_2.5.2p2
00:33:55: SSH2 1: send: len 280 (includes padlen 4)
00:33:55: SSH2 1: SSH2_MSG_KEXINIT sent
00:33:55: SSH2 1: ssh_receive: 536 bytes received
00:33:55: SSH2 1: input: packet len 632
00:33:55: SSH2 1: partial packet 8, need 624, maclen 0
00:33:55: SSH2 1: ssh_receive: 96 bytes received
00:33:55: SSH2 1: partial packet 8, need 624, maclen 0
00:33:55: SSH2 1: input: padlen 11
00:33:55: SSH2 1: received packet type 20
00:33:55: SSH2 1: SSH2_MSG_KEXINIT received
00:33:55: SSH2: kex: client->server aes128-cbc hmac-md5 none
00:33:55: SSH2: kex: server->client aes128-cbc hmac-md5 none
00:33:55: SSH2 1: expecting SSH2_MSG_KEXDH_INIT
00:33:55: SSH2 1: ssh_receive: 144 bytes received
00:33:55: SSH2 1: input: packet len 144
00:33:55: SSH2 1: partial packet 8, need 136, maclen 0
00:33:55: SSH2 1: input: padlen 5
00:33:55: SSH2 1: received packet type 30
00:33:55: SSH2 1: SSH2_MSG_KEXDH_INIT received
00:33:55: SSH2 1: signature length 111
00:33:55: SSH2 1: send: len 384 (includes padlen 7)
00:33:55: SSH2: kex_derive_keys complete
00:33:55: SSH2 1: send: len 16 (includes padlen 10)
00:33:55: SSH2 1: newkeys: mode 1
00:33:55: SSH2 1: SSH2_MSG_NEWKEYS sent
00:33:55: SSH2 1: waiting for SSH2_MSG_NEWKEYS
00:33:55: SSH2 1: ssh_receive: 16 bytes received
00:33:55: SSH2 1: input: packet len 16
00:33:55: SSH2 1: partial packet 8, need 8, maclen 0
00:33:55: SSH2 1: input: padlen 10
00:33:55: SSH2 1: newkeys: mode 0
00:33:55: SSH2 1: received packet type 2100:33:55: SSH2 1: SSH2_MSG_NEWKEYS received
00:33:56: SSH2 1: ssh_receive: 48 bytes received
00:33:56: SSH2 1: input: packet len 32
00:33:56: SSH2 1: partial packet 16, need 16, maclen 16
00:33:56: SSH2 1: MAC #3 ok
00:33:56: SSH2 1: input: padlen 10
00:33:56: SSH2 1: received packet type 5
00:33:56: SSH2 1: send: len 32 (includes padlen 10)
00:33:56: SSH2 1: done calc MAC out #3
00:33:56: SSH2 1: ssh_receive: 64 bytes received
00:33:56: SSH2 1: input: packet len 48
00:33:56: SSH2 1: partial packet 16, need 32, maclen 16
00:33:56: SSH2 1: MAC #4 ok
00:33:56: SSH2 1: input: padlen 9
00:33:56: SSH2 1: received packet type 50
00:33:56: SSH2 1: send: len 32 (includes padlen 13)
00:33:56: SSH2 1: done calc MAC out #4
00:34:04: SSH2 1: ssh_receive: 160 bytes received
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #5 ok
00:34:04: SSH2 1: input: padlen 13
00:34:04: SSH2 1: received packet type 50
00:34:04: SSH2 1: send: len 16 (includes padlen 10)
00:34:04: SSH2 1: done calc MAC out #5
00:34:04: SSH2 1: authentication successful for lab
00:34:04: SSH2 1: input: packet len 64
```

```
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #6 ok
00:34:04: SSH2 1: input: padlen 6
00:34:04: SSH2 1: received packet type 2
00:34:04: SSH2 1: ssh_receive: 64 bytes received
00:34:04: SSH2 1: input: packet len 48
00:34:04: SSH2 1: partial packet 16, need 32, maclen 16
00:34:04: SSH2 1: MAC #7 ok
00:34:04: SSH2 1: input: padlen 19
00:34:04: SSH2 1: received packet type 90
00:34:04: SSH2 1: channel open request
00:34:04: SSH2 1: send: len 32 (includes padlen 10)
00:34:04: SSH2 1: done calc MAC out #6
00:34:04: SSH2 1: ssh_receive: 192 bytes received
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #8 ok
00:34:04: SSH2 1: input: padlen 13
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: pty-req request
00:34:04: SSH2 1: setting TTY - requested: height 24, width 80; set: height 24,
width 80
00:34:04: SSH2 1: input: packet len 96
00:34:04: SSH2 1: partial packet 16, need 80, maclen 16
00:34:04: SSH2 1: MAC #9 ok
00:34:04: SSH2 1: input: padlen 11
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: x11-req request
00:34:04: SSH2 1: ssh_receive: 48 bytes received
00:34:04: SSH2 1: input: packet len 32
00:34:04: SSH2 1: partial packet 16, need 16, maclen 16
00:34:04: SSH2 1: MAC #10 ok
00:34:04: SSH2 1: input: padlen 12
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: shell request
00:34:04: SSH2 1: shell message received
00:34:04: SSH2 1: starting shell for vty
00:34:04: SSH2 1: send: len 48 (includes padlen 18)
00:34:04: SSH2 1: done calc MAC out #7
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #11 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #8
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #12 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #9
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #13 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #10
00:34:08: SSH2 1: ssh_receive: 48 bytes received
```

```
00:34:08: SSH2 1: input: packet len 32
00:34:08: SSH2 1: partial packet 16, need 16, maclen 16
00:34:08: SSH2 1: MAC #14 ok
00:34:08: SSH2 1: input: padlen 17
00:34:08: SSH2 1: received packet type 94
00:34:08: SSH2 1: send: len 32 (includes padlen 17)
00:34:08: SSH2 1: done calc MAC out #11
00:34:08: SSH2 1: ssh_receive: 48 bytes received
00:34:08: SSH2 1: input: packet len 32
00:34:08: SSH2 1: partial packet 16, need 16, maclen 16
00:34:08: SSH2 1: MAC #15 ok
00:34:08: SSH2 1: input: padlen 17
00:34:08: SSH2 1: received packet type 94
00:34:08: SSH2 1: send: len 32 (includes padlen 16)
00:34:08: SSH2 1: done calc MAC out #12
00:34:08: SSH2 1: send: len 48 (includes padlen 18)
00:34:08: SSH2 1: done calc MAC out #13
00:34:08: SSH2 1: send: len 16 (includes padlen 6)
00:34:08: SSH2 1: done calc MAC out #14
00:34:08: SSH2 1: send: len 16 (includes padlen 6)
00:34:08: SSH2 1: done calc MAC out #15
00:34:08: SSH1: Session terminated normally
```

セキュア シェル バージョン 2 サポートの設定例

例：セキュア シェル バージョン 2 の設定

```
Device# configure terminal
Device(config)# ip ssh version 2
```

例：リモート デバイスでの暗号化セッションの開始

```
Device# ssh -v 2 -c aes256-cbc -m hmac-sha1-160 -l shaship 10.76.82.24
```

例：サーバ側 SCP の設定

次の例では、SCP のサーバ側機能の設定方法を示します。この例では、デバイスでの AAA 認証および許可も設定しています。この例では、ローカルに定義されたユーザ名とパスワードを使用します。

```
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication login default local
Device(config)# aaa authorization exec default local
Device(config)# username samplename privilege 15 password password1
Device(config)# ip ssh time-out 120
Device(config)# ip ssh authentication-retries 3
Device(config)# ip scp server enable
```

例：SNMP トラップの設定

次の例では、設定済みの SNMP トラップを示します。トラップ通知は、SSH セッションが終了すると自動的に生成されます。この例の a、b、c、d は SSH クライアントの IP アドレスです。

```
snmp-server
snmp-server host a.b.c.d public tty
```

次に、**debug snmp packet** コマンドの出力例を示します。出力には、SSH セッションの SNMP トラップ情報が含まれます。

```
Device1# debug snmp packet

SNMP packet debugging is on
Device1# ssh -l lab 10.0.0.2
Password:

Device2# exit

[Connection to 10.0.0.2 closed by foreign host]
Device1#
*Jul 18 10:18:42.619: SNMP: Queuing packet to 10.0.0.2
*Jul 18 10:18:42.619: SNMP: V1 Trap, ent cisco, addr 10.0.0.1, gentrap 6, spectrap 1
local.9.3.1.1.2.1 = 6
tcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 4
ltcpConnEntry.5.10.0.0.1.22.10.0.0.2.55246 = 1015
ltcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 1056
ltcpConnEntry.2.10.0.0.1.22.10.0.0.2.55246 = 1392
local.9.2.1.18.2 = lab
*Jul 18 10:18:42.879: SNMP: Packet sent via UDP to 10.0.0.2

Device1#
```

例：SSH キーボード インタラクティブ認証

例：SNMP のデバッグ

次に、**debug snmp packet** コマンドの出力例を示します。出力には、SSH セッションの SNMP トラップ情報が含まれます。

```
Device1# debug snmp packet

SNMP packet debugging is on
Device1# ssh -l lab 10.0.0.2
Password:

Device2# exit

[Connection to 10.0.0.2 closed by foreign host]
Device1#
*Jul 18 10:18:42.619: SNMP: Queuing packet to 10.0.0.2
*Jul 18 10:18:42.619: SNMP: V1 Trap, ent cisco, addr 10.0.0.1, gentrap 6, spectrap 1
local.9.3.1.1.2.1 = 6
tcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 4
```

```
ltcpConnEntry.5.10.0.0.1.22.10.0.0.2.55246 = 1015
ltcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 1056
ltcpConnEntry.2.10.0.0.1.22.10.0.0.2.55246 = 1392
local.9.2.1.18.2 = lab
*Jul 18 10:18:42.879: SNMP: Packet sent via UDP to 10.0.0.2

Device1#
```

例：SSH のデバッグの強化

次に、**debug ip ssh detail** コマンドの出力例を示します。出力には、SSH プロトコルとチャネル要求に関するデバッグ情報が含まれます。

```
Device# debug ip ssh detail

00:04:22: SSH0: starting SSH control process
00:04:22: SSH0: sent protocol version id SSH-1.99-Cisco-1.25
00:04:22: SSH0: protocol version id is - SSH-1.99-Cisco-1.25
00:04:22: SSH2 0: SSH2_MSG_KEXINIT sent
00:04:22: SSH2 0: SSH2_MSG_KEXINIT received
00:04:22: SSH2:kex: client->server enc:aes128-cbc mac:hmac-sha1
00:04:22: SSH2:kex: server->client enc:aes128-cbc mac:hmac-sha1
00:04:22: SSH2 0: expecting SSH2_MSG_KEXDH_INIT
00:04:22: SSH2 0: SSH2_MSG_KEXDH_INIT received
00:04:22: SSH2: kex_derive_keys complete
00:04:22: SSH2 0: SSH2_MSG_NEWKEYS sent
00:04:22: SSH2 0: waiting for SSH2_MSG_NEWKEYS
00:04:22: SSH2 0: SSH2_MSG_NEWKEYS received
00:04:24: SSH2 0: authentication successful for lab
00:04:24: SSH2 0: channel open request
00:04:24: SSH2 0: pty-req request
00:04:24: SSH2 0: setting TTY - requested: height 24, width 80; set: height 24, width
80
00:04:24: SSH2 0: shell request
00:04:24: SSH2 0: shell message received
00:04:24: SSH2 0: starting shell for vty
00:04:38: SSH0: Session terminated normally
```

次に、**debug ip ssh packet** コマンドの出力例を示します。出力には、SSH パケットに関するデバッグ情報が含まれます。

```
Device# debug ip ssh packet

00:05:43: SSH2 0: send:packet of  length 280 (length also includes padlen of 4)
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: input: total packet length of 280 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 24 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 272 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 4 bytes
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: input: total packet length of 144 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
```

```
00:05:43: SSH2 0: ssh_receive: 64 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
00:05:43: SSH2 0: ssh_receive: 16 bytes received
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 136 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 6 bytes
00:05:43: SSH2 0: signature length 143
00:05:43: SSH2 0: send:packet of  length 448 (length also includes padlen of 7)
00:05:43: SSH2 0: send:packet of  length 16 (length also includes padlen of 10)
00:05:43: SSH2 0: newkeys: mode 1
00:05:43: SSH2 0: ssh_receive: 16 bytes received
00:05:43: SSH2 0: input: total packet length of 16 bytes
00:05:43: SSH2 0: partial packet length(block size)8 bytes,needed 8 bytes, maclen 0
00:05:43: SSH2 0: input: padlength 10 bytes
00:05:43: SSH2 0: newkeys: mode 0
00:05:43: SSH2 0: ssh_receive: 52 bytes received
00:05:43: SSH2 0: input: total packet length of 32 bytes
00:05:43: SSH2 0: partial packet length(block size)16 bytes,needed 16 bytes, maclen 20
00:05:43: SSH2 0: MAC compared for #3 :ok
```

セキュア シェル バージョン 2 サポートの追加情報

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*

標準

標準	タイトル
IETF Secure Shell Version 2 Draft 規格	Internet Engineering Task Force の Web サイト

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカル サポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

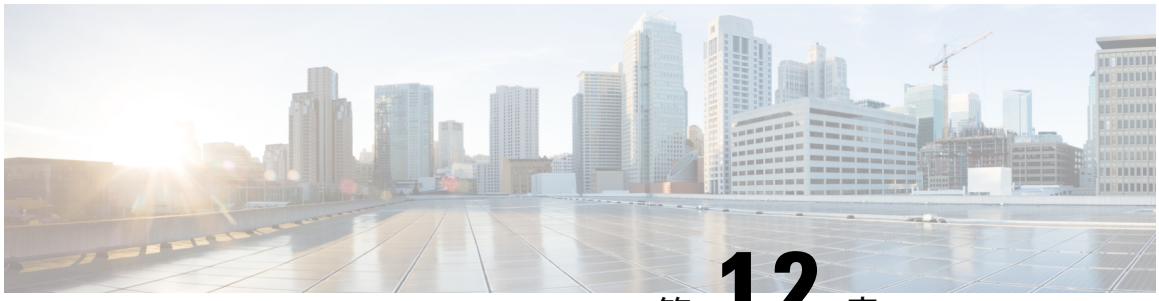
セキュアシェルバージョン 2 サポートの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	セキュア シェル バージョン 2 サポート	セキュア シェル バージョン 2 サポート機能を使用して、セキュア シェル（SSH）バージョン 2 を設定できます（SSH バージョン 1 のサポートは、以前の Cisco IOS ソフトウェア リリースで実装されていました）。SSH は、信頼性の高いトランスポート層の上部で実行され、強力な認証機能と暗号化機能を提供します。SSH バージョン 2 は、AES カウンタベース暗号化モードもサポートします。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 12 章

SSH File Transfer Protocol の設定

セキュアシェル（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 SSH File Transfer Protocol（SFTP）のサポートが含まれています。この機能は、デバイス設定またはデバイスイメージファイルをコピーするための安全で認証された方式を提供します。

SSH File Transfer Protocol の前提条件

- SSH を有効にする必要があります。
- **ip ssh source-interface** *interface-type interface-number* コマンドを設定する必要があります。

SSH File Transfer Protocol の制約事項

- SFTP サーバはサポートされていません。
- SFTP 起動はサポートされていません。
- **sftp** コマンドでの **install add** オプションはサポートされていません。

SSH File Transfer Protocol に関する情報

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、対応するデバイスで常に有効になっています。したがって、適切な権限を持つ SFTP サーバのユーザは、デバイスとの間でファイルをコピーできます。

SFTP クライアントは VRF 対応です。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仮想ルーティングおよび転送（VRF）を使用するようにセキュア FTP クライアントを設定できます。

SSH File Transfer Protocol の設定方法

ここでは、SFTP の設定を構成するさまざまな作業について説明します。

SFTP の設定

次の操作を行ってください。

始める前に

SFTP クライアント側機能用にシスコ デバイスを設定するには、最初に **ip ssh source-interface** *interface-type interface-number* コマンドを設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip ssh source-interface** *interface-type interface-number* 例： `Device(config)# ip ssh source-interface gigabitethernet 1/0/1`	SSH セッションの送信元 IP を定義します。

	コマンドまたはアクション	目的
ステップ 4	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	（任意）SFTP クライアント側機能を表示します。
ステップ 6	**debug ip sftp** 例： `Device# debug ip sftp`	（任意）SFTP デバッグを有効にします。

SFTP コピー操作の実行

ドメインネームシステム（DNS）が設定されている場合、SFTP コピーは対応するサーバの IP またはホスト名を取得します。SFTP コピー操作を実行するには、特権 EXEC モードで次のコマンドを使用します。

コマンド	目的
Device# **copy ios-file-system:file sftp://user:pwd@server-ip//filepath** または Device# **copy ios-file-system: sftp:**	ローカル Cisco IOS ファイルシステムからサーバにファイルをコピーします。 サーバのユーザ名、パスワード、IP アドレス、およびファイルパスを指定します。
Device# **copy sftp://user:pwd@server-ip //filepath ios-file-system:file** または Device# **copy sftp: ios-file-system:**	サーバからローカル Cisco IOS ファイルシステムにファイルをコピーします。 サーバのユーザ名、パスワード、IP アドレス、およびファイルパスを指定します。

例：SSH File Transfer Protocol の設定

次に、SFTP のクライアント側機能を設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# ip ssh source-interface gigabitethernet 1/0/1
Device(config)# exit
```

その他の参考資料

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*
セキュアシェルバージョン 1 と 2 のサポート	セキュア シェルの設定

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカル サポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

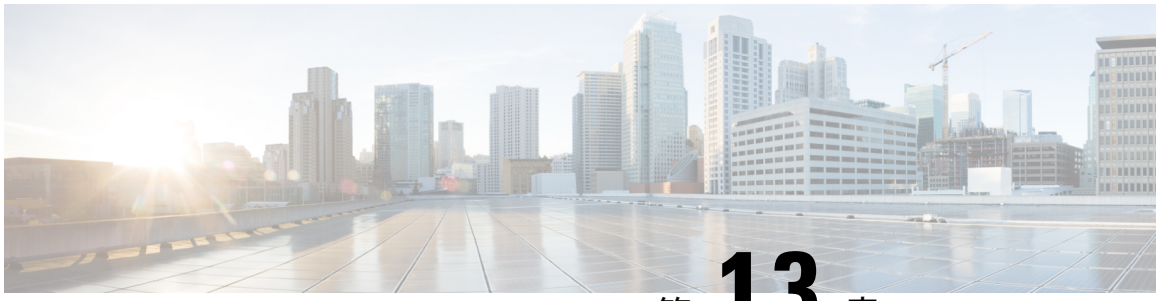
SSH File Transfer Protocol の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	SSH ファイル転送プロトコル	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 SFTP のサポートが含まれています。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 13 章

SSH 認証の X.509v3 証明書

SSH 認証用の X.509v3 証明書機能は、公開キー アルゴリズム（PKI）を使用してサーバおよびユーザの認証を行い、認証局（CA）が署名し発行したデジタル証明書を介してキー ペアの所有者のアイデンティティをセキュア シェル（SSH）プロトコルによって検証することを可能します。

このモジュールでは、デジタル証明書用のサーバおよびユーザ証明書プロファイルを設定する方法について説明します。

SSH 認証の X.509v3 証明書の前提条件

SSH 認証用の X.509v3 証明書機能では、 **ip ssh server algorithm authentication** コマンドの代わりに **ip ssh server authenticate user** コマンドが置き換えられます。**default ip ssh server authenticate user** コマンドを設定し、コンフィギュレーションから **ip ssh server authenticate user** コマンドを削除します。IOS セキュアシェル（SSH）サーバは **ip ssh server algorithm authentication** コマンドを使用して起動します。

ip ssh server authenticate user コマンドを実行すると、次のメッセージが表示されます。

警告

SSH コマンドを受け入れました。ただし、この CLI はまもなく廃止されます。新しい CLI **ip ssh server algorithm authentication**に移動してください。**default ip ssh server authenticate user** を設定し、CLI を無効にします。

SSH 認証の X.509v3 証明書の制約事項

- SSH 認証用の X.509v3 証明書機能の実装は、Cisco IOS セキュア シェル（SSH）サーバ側にのみ適用できます。
- Cisco IOS SSH サーバは、サーバおよびユーザ認証について、x509v3-ssh-rsa アルゴリズム ベースの証明書のみをサポートします。

SSH 認証用の X.509v3 証明書に関する情報

SSH 認証用の X.509v3 証明書の概要

セキュア シェル（SSH）プロトコルは、ネットワーク デバイスへの安全なリモート アクセス接続を提供します。クライアントとサーバの間の通信は暗号化されます。

公開キー暗号化を使用して認証を行う SSH プロトコルが 2 つあります。トランスポート層プロトコルは、デジタル署名アルゴリズム（公開キーアルゴリズムと呼ばれます）を使用して、サーバをクライアントに対して認証します。一方、ユーザ認証プロトコルは、デジタル署名を使用して、クライアントをサーバに対して認証します（公開キー認証）。

認証の有効性は、公開署名キーとその署名者のアイデンティティとの関連の強さに依存します。X.509 バージョン 3（X.509v3）などのデジタル証明書は、アイデンティティ管理のために使用されます。X.509v3 は、信頼できるルート認証局とその中間認証局による署名の連鎖を使用して、公開署名キーを特定のデジタル アイデンティティにバインドします。この実装により、公開キー アルゴリズムを使用したサーバとユーザの認証が可能になるとともに、認証局（CA）が署名し発行したデジタル証明書を介してキー ペアの所有者のアイデンティティを SSH で検証することが可能になります。

X.509v3 を使用したサーバおよびユーザ認証

サーバ認証の場合、セキュア シェル（SSH）サーバが確認のためにそれ自体の証明書を SSH クライアントに送信します。このサーバ証明書は、サーバ証明書プロファイル（ssh-server-cert-profile-server コンフィギュレーション モード）で設定されたトラストポイントに関連付けられます。

ユーザ認証の場合、SSH クライアントが確認のためにユーザの証明書を IOS SSH サーバに送信します。SSH サーバは、サーバ証明書プロファイル（ssh-server-cert-profile-user コンフィギュレーション モード）で設定された公開キー インフラストラクチャ（PKI）トラストポイントを使用して、受信したユーザ証明書を確認します。

デフォルトでは、証明書ベースの認証が、IOS SSH サーバ端末でサーバおよびユーザに対して有効になっています。

OCSP 応答ステープリング

オンライン証明書ステータス プロトコル（OCSP）では、識別された証明書の（失効）状態をアプリケーションが判断することが可能です。このプロトコルは、証明書のステータスをチェックするアプリケーションとそのステータスを提供するサーバとの間でやり取りする必要があるデータを指定します。OCSP クライアントは OCSP レスポンダにステータス要求を発行し、応答を受信するまで証明書の受け入れを保留します。OCSP 応答には、少なくとも、要求の処理ステータスを示す responseStatus フィールドが含まれます。

公開キー アルゴリズムの場合、キーの形式は、1 つ以上の X.509v3 証明書のシーケンスと、その後に続く 0 個以上の OCSP 応答のシーケンスから成ります。

SSH 認証機能向けの X.509v3 証明書は、OCSP 応答ステープリングを使用します。OCSP 応答ステープリングを使用することにより、デバイスは、OCSP サーバにアクセスしてから結果を証明書とともにステープリングして、ピアから OCSP レスポンダにアクセスさせるのではなくピアに情報を送ることで、自身の証明書の失効情報を取得します。

SSH 認証用の X.509v3 証明書の設定方法

サーバ認証用のデジタル証明書の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip ssh server algorithm hostkey** {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]} 例： `Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa`	ホスト キー アルゴリズムの順序を定義します。セキュア シェル（SSH）クライアントとネゴシエートされるのは、設定済みアルゴリズムのみです。

	コマンドまたはアクション	目的
		（注） IOS SSH サーバには、1 つ以上の設定済みホスト キー アルゴリズムが必要です。 • **x509v3-ssh-rsa**：証明書ベースの認証 • **ssh-rsa**：公開キーベースの認証
ステップ 4	**ip ssh server certificate profile** 例： `Device(config)# ip ssh server certificate profile`	サーバ証明書プロファイルおよびユーザ証明書プロファイルを設定し、SSH 証明書プロファイル コンフィギュレーション モードを開始します。
ステップ 5	**server** 例： `Device(ssh-server-cert-profile)# server`	サーバ証明書プロファイルを設定し、SSH サーバ証明書プロファイルのユーザ コンフィギュレーション モードを開始します。 • サーバ プロファイルは、サーバ認証時にサーバ証明書を SSH クライアントに送信するために使用されます。
ステップ 6	**trustpoint sign** *PKI-trustpoint-name* 例： `Device(ssh-server-cert-profile-server)# trustpoint sign trust1`	公開キー インフラストラクチャ（PKI）トラストポイントをサーバ証明書プロファイルにアタッチします。 • SSH サーバは、この PKI トラストポイントに関連付けられた証明書をサーバ認証に使用します。
ステップ 7	**ocsp-response include** 例： `Device(ssh-server-cert-profile-server)# ocsp-response include`	（任意）Online Certificate Status Protocol（OCSP）の応答または OCSP ステープリングをサーバ証明書と一緒に送信します。 （注） デフォルトでは、OCSP 応答はサーバ証明書と一緒には送信されません。

	コマンドまたはアクション	目的
ステップ 8	**end** 例： `Device(ssh-server-cert-profile-server)# end`	SSH サーバ証明書プロファイルのサーバ コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 9	**line vty line_number** [*ending_line_number*] 例： `Device(config)# line vty line_number [ending_line_number]`	ライン コンフィギュレーション モードを開始して、仮想端末回線設定を設定します。line_number および ending_line_number には、回線のペアを指定します。指定できる範囲は 0 ～ 15 です。
ステップ 10	**transport input ssh** 例： `Device(config-line)#transport input ssh`	非 SSH Telnet によるデバイスへの接続を許可しない設定です。これにより、ルータは SSH 接続に限定されます。

ユーザ認証用のデジタル証明書の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip ssh server algorithm authentication** {**publickey** | **keyboard** | **password**} 例： `Device(config)# ip ssh server algorithm authentication publickey`	ユーザ認証アルゴリズムの順序を定義します。セキュア シェル（SSH）クライアントとネゴシエートされるのは、設定済みアルゴリズムのみです。

	コマンドまたはアクション	目的
		（注） • IOS SSH サーバには、1 つ以上の設定済みユーザ認証アルゴリズムが必要です。 • ユーザ認証に証明書方式を使用するには、**publickey** キーワードを設定する必要があります。
ステップ 4	**ip ssh server algorithm publickey** {**x509v3-ssh-rsa** [**ssh-rsa**] | **ssh-rsa** [**x509v3-ssh-rsa**]} 例： `Device(config)# ip ssh server algorithm publickey x509v3-ssh-rsa`	公開キーアルゴリズムの順序を定義します。SSH クライアントによってユーザ認証に許可されるのは、設定済みのアルゴリズムのみです。 （注） IOS SSH クライアントには、1 つ以上の設定済み公開キーアルゴリズムが必要です。 • **x509v3-ssh-rsa**：証明書ベースの認証 • **ssh-rsa**：公開キーベースの認証
ステップ 5	**ip ssh server certificate profile** 例： `Device(config)# ip ssh server certificate profile`	サーバ証明書プロファイルおよびユーザ証明書プロファイルを設定し、SSH 証明書プロファイルコンフィギュレーション モードを開始します。
ステップ 6	**user** 例： `Device(ssh-server-cert-profile)# user`	ユーザ証明書プロファイルを設定し、SSH サーバ証明書プロファイルのユーザ コンフィギュレーション モードを開始します。
ステップ 7	**trustpoint verify** *PKI-trustpoint-name* 例： `Device(ssh-server-cert-profile-user)# trustpoint verify trust2`	受信したユーザ証明書の確認に使用される公開キー インフラストラクチャ（PKI）トラストポイントを設定します。

	コマンドまたはアクション	目的
		（注） 同じコマンドを複数回実行することで、複数のトラストポイントを設定します。最大10のトラストポイントを設定できます。
ステップ 8	**ocsp-response required** 例： `Device(ssh-server-cert-profile-user)# ocsp-response required`	（任意）受信したユーザ証明書による Online Certificate Status Protocol（OCSP）の応答の有無を要求します。 （注） デフォルトでは、ユーザ証明書はOCSP応答なしで受け入れられます。
ステップ 9	**end** 例： `Device(ssh-server-cert-profile-user)# end`	SSH サーバ証明書プロファイルのユーザ コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 10	**line vty line_number** [*ending_line_number*] 例： `Device(config)# line vty line_number [ending_line_number]`	ライン コンフィギュレーション モードを開始して、仮想端末回線設定を設定します。line_number および ending_line_number には、回線のペアを指定します。指定できる範囲は 0 ～ 15 です。
ステップ 11	**transport input ssh** 例： `Device(config-line)#transport input ssh`	非 SSH Telnet によるデバイスへの接続を許可しない設定です。これにより、ルータは SSH 接続に限定されます。

デジタル証明書を使用したサーバおよびユーザ認証の確認

手順

ステップ 1 **enable**

特権 EXEC モードを有効にします。

- パスワードを入力します（要求された場合）。

例：

```
Device> enable
```

ステップ 2 show ip ssh

現在設定されている認証方式を表示します。証明書ベース認証の使用を確認するには、x509v3-ssh-rsa アルゴリズムが設定済みのホスト キー アルゴリズムであることを確認します。

例：

```
Device# show ip ssh

SSH Enabled - version 1.99
Authentication methods:publickey,keyboard-interactive,password
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa
Hostkey Algorithms:x509v3-ssh-rsa,ssh-rsa
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
```

ステップ 3 debug ip ssh detail

SSH 詳細のデバッグメッセージをオンにします。

例：

```
Device# debug ip ssh detail

ssh detail messages debugging is on
```

ステップ 4 show log

デバッグメッセージログを表示します。

例：

```
Device# show log

Syslog logging: enabled (0 messages dropped, 9 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)

No Active Message Discriminator.



No Inactive Message Discriminator.


    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 233 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    File logging: disabled
    Persistent logging: disabled

No active filter modules.

    Trap logging: level informational, 174 message lines logged
```

```
        Logging Source-Interface:       VRF Name:

Log Buffer (4096 bytes):
5 IST: SSH2 CLIENT 0: SSH2_MSG_KEXINIT sent
*Sep  6 14:44:08.496 IST: SSH0: protocol version id is - SSH-1.99-Cisco-1.25
*Sep  6 14:44:08.496 IST: SSH2 0: kexinit sent: kex algo =
diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1
*Sep  6 14:44:08.496 IST: SSH2 0: Server certificate trustpoint not found. Skipping
hostkey algo = x509v3-ssh-rsa
*Sep  6 14:44:08.496 IST: SSH2 0: kexinit sent: hostkey algo = ssh-rsa
*Sep  6 14:44:08.496 IST: SSH2 0: kexinit sent: encryption algo =
aes128-ctr,aes192-ctr,aes256-ctr
*Sep  6 14:44:08.496 IST: SSH2 0: kexinit sent: mac algo =
hmac-sha2-256,hmac-sha2-512,hmac-sha1,hmac-sha1-96
*Sep  6 14:44:08.496 IST: SSH2 0: SSH2_MSG_KEXINIT sent
*Sep  6 14:44:08.496 IST: SSH2 0: SSH2_MSG_KEXINIT received
*Sep  6 14:44:08.496 IST: SSH2 0: kex: client->server enc:aes128-ctr mac:hmac-sha2-256
*Sep  6 14:44:08.496 IST: SSH2 0: kex: server->client enc:aes128-ctr mac:hmac-sha2-256
*Sep  6 14:44:08.496 IST: SSH2 0: Using hostkey algo = ssh-rsa
*Sep  6 14:44:08.496 IST: SSH2 0: Using kex_algo = diffie-hellman-group-exchange-sha1
*Sep  6 14:44:08.497 IST: SSH2 CLIENT 0: SSH2_MSG_KEXINIT received
*Sep  6 14:44:08.497 IST: SSH2 CLIENT 0: kex: server->client enc:aes128-ctr
mac:hmac-sha2-256
*Sep  6 14:44:08.497 IST: SSH2 CLIENT 0: kex: client->server enc:aes128-ctr
mac:hmac-sha2-256
*Sep  6 14:44:08.497 IST: SSH2 CLIENT 0: Using hostkey algo = ssh-rsa
*Sep  6 14:44:08.497 IST: SSH2 CLIENT 0: Using kex_algo =
diffie-hellman-group-exchange-sha1
*Sep  6 14:44:08.497 IST: SSH2 CLIENT 0: SSH2_MSG_KEX_DH_GEX_REQUEST sent
*Sep  6 14:44:08.497 IST: SSH2 CLIENT 0: Range sent- 2048  < 2048  < 4096
*Sep  6 14:44:08.497 IST: SSH2 0: SSH2_MSG_KEX_DH_GEX_REQUEST received
*Sep  6 14:44:08.497 IST: SSH2 0: Range sent by client is - 2048 < 2048 < 4096
*Sep  6 14:44:08.497 IST: SSH2 0:  Modulus size established : 2048 bits
*Sep  6 14:44:08.510 IST: SSH2 0: expecting SSH2_MSG_KEX_DH_GEX_INIT
*Sep  6 14:44:08.510 IST: SSH2 CLIENT 0: SSH2_MSG_KEX_DH_GEX_GROUP received
*Sep  6 14:44:08.510 IST: SSH2 CLIENT 0: Server has chosen 2048 -bit dh keys
*Sep  6 14:44:08.523 IST: SSH2 CLIENT 0: expecting SSH2_MSG_KEX_DH_GEX_REPLY
*Sep  6 14:44:08.524 IST: SSH2 0: SSH2_MSG_KEXDH_INIT received
*Sep  6 14:44:08.555 IST: SSH2: kex_derive_keys complete
*Sep  6 14:44:08.555 IST: SSH2 0: SSH2_MSG_NEWKEYS sent
*Sep  6 14:44:08.555 IST: SSH2 0: waiting for SSH2_MSG_NEWKEYS
*Sep  6 14:44:08.555 IST: SSH2 CLIENT 0: SSH2_MSG_KEX_DH_GEX_REPLY received
*Sep  6 14:44:08.555 IST: SSH2 CLIENT 0: Skipping ServerHostKey Validation
*Sep  6 14:44:08.571 IST: SSH2 CLIENT 0: signature length 271
*Sep  6 14:44:08.571 IST: SSH2: kex_derive_keys complete
*Sep  6 14:44:08.571 IST: SSH2 CLIENT 0: SSH2_MSG_NEWKEYS sent
*Sep  6 14:44:08.571 IST: SSH2 CLIENT 0: waiting for SSH2_MSG_NEWKEYS
*Sep  6 14:44:08.571 IST: SSH2 CLIENT 0: SSH2_MSG_NEWKEYS received
*Sep  6 14:44:08.571 IST: SSH2 0: SSH2_MSG_NEWKEYS received
*Sep  6 14:44:08.571 IST: SSH2 0: Authentications that can continue =
publickey,keyboard-interactive,password
*Sep  6 14:44:08.572 IST: SSH2 0: Using method = none
*Sep  6 14:44:08.572 IST: SSH2 0: Authentications that can continue =
publickey,keyboard-interactive,password
*Sep  6 14:44:08.572 IST: SSH2 0: Using method = keyboard-interactive
*Sep  6 14:44:11.983 IST: SSH2 0: authentication successful for cisco
*Sep  6 14:44:11.984 IST: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: cisco] [Source:
 192.168.121.40] [localport: 22] at 14:44:11 IST Thu Sep 6 2018
*Sep  6 14:44:11.984 IST: SSH2 0: channel open request
*Sep  6 14:44:11.985 IST: SSH2 0: pty-req request
*Sep  6 14:44:11.985 IST: SSH2 0: setting TTY - requested: height 24, width 80; set:
height 24, width 80
*Sep  6 14:44:11.985 IST: SSH2 0: shell request
*Sep  6 14:44:11.985 IST: SSH2 0: shell message received
```

```
*Sep  6 14:44:11.985 IST: SSH2 0: starting shell for vty
*Sep  6 14:44:22.066 IST: %SYS-6-LOGOUT: User cisco has exited tty session
1(192.168.121.40)
*Sep  6 14:44:22.166 IST: SSH0: Session terminated normally
*Sep  6 14:44:22.167 IST: SSH CLIENT0: Session terminated normally
```

ステップ 5 debug ip packet

IP パケット詳細のデバッグをオンにします。

例：

```
Device# debug ip packet
```

ステップ 6 show log

デバッグメッセージログを表示します。

例：

```
Device# show log

yslog logging: enabled (0 messages dropped, 9 messages rate-limited, 0 flushes, 0 overruns,
 xml disabled, filtering disabled)

No Active Message Discriminator.



No Inactive Message Discriminator.


    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 1363 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    File logging: disabled
    Persistent logging: disabled

No active filter modules.

    Trap logging: level informational, 176 message lines logged
        Logging Source-Interface:       VRF Name:

Log Buffer (4096 bytes):
bleid=0, s=192.168.121.40 (local), d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed
 via RIB
*Sep  6 14:45:45.177 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, sending
*Sep  6 14:45:45.177 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, output feature, NAT Inside(8), rtype 1, forus FALSE,
 sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.177 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.177 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.177 IST: IP: s=192.168.121.40 (local), d=192.168.121.40, len 40, local
 feature, feature skipped, NAT(2), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk
FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), routed via RIB
```

```
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, sending
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, output feature, NAT Inside(8), rtype 1, forus FALSE,
 sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40, len 40, local
 feature, feature skipped, NAT(2), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk
FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, sending
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, output feature, NAT Inside(8), rtype 1, forus FALSE,
 sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40, len 40, local
 feature, feature skipped, NAT(2), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk
FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, sending
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, output feature, NAT Inside(8), rtype 1, forus FALSE,
 sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.179 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.179 IST: IP: s=192.168.121.40 (local), d=192.168.121.40, len 40, local
 feature, feature skipped, NAT(2), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk
FALSE
*Sep  6 14:45:45.179 IST: IP: tableid=0, s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.179 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, sending
*Sep  6 14:45:45.179 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, output feature, NAT Inside(8), rtype 1, forus FALSE,
 sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.179 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
```

SSH 認証用の X.509v3 証明書の設定例

例：サーバ認証用のデジタル証明書の設定

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa
```

```
Device(config)# ip ssh server certificate profile
Device(ssh-server-cert-profile)# server
Device(ssh-server-cert-profile-server)# trustpoint sign trust1
Device(ssh-server-cert-profile-server)# exit
```

例：ユーザ認証用のデジタル証明書の設定

```
Device> enable
Device# configure terminal
Device(config)# ip ssh server algorithm authentication publickey
Device(config)# ip ssh server algorithm publickey x509v3-ssh-rsa
Device(config)# ip ssh server certificate profile
Device(ssh-server-cert-profile)# user
Device(ssh-server-cert-profile-user)# trustpoint verify trust2
Device(ssh-server-cert-profile-user)# end
```

SSH 認証用の X.509v3 証明書に関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Cisco IOS* リリース *15.2(7)E*（*Catalyst* マイクロスイッチ）統合プラットフォーム コマンド リファレンス
PKI 設定	PKI 展開での Cisco IOS 証明書サーバの設定および管理

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

SSH 認証用の X.509v3 証明書の機能履歴

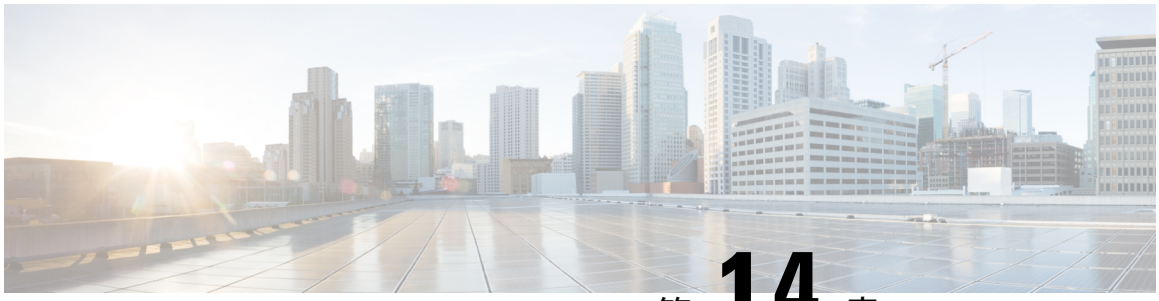
次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	SSH 認証の X.509v3 証明書	SSH認証のX.509v3証明書機能は、サーバ内でX.509v3デジタル証明書を使用し、SSHサーバ側でユーザ認証を使用します。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。

SSH 認証用の X.509v3 証明書の機能履歴



第 14 章

Secure Socket Layer HTTP の設定

この機能は、Cisco IOS ソフトウェアでの HTTP 1.1 サーバおよび HTTP 1.1 クライアントに対する Secure Socket Layer（SSL）バージョン 3.0 のサポートを提供します。SSL は、サーバ認証、暗号化、メッセージ整合性を提供し、セキュリティ保護された HTTP 通信を実現します。SSL は、HTTP クライアント認証も実現します。HTTP over SSL は HTTPS と略されます。

Secure Socket Layer HTTP に関する情報

セキュア HTTP サーバおよびクライアントの概要

セキュア HTTP 接続の場合、HTTP サーバが送受信するデータは暗号化されてインターネットに送信されます。SSL 暗号化を伴う HTTP は、Web ブラウザからスイッチを設定するような機能に、セキュアな接続を提供します。シスコが実装するセキュア HTTP サーバおよび HTTP クライアントでは、アプリケーション層の暗号化に SSL バージョン 3.0 を使用します。HTTP over SSL は、HTTPS と省略されます（セキュアな接続の場合、URL が http://の代わりに https://で始まります）。

（注） SSL は 1999 年に Transport Layer Security（TLS）に発展しましたが、このような特定のコンテキストでまだ使用されています。

セキュア HTTP サーバ（スイッチ）の主な役割は、指定のポート（デフォルトの HTTPS ポートは 443）で HTTPS 要求を待ち受けて、HTTP 1.1 Web サーバへその要求を渡すことです。

HTTP 1.1 サーバはその要求を処理して、セキュア HTTP サーバへ応答（呼び出す）します。セキュア HTTP サーバは HTTP 1.1 サーバの代わりに、元の要求に応えます。

セキュア HTTP クライアント（Web ブラウザ）の主な役割は、Cisco IOS アプリケーション要求に応答して、そのアプリケーションが要求した HTTPS User Agent サービスを実行し、応答を（そのアプリケーションに）返すことです。

CA のトラストポイント

認証局（CA）は、要求を認可して参加するネットワーク デバイスに証明書を発行します。これらのサービスは、参加するデバイスに対する中央集中的なセキュリティ キーおよび証明書の管理を提供します。特定の CA サーバはトラストポイントと呼ばれます。

接続が実行されると、HTTPS サーバは、トラストポイントとなる特定の CA から得た X.509v3 の証明書を発行することで、セキュアな接続をクライアントに提供します。クライアント（通常、Web ブラウザ）は、その証明書の認証に必要な公開キーを保有しています。

セキュア HTTP 接続には、CA のトラストポイントを設定することを強く推奨します。HTTPS サーバを実行しているデバイスに CA のトラストポイントが設定されていないと、サーバは自身を認証して必要な RSA のキーのペアを生成します。自身で認証した（自己署名）証明書は適切なセキュリティではないので、接続するクライアントはその証明書が自己証明書であることを通知し、ユーザに接続の選択（確立または拒否）をさせる必要があります。この選択肢は内部ネットワーク トポロジ（テスト用など）に役立ちます。

CA のトラストポイントを設定していないと、セキュア HTTP 接続を有効にした場合、そのセキュア HTTP サーバ（またはクライアント）に対する一時的または永続的な自己署名証明書が自動的に生成されます。

- デバイスにホスト名とドメイン名が設定されてない場合、一時的な自己署名証明書が生成されます。デバイスが再起動すると、一時的な自己署名証明書がすべて失われて、新しい一時的な自己署名証明書が割り当てられます。
- デバイスにホスト名とドメイン名が設定されている場合、永続的な自己署名証明書が生成されます。デバイスを再起動したり、セキュア HTTP サーバをディセーブルにしたりする場合も、この証明書はアクティブなままで残り、再度セキュア HTTP 接続をイネーブルにする際にも存在しています。

（注） 認証局およびトラストポイントは、個々のデバイスで設定する必要があります。他のデバイスからコピーすると、それらはデバイス上で無効になります。

新しい証明書を登録した場合、新しい設定の変更は、サーバが再起動するまで HTTPS サーバに適用されません。CLI を使用するか、または物理的な再起動によって、サーバを再起動できます。サーバを再起動すると、デバイスは新しい証明書の使用を開始します。

自己署名証明書が生成された場合、その情報は **show running-config** 特権 EXEC コマンドで出力できます。自己署名証明書を表示するコマンドの出力（show running-config コマンド）を例として一部示します。

```
Device# show running-config
Building configuration...

<output truncated>

crypto pki trustpoint TP-self-signed-3080755072
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-3080755072
 revocation-check none
 rsakeypair TP-self-signed-3080755072
!
!
crypto ca certificate chain TP-self-signed-3080755072
 certificate self-signed 01
  3082029F 30820208 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  59312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 33303830 37353530 37323126 30240609 2A864886 F70D0109
  02161743 45322D33 3535302D 31332E73 756D6D30 342D3335 3530301E 170D3933
  30333031 30303030 35395A17 0D323030 31303130 30303030 305A3059 312F302D

<output truncated>
```

自己署名証明書は、セキュア HTTP サーバを無効にして、**no crypto pki trustpoint TP-self-signed-30890755072** グローバル コンフィギュレーション コマンドを入力することで削除できます。その後、セキュア HTTP サーバを再度有効にすると、自己署名証明書が新たに生成されます。

（注） *TP self-signed* の後ろに表示されている値は、デバイスのシリアル番号によって異なります。

オプションのコマンド（**ip http secure-client-auth**）を使用すると、HTTPS サーバがクライアントからの X.509v3 証明書を要求します。クライアントの認証は、サーバ自身の認証よりも高いセキュリティを提供します。

CipherSuite

CipherSuite は暗号化アルゴリズムおよびダイジェストアルゴリズムを指定して、SSL 接続に使用します。HTTPS サーバに接続すると、クライアントの Web ブラウザは、サポート対象の CipherSuite のリストを提供します。その後クライアントとサーバは、両方でサポートされている暗号化アルゴリズムで最適なものをリストから選択してネゴシエートします。たとえば、Netscape Communicator 4.76 は、米国のセキュリティ（RSA 公開キー暗号 MD2、MD5、RC2-CBC、RC4、DES-CBC、および DES-EDE3-CBC）をサポートしています。

最適な暗号化には、128 ビット暗号化をサポートするクライアントブラウザ（Microsoft Internet Explorer バージョン 5.5 以降または Netscape Communicator バージョン 4.76 以降など）が必要です。SSL_RSA_WITH_DES_CBC_SHA CipherSuite は、128 ビット暗号化を提供しないため、他の CipherSuite よりもセキュリティが低くなります。

CipherSuite は、よりセキュリティが高く、複雑になればなるほど、わずかですが処理時間が必要になります。次に、スイッチでサポートされる CipherSuite およびルータの処理負荷（速さ）による CipherSuite のランク（速い順）を定義します。

1. SSL_RSA_WITH_DES_CBC_SHA：メッセージの暗号化に DES-CBC、およびメッセージ ダイジェストにセキュア ハッシュ アルゴリズム（SHA）を使用した RSA のキー交換（RSA 公開キー暗号化）
2. SSL_RSA_WITH_NULL_SHA：メッセージの暗号化に NULL、およびメッセージ ダイジェストに SHA を使用したキー交換（SSL 3.0 専用）。
3. SSL_RSA_WITH_NULL_MD5：メッセージの暗号化に NULL、およびメッセージ ダイジェストに MD5 を使用したキー交換（SSL 3.0 専用）。
4. SSL_RSA_WITH_RC4_128_MD5：RC4 128 ビット暗号化、およびメッセージ ダイジェストに MD5 を使用した RSA のキー交換
5. SSL_RSA_WITH_RC4_128_SHA：RC4 128 ビット暗号化、およびメッセージ ダイジェストに SHA を使用した RSA のキー交換
6. SSL_RSA_WITH_3DES_EDE_CBC_SHA：メッセージの暗号化に 3DES と DES-EDE3-CBC、およびメッセージ ダイジェストに SHA を使用した RSA のキー交換（RSA 公開キー暗号化）
7. SSL_RSA_WITH_AES_128_CBC_SHA：AES 128 ビット暗号化、およびメッセージダイジェストに SHA を使用した RSA のキー交換（SSL 3.0 専用）。
8. SSL_RSA_WITH_AES_256_CBC_SHA：AES 256 ビット暗号化、およびメッセージダイジェストに SHA を使用した RSA のキー交換（SSL 3.0 専用）。
9. SSL_RSA_WITH_AES_128_CBC_SHA：AES 128 ビット暗号化、およびメッセージダイジェストに SHA を使用した RSA のキー交換（SSL 3.0 専用）。
10. SSL_RSA_WITH_AES_256_CBC_SHA：AES 256 ビット暗号化、およびメッセージダイジェストに SHA を使用した RSA のキー交換（SSL 3.0 専用）。

（注） Chrome の最新バージョンは 4 つの元の暗号スイートをサポートしません。そのため、Web GUI とゲスト ポータル両方へのアクセスが拒否されます。

（暗号化およびダイジェスト アルゴリズムをそれぞれ指定して組み合わせた）RSA は、SSL 接続においてキーの生成および認証の両方に使用されます。これは、CA のトラストポイントが設定されているかどうかにかかわりません。

SSL のデフォルト設定

SSL のデフォルト設定は次のとおりです。

- 標準の HTTP サーバはイネーブルに設定されています。
- SSL はイネーブルに設定されています。
- CA のトラストポイントは設定されていません。

• 自己署名証明書は生成されていません。

SSL の設定時の注意事項

SSL をスイッチ クラスタで使用すると、SSL セッションがクラスタ コマンダで終了します。クラスタ メンバのスイッチは標準の HTTP で動作させる必要があります。

CA のトラストポイントを設定する前に、システム クロックが設定されていることを確認してください。クロックが設定されていないと、不正な日付により証明書が拒否されます。

Secure Socket Layer HTTP の設定方法

セキュア HTTP サーバの設定

セキュア HTTP サーバを設定するには、特権 EXEC モードで次の手順を実行します。

始める前に

証明に証明書の認証を使用する場合、前の手順を使用してデバイスの CA トラストポイントを設定してから、HTTP サーバを有効にする必要があります。CA のトラストポイントを設定していない場合、セキュア HTTP サーバを最初に有効にした時点で、自己署名証明書が生成されます。サーバを設定した後、標準およびセキュア HTTP サーバ両方に適用するオプション（パス、適用するアクセスリスト、最大接続数、またはタイムアウトポリシー）を設定できます。

Web ブラウザを使用してセキュア HTTP 接続を確認するには、https://*URL* を入力します（URL は IP アドレス、またはサーバデバイスのホスト名）。デフォルト ポート以外のポートを設定している場合、URL の後ろにポート番号も指定する必要があります。次に例を示します。

（注） AES256_SHA2 はサポートされません。

```
https://209.165.129:1026
```

または

```
https://host.domain.com:1026
```

アクセスリスト（IPv4 ACL のみ）を指定するための従来の **ip http access-class** *access-list-number* コマンドは廃止予定です。引き続きこのコマンドを使用して、HTTP サーバへのアクセスを許可するアクセス リストを指定できます。2 つの新しいコマンドは、IPv4 および IPv6 ACL を指定するためのサポートを有効にするために導入されました。これらは、IPv4 ACL を指定するための **ip http access-class ipv4** *access-list-name* | *access-list-number* と、IPv6 ACL を指定する

ための **ip http access-class ipv6** *access-list-name* です。警告メッセージの受信を防ぐために、新しい CLI の使用をお勧めします。

アクセス リストを指定する際は、次の考慮事項があります。

- 存在しないアクセスリストを指定すると、設定は実行されますが、次の警告メッセージを受信します。

  ```
  ACL being attached does not exist, please configure it
  ```

- HTTP サーバにアクセスリストを指定するために **ip http access-class** コマンドを使用すると、次の警告メッセージが表示されます。

  ```
  This CLI will be deprecated soon, Please use new CLI ip http
  access-class ipv4/ipv6 <access-list-name>| <access-list-number>
  ```

- **ip http access-class ipv4** *access-list-name* | *access-list-number* または **ip http access-class ipv6** *access-list-name* を使用した場合に、アクセスリストがすでに **ip http access-class** を使用して設定されていた場合は、次の警告メッセージが表示されます。

  ```
  Removing ip http access-class <access-list-number>
  ```

ip http access-class *access-list-number* と **ip http access-class ipv4** *access-list-name* | *access-list-number* は同じ機能を共有しています。コマンドを実行するごとに、その前のコマンドのコンフィギュレーションは上書きされます。2 つのコマンドの設定間の次の組み合わせによって、実行コンフィギュレーションへの影響が説明されます。

- **ip http access-class** *access-list-number* がすでに設定されている場合に、**ip http access-class ipv4** *access-list-number* コマンドを使用して設定を行おうとした場合、**ip http access-class** *access-list-number* の設定は削除され、**ip http access-class ipv4** *access-list-number* の設定が実行コンフィギュレーションに追加されます。

- **ip http access-class** *access-list-number* がすでに設定されている場合に、**ip http access-class ipv4** *access-list-name* コマンドを使用して設定を行おうとした場合、**ip http access-class** *access-list-number* の設定は削除され、**ip http access-class ipv4** *access-list-name* の設定が実行コンフィギュレーションに追加されます。

- **ip http access-class ipv4** *access-list-number* がすでに設定されている場合に、**ip http access-class** *access-list-name* を使用して設定を行おうとした場合、**ip http access-class ipv4** *access-list-number* の設定は削除され、**ip http access-class** *access-list-name* の設定が実行コンフィギュレーションに追加されます。

- **ip http access-class ipv4** *access-list-name* がすでに設定されている場合に、**ip http access-class** *access-list-number* を使用して設定を行おうとした場合、**ip http access-class ipv4** *access-list-name* の設定は削除され、**ip http access-class** *access-list-number* の設定が実行コンフィギュレーションに追加されます。

手順

	コマンドまたはアクション	目的

ステップ 1	**show ip http server status**
	例：
	`Device# show ip http server status`
		（任意）HTTP サーバのステータスを表示して、セキュア HTTP サーバの機能がソフトウェアでサポートされているかどうかを判断します。出力で、次のラインのどちらかを確認してください。
		`HTTP secure server capability: Present`
		または
		`HTTP secure server capability: Not present`

ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip http secure-server** 例： `Device(config)# ip http secure-server`	HTTPS サーバがディセーブルの場合、イネーブルにします。HTTPS サーバは、デフォルトでイネーブルに設定されています。
ステップ 4	**ip http secure-port** *port-number* 例： `Device(config)# ip http secure-port 443`	（任意）HTTPS サーバに使用するポート番号を指定します。デフォルトのポート番号は 443 です。443 または 1025 ～ 65535 の範囲で指定できます。
ステップ 5	**ip http secure-ciphersuite** {[**3des-ede-cbc-sha**] [**rc4-128-md5**] [**rc4-128-sha**] [**des-cbc-sha**]} 例： `Device(config)# ip http secure-ciphersuite rc4-128-md5`	（任意）HTTPS 接続の暗号化に使用する CipherSuite（暗号化アルゴリズム）を指定します。特定の CipherSuite を指定する理由がなければ、サーバとクライアントが、両方がサポートする CipherSuite でネゴシエートするように設定します。これがデフォルトです。

	コマンドまたはアクション	目的
ステップ 6	**ip http secure-client-auth** 例： `Device(config)# ip http secure-client-auth`	（任意）HTTP サーバを設定して、接続処理の間、認証のために、クライアントからの X.509v3 証明書を要求します。デフォルトでは、クライアントがサーバからの証明書を要求する設定になっていますが、サーバはクライアントを認証しないようになっています。
ステップ 7	**ip http secure-trustpoint** *name* 例： `Device(config)# ip http secure-trustpoint your_trustpoint`	X.509v3 セキュリティ証明書の取得およびクライアントの証明書接続の認証に使用する CA のトラストポイントを指定します。 （注） このコマンドの使用は、前の手順に従って CA のトラストポイントをすでに設定しているという前提を踏まえて説明しています。
ステップ 8	**ip http path** *path-name* 例： `Device(config)# ip http path /your_server:80`	（任意）HTML ファイルのベースとなる HTTP パスを設定します。パスは、ローカル システムにある HTTP サーバ ファイルの場所を指定します（通常、システムのフラッシュ メモリを指定します）。
ステップ 9	**ip http access-class** *access-list-number* 例： `Device(config)# ip http access-class 2`	（任意）HTTP サーバへのアクセスの許可に使用するアクセス リストを指定します。
ステップ 10	**ip http access-class** { **ipv4** {*access-list-number* | *access-list-name*} | **ipv6** {*access-list-name*} } 例： `Device(config)# ip http access-class ipv4 4`	（任意）HTTP サーバへのアクセスの許可に使用するアクセス リストを指定します。
ステップ 11	**ip http max-connections** *value* 例： `Device(config)# ip http`	（任意）HTTP サーバへの同時最大接続数を指定します。値は 10 以上にすることを推奨します。これは、UI が想定

	コマンドまたはアクション	目的
	max-connections 4	どおりに機能するために必要な値です。
ステップ 12	**ip http timeout-policy idle** *seconds* **life** *seconds* **requests** *value* 例： `Device(config)# ip http timeout-policy idle 120 life 240 requests 1`	（任意）指定の状況下における、HTTP サーバへの接続最大時間を指定します。 • **idle**：データの受信がないか、応答データが送信できない場合の最大時間。指定できる範囲は 1 ～ 600 秒です。デフォルト値は180秒（3分）です。 • **life**：接続を確立している最大時間。指定できる範囲は 1 ～ 86400 秒（24 時間）です。デフォルト値は 180 秒です。 • **requests**：永続的な接続で処理される要求の最大数。最大値は 86400 です。デフォルトは 1 です。
ステップ 13	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

セキュア HTTP クライアントの設定

始める前に

標準の HTTP クライアントおよびセキュア HTTP クライアントは常にイネーブルです。証明書の認証にはセキュア HTTP クライアントの証明書が必要です。次の手順では、前の手順で CA のトラストポイントをデバイスに設定していることを前提にしています。CA のトラストポイントが設定されておらず、リモートの HTTPS サーバがクライアントの認証を要求した場合、セキュア HTTP クライアントへの接続は失敗します。

手順

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

	コマンドまたはアクション	目的
	`Device> enable`	
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip http client secure-trustpoint** *name* 例： `Device(config)# ip http client secure-trustpoint your_trustpoint`	（任意）リモートの HTTP サーバがクライアント認証を要求した場合に使用する、CA のトラストポイントを指定します。このコマンドの使用は、前の手順を使用して CA のトラストポイントをすでに設定しているという前提を踏まえて説明しています。クライアント認証が必要ない場合、またはプライマリのトラストポイントがすでに設定されている場合は、このコマンドは任意です。
ステップ 4	**ip http client secure-ciphersuite** {[**3des-ede-cbc-sha**] [**rc4-128-md5**] [**rc4-128-sha**] [**des-cbc-sha**]} 例： `Device(config)# ip http client secure-ciphersuite rc4-128-md5`	（任意）HTTPS 接続の暗号化に使用する CipherSuite（暗号化アルゴリズム）を指定します。特定の CipherSuite を指定する理由がなければ、サーバとクライアントが、両方がサポートする CipherSuite でネゴシエートするように設定します。これがデフォルトです。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

CA のトラストポイントの設定

セキュア HTTP 接続には、CA のトラストポイントを正式に設定することを推奨します。CA トラストポイントは、自己署名証明書より高いセキュリティがあります。

手順

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

	コマンドまたはアクション	目的
	`Device> enable`	
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**hostname** *hostname* 例： `Device(config)# hostname your_hostname`	デバイスのホスト名を指定します（以前ホスト名を設定していない場合のみ必須）。ホスト名はセキュリティ キーと証明書に必要です。
ステップ 4	**ip domain-name** *domain-name* 例： `Device(config)# ip domain-name your_domain`	デバイスの IP ドメイン名を指定します（以前 IP ドメイン名を設定していない場合のみ必須）。IP ドメイン名はセキュリティ キーと証明書に必要です。
ステップ 5	**crypto key generate rsa** 例： `Device(config)# crypto key generate rsa`	（任意）RSA キーペアを生成します。RSA キーのペアは、デバイスの証明書を入手する前に必要です。RSA キーのペアは自動的に生成されます。必要であれば、このコマンドを使用してキーを再生成できます。
ステップ 6	**crypto ca trustpoint** *name* 例： `Device(config)# crypto ca trustpoint your_trustpoint`	CA のトラストポイントにローカルの設定名を指定して、CA トラストポイント コンフィギュレーション モードを開始します。
ステップ 7	**enrollment url** *url* 例： `Device(ca-trustpoint)# enrollment url http://your_server:80`	デバイスによる証明書要求の送信先の URL を指定します。
ステップ 8	**enrollment http-proxy** *host-name port-number* 例：	（任意）HTTP プロキシサーバを経由して CA から証明書を取得するようにデバイスを設定します。

	コマンドまたはアクション	目的
	`Device(ca-trustpoint)# enrollment http-proxy your_host 49`	• *host-name* には、CA を取得するために使用するプロキシサーバを指定します。 • *port-number* には、CA にアクセスするために使用するポート番号を指定します。
ステップ 9	**crl query** *url* 例： `Device(ca-trustpoint)# crl query ldap://your_host:49`	ピアの証明書が取り消されていないかを確認するために、証明書失効リスト（CRL）を要求するようにデバイスを設定します。
ステップ 10	**primary** *name* 例： `Device(ca-trustpoint)# primary your_trustpoint`	（任意）トラストポイントが CA 要求に対してプライマリ（デフォルト）トラストポイントとして使用されるように指定します。 • *name* には、設定したトラストポイントを指定します。
ステップ 11	**exit** 例： `Device(ca-trustpoint)# exit`	CA トラストポイントコンフィギュレーションモードを終了し、グローバルコンフィギュレーションモードに戻ります。
ステップ 12	**crypto ca authentication** *name* 例： `Device(config)# crypto ca authentication your_trustpoint`	CA の公開キーを取得して CA を認証します。**crypto ca trustpoint** コマンドで使用されている名前と同じ名前を使用します。
ステップ 13	**crypto ca enroll** *name* 例： `Device(config)# crypto ca enroll your_trustpoint`	指定した CA トラストポイントから証明書を取得します。このコマンドは、各 RSA キーのペアに対して 1 つの署名入りの証明書を要求します。
ステップ 14	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

セキュア HTTP サーバおよびクライアントのステータスのモニタリング

SSL セキュア サーバおよびクライアントのステータスをモニタするには、次の表の特権 EXEC コマンドを使用します。

表 *16 : SSL* セキュア サーバおよびクライアントのステータスを表示するコマンド

コマンド	目的
show ip http client secure status	セキュア HTTP クライアントの設定を表示します。
show ip http server secure status	セキュア HTTP サーバの設定を表示します。
show running-config	セキュア HTTP 接続に対して生成された自己署名証明書を表示します。

Secure Socket Layer HTTP の設定例

例：Secure Socket Layer HTTP の設定

次の例は、セキュア HTTP サーバがイネーブルで、セキュア HTTP サーバ用のポートが 1025 に設定され、認証にリモート CA トラストポイントサーバ *CA-trust-local* を使用する場合のコンフィギュレーション セッションです。

```
Device# show ip http server status

HTTP server status: Disabled
HTTP server port: 80
HTTP server authentication method: enable
HTTP server access class: 0
HTTP server base path:
Maximum number of concurrent server connections allowed: 5
Server idle time-out: 600 seconds
Server life time-out: 600 seconds
Maximum number of requests allowed on a connection: 1
HTTP secure server capability: Present
HTTP secure server status: Disabled
HTTP secure server port: 443
HTTP secure server ciphersuite: 3des-ede-cbc-sha des-cbc-sha rc4-128-md5 rc4-12a
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint:


Device# configure terminal
Device(config)# ip http secure-server
Device(config)# ip http client secure-trustpoint CA-trust-local
Device(config)# ip http secure-port 1024
```

```
Invalid secure port value.
Device(config)# ip http secure-port 1025
Device(config)# ip http secure-ciphersuite rc4-128-sha rc4-128-md5
Device(config)# end
```

```
Device# show ip http serversecure status

HTTP secure server status: Enabled
HTTP secure server port: 1025
HTTP secure server ciphersuite: rc4-128-md5 rc4-128-sha
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint: CA-trust-local
```

次の例では、CA トラストポイント「CA-trust-local」が指定されており、HTTPS クライアントはクライアント認証要求に対してこのトラストポイントを使用するように設定されています。

```
Device# config terminal
Device(config)# crypto ca trustpoint CA-trust-local
Device(ca-trustpoint)# enrollment url http://example.com
Device(ca-trustpoint)# crl query ldap://example.com
Device(ca-trustpoint)# primary
Device(ca-trustpoint)# exit
Device(config)# ip http client secure-trustpoint CA-trust-local
Device(config)# end
Device# copy running-config startup-config
```

Secure Socket Layer HTTP に関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

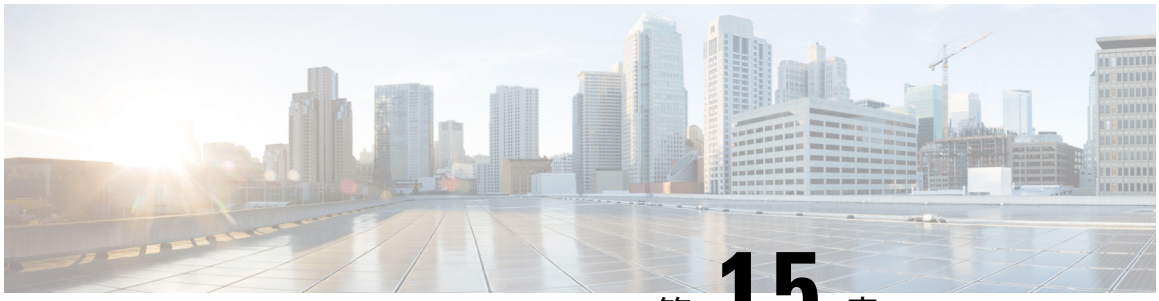
Secure Socket Layer HTTP の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	Secure Socket Layer HTTP	シスコが実装するセキュア HTTP サーバおよび HTTP クライアントでは、アプリケーション層の暗号化に SSL バージョン 3.0 を使用します。セキュア HTTP 接続の場合、HTTP サーバが送受信するデータは暗号化されてインターネットに送信されます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 15 章

認証局の相互運用性

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

認証局の前提条件

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

認証局の制約事項

CA を設定する際には次の制約事項が適用されます。

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

認証局について

この項では、認証局について説明します。

CA でサポートされる規格

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

シスコでは、この機能で次の規格をサポートしています。

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

CA の目的

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

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

公開キー暗号化によりイネーブルにされたデジタル署名は、デバイスおよび個人ユーザをデジタル認証します。RSA 暗号化システムなどの Public Key Cryptography では、各ユーザは、公開キーと秘密キーの両方を含むキーペアを使用します。これらのキーは、補足として機能し、一方で暗号化されたものは、もう一方で復号化できます。つまり、署名は、データがユーザの秘密キーで暗号化されるときに形成されます。受信者は、送信者の公開キーを使用してメッセージを復号化することで、署名を検証します。送信側の公開キーを使用してメッセージを復号できたという事実から、そのメッセージが秘密キーの所有者つまり送信者によって作成されたことがわかります。このプロセスでは、受信者が送信者の公開キーのコピーを持っていること、およびそのキーが送信者になりすました別人ではなく送信者本人のものであることを受信者が強く確信していることが重要です。

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

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

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

登録局

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

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

認証局の設定方法

この項では、認証局の設定方法を説明します。

NVRAM メモリ使用率の管理

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

通常、デバイスには次の証明書が保存されます。

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

CRL は通常、次の条件に従ってデバイスで保存されます。

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

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

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

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

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

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

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

手順

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

デバイス ホスト名および IP ドメイン名の設定

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

手順

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

	コマンドまたはアクション	目的
ステップ 4	**ip domain-name** *name* 例： `Device(config)# ip domain-name domain.com`	デバイスのIP ドメイン名を設定します。
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーションを終了して、特権 EXEC モードに戻ります。

RSA キー ペアの生成

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

手順

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

認証局の宣言

デバイスが使用する 1 つの認証局（CA）を宣言する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**crypto pki trustpoint** *name* 例： `Device(config)# crypto pki trustpoint ka`	デバイスが使用する認証局（CA）を宣言し、CA トラストポイント コンフィギュレーション モードを開始します。
ステップ 4	**enrollment url** *url* 例： `Device(ca-trustpoint)# enrollment url http://entrust:81`	登録要求の送信先とする CA サーバの URL を指定します。
ステップ 5	**enrollment command** 例： `Device(ca-trustpoint)# enrollment command`	登録のため CA に送信される HTTP コマンドを指定します。
ステップ 6	**exit** 例： `Device(ca-trustpoint)# exit`	CAプロファイル登録コンフィギュレーションモードを終了してグローバルコンフィギュレーションモードに戻ります。
ステップ 7	**crypto pki trustpoint** *name* 例： `Device(config)# crypto pki trustpoint ka`	デバイスで使用するトラストポイントを宣言し、CA トラストポイント コンフィギュレーション モードを開始します。
ステップ 8	**crl query ldap**://*url*:[*port*] 例： `Device(ca-trustpoint)# crl query ldap://bar.cisco.com:3899`	証明書失効リスト（CRL）を照会し、ピアの証明書が失効していないことを確認します。
ステップ 9	**enrollment** {**mode ra** | **retry count** *number* | **retry period** *minutes* | **url** *url*} 例： `Device(ca-trustpoint)# enrollment retry period 2`	証明書要求を再試行するまでの登録待機時間を指定します。

	コマンドまたはアクション	目的
ステップ 10	**enrollment** {**mode ra** | **retry count** *number* | **retry period** *minutes* | **url** *url*} 例： `Device(ca-trustpoint)# enrollment retry count 8`	以前の要求への応答が得られない場合にデバイスが証明書要求を再送信する回数を指定します。
ステップ 11	**revocation-check** *method1* [*method2 method3*] 例： `Device(ca-trustpoint)# revocation-check crl ocsp`	証明書の失効ステータスをチェックします。
ステップ 12	**end** 例： `Device(ca-trustpoint)# end`	CA トラストポイント コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

ルート CA（信頼できるルート）の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**crypto pki trustpoint** *name* 例： `Device(config)# crypto pki trustpoint ka`	デバイスで使用するトラストポイントを宣言し、CA トラストポイント コンフィギュレーション モードを開始します。
ステップ 4	**revocation-check** *method1* [*method2 method3*] 例： `Device(ca-trustpoint)# revocation-check ocsp`	証明書の失効ステータスをチェックします。
ステップ 5	**root tftp** *server-hostname filename* 例：	TFTP 経由で認証局（CA）の証明書を取得します。

	コマンドまたはアクション	目的
	`Device(ca-trustpoint)# root tftp server1 file1`	
ステップ 6	**enrollment http-proxy** *hostname port-number* 例： `Device(ca-trustpoint)# enrollment http-proxy host2 8080`	HTTP を使用して、プロキシサーバ経由で認証局（CA）にアクセスします。
ステップ 7	**end** 例： `Device(ca-trustpoint)# end`	CA トラストポイント コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

CA の認証

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

CA の公開キーを取得するには次のタスクを実行します。

手順

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

署名証明書の要求

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

CA から署名証明書を要求するには、次の作業を実行します。

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

手順

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

次のタスク

設定の保存

設定の変更を行った場合は、必ず作業結果を保存するようにしてください。

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

認証局のモニタリングと維持

この項では、認証局のモニタリングと維持について説明します。

証明書失効リストの要求

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

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

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

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

手順

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

証明書失効リストの照会

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

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

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

手順

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

デバイスからの RSA キーの削除

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

]

手順

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

次のタスク

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

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

ピアの公開キーの削除

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例：	グローバル コンフィギュレーション モードを開始します。

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

設定からの証明書の削除

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**show crypto pki certificates** 例： `Device# show crypto pki certificates`	デバイスの証明書、認証局（CA）証明書、および任意の登録局（RA）証明書に関する情報を表示します。
ステップ 3	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 4	**crypto pki certificate chain** *name* 例：	証明書チェーン コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	`Device(config)# crypto pki certificate chain myca`	
ステップ 5	**no certificate** *certificate-serial-number* 例： `Device(config-cert-chain)# no certificate 0123456789ABCDEF0123456789ABCDEF`	証明書を削除します。
ステップ 6	**exit** 例： `Device(config-cert-chain)# exit`	証明書チェーン コンフィギュレーション モードを終了し、グローバル コンフィギュレーションモードに戻ります。
ステップ 7	**no crypto pki import** *name* **certificate** 例： `Device(config)# no crypto pki import MS certificate`	証明書を手動で削除します。
ステップ 8	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

キーと証明書の表示

キーと証明書を表示するには次の作業を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**show crypto key mypubkey rsa** [*keyname*] 例： `Device# show crypto key mypubkey rsa [keyname]`	デバイスで設定されている RSA 公開キーを表示します。
ステップ 3	**show crypto key pubkey-chain rsa** 例： `Device# show crypto key pubkey-chain rsa`	デバイスに保存されている、ピアのRSA公開キーを表示します。
ステップ 4	**show crypto key pubkey-chain rsa** [**name** *key-name* | **address** *key-address*]	特定のキーのアドレスを表示します。

	コマンドまたはアクション	目的
	例： `Device# show crypto key pubkey-chain rsa address 209.165.202.129`	
ステップ 5	**show crypto pki certificates** 例： `Device# show crypto pki certificates`	デバイスの証明書、認証局（CA）証明書、および任意の登録局（RA）証明書に関する情報を表示します。
ステップ 6	**show crypto pki trustpoints** 例： `Device# show crypto pki certificates`	デバイスで設定されているトラストポイントを表示します。

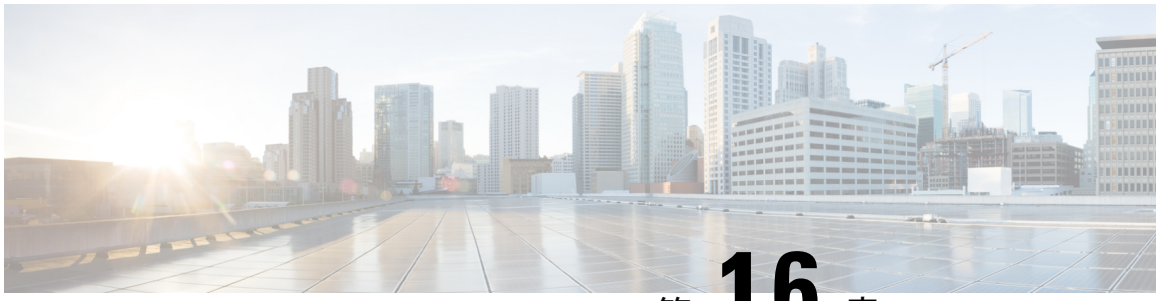
認証局相互運用性の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	認証局の相互運用性	CA の相互運用性により、Cisco IOS デバイスと CA の通信が可能になり、Cisco IOS デバイスが CA からデジタル証明書を取得して使用できるようになります。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 16 章

アクセス コントロール リストの概要

アクセスリストは、パケットをデバイスのインターフェイスで転送するかブロックするかを制御して、ネットワーク トラフィックをフィルタリングします。デバイスは各パケットを調べ、アクセスリスト内で指定されている基準に基づいて、そのパケットの転送またはドロップを決定します。

アクセスリストで指定できる条件には、トラフィックの送信元アドレス、トラフィックの宛先アドレス、または上位層のプロトコルなどが含まれます。

（注） これらのリストは認証を必要としないため、一部のユーザは基本的なアクセスリストを回避できる可能性があります。

アクセス コントロール リストについて

アクセスリストは、パケットをデバイスのインターフェイスで転送するかブロックするかを制御して、ネットワーク トラフィックをフィルタリングします。デバイスは各パケットを調べ、アクセスリスト内で指定されている基準に基づいて、そのパケットの転送またはドロップを決定します。

アクセスリストで指定できる条件には、トラフィックの送信元アドレス、トラフィックの宛先アドレス、または上位層のプロトコルなどが含まれます。

（注） これらのリストは認証を必要としないため、一部のユーザは基本的なアクセスリストを回避できる可能性があります。

アクセス リストの定義

アクセスリストは、少なくとも 1 つの **permit** ステートメント、および任意の 1 つまたは複数の **deny** ステートメントで構成される順次リストです。IP アドレス リストの場合、ステートメントは IP アドレス、上位層の IP プロトコルなどの IP パケットのフィールドに適用できます。アクセスリストは名前または番号で識別および参照されます。アクセスリストはパケットフィルタとして動作し、アクセスリストに定義されている条件に基づいてパケットのフィルタ処理を行います。

アクセスリストを設定しても、アクセスリストがインターフェイスまたは仮想端末回線（VTY）に適用されるか、アクセスリストを受け入れるコマンドで参照されるまでは、有効になりません。複数のコマンドから同じアクセス リストを参照できます。

次に、branchoffices という名前の IP アクセス リストを作成するための設定例を示します。ACL は着信パケットの gigabitEthernet に適用されます。このインターフェイスにアクセスできるのは、個々の各送信元アドレスとマスク ペアで指定されているネットワーク上の送信元のみです。ネットワーク 172.20.7.0 上の送信元から発信されるパケットの宛先に、制限はありません。ネットワーク 172.29.2.0 上の送信元から発信されるパケットの宛先は、172.25.5.4 にする必要があります。

```
ip access-list extended branchoffices
 10 permit 172.20.7.0 0.0.0.3 any
 20 permit 172.29.2.0 0.0.0.255 host 172.25.5.4
!
gigabitEthernet 1/0/1
 ip access-group branchoffices in
```

アクセス コントロール リストの機能

アクセス リストを設定する理由は多数あります。たとえば、ルーティング アップデートのコンテンツの制限や、トラフィック フローの制御などです。アクセス リストを設定する最も重要な理由の 1 つは、このモジュールの要であるネットワークにセキュリティを提供することです。

アクセスリストを使用することで、ネットワークにアクセスするための基本的なセキュリティレベルが実現します。デバイスでアクセス リストを設定しないと、デバイスを通過するすべてのパケットに、ネットワーク全体へのアクセスが許可されます。

アクセスリストでは、あるホストにはネットワークの一部へのアクセスを許可する一方、別のホストにはそれと同じ領域へのアクセスを禁止することが可能です。次の図では、ホスト A にはヒューマン リソース ネットワークへのアクセスが許可されていますが、ホスト B にはヒューマン リソース ネットワークへのアクセスが禁止されています。

また、アクセス リストを使用して、デバイス インターフェイスで転送またはブロックするトラフィックの種類を定義することもできます。たとえば、電子メール トラフィックのルーティングを許可し、同時にすべての Telnet トラフィックをブロックすることができます。

IP アクセス リストの目的

アクセス リストは、パケット フィルタリングを実行して、ネットワークを介して移動するパケットとその場所を制御します。この処理は、ネットワーク トラフィックを制限し、ユーザやデバイスによるネットワークへのアクセスを制限するのに役立ちます。アクセス リストの用途は多様なので、多くのコマンドシンタックスでアクセス リストが参照されます。アクセス リストを使用して、次のようなことを実行できます。

- インターフェイスでの着信パケットのフィルタリング
- インターフェイスでの発信パケットのフィルタリング
- ルーティング アップデートの内容の制限
- アドレスまたはプロトコルに基づくデバッグ出力の制限
- 仮想端末回線アクセスの制御
- 輻輳回避、輻輳管理、プライオリティおよびカスタム キューイングなどの高度な機能に使用されるトラフィックの特定または分類
- ダイヤルオンデマンド ルーティング（DDR）呼び出しのトリガー

ACL を設定する理由

アクセス リストを設定する理由は多数あります。たとえば、アクセス リストを使用して、スイッチング アップデートのコンテンツを制限したり、トラフィック フローを制御したりできます。アクセスリストを設定する最も重要な理由の1つは、ネットワークに対するアクセスを制御することで、ネットワークに基本レベルのセキュリティを提供することです。デバイスでアクセスリストを設定しない場合、デバイスを通過するすべてのパケットは、ネットワークのすべての部分で許可される可能性があります。

アクセスリストで、ネットワークの一部に対してアクセスを許可するホストと、同じ領域に対してアクセスを禁止するホストを設定できます。たとえば、適切なアクセスリストをデバイスのインターフェイスに適用することで、ホスト A にはヒューマン リソース ネットワークへのアクセスが許可され、ホスト B にはヒューマン リソース ネットワークへのアクセスが禁止されます。

ネットワークの2つの部分の間に配置されたデバイスにアクセスリストを使用して、内部ネットワークの特定の部分で発着信するトラフィックを制御できます。

アクセスリストのセキュリティ上の利点を実現するために、少なくとも境界デバイスでアクセスリストを設定する必要があります。境界デバイスとは、ネットワークのエッジにあるデバイスです。このようなアクセスリストは、外部ネットワークから、または内部ネットワークのあまり制御されていない領域から、内部ネットワークの機密性が高い領域に対する基本的なバッファとして機能します。このような境界デバイスでは、デバイスのインターフェイスに設定されている各ネットワーク プロトコルに合わせてアクセス リストを設定する必要があります。着信トラフィック、発信トラフィック、またはその両方がインターフェイスでフィルタされるように、アクセス リストを設定できます。

アクセス リストは個々のプロトコル ベースで定義されます。つまり、各プロトコルのトラフィック フローを制御する場合、インターフェイスでイネーブルにするプロトコルごとにアクセス リストを定義する必要があります。

アクセス リストのソフトウェア処理

アクセス リストがインターフェイス、vty に適用されるとき、あるいはコマンドで参照されるときの処理方法を説明した一般的な手順を次に示します。この手順は、アクセス リスト エントリが 13 以下のアクセス リストに適用されます。

- ソフトウェアが IP パケットや各パケットのテスト部分を受け取ります。これらは、アクセスリストの条件に一度に 1 つずつ（**permit** または **deny** ステートメント）照らし合わせてフィルタリングされます。たとえば、ソフトウェアは、**permit** あるいは **deny** ステートメントの送信元アドレスおよび宛先アドレスに照らし合わせてパケットの送信元アドレスおよび宛先アドレスをテストします。
- パケットがアクセス リストのステートメントに一致しないと、そのパケットはリスト内の次のステートメントに対してテストされます。
- パケットとアクセス リスト ステートメントが一致すると、リスト内の残りのステートメントはスキップされ、パケットは一致したステートメントに指定されたとおりに許可または拒否されます。パケットが許可されるか拒否されるかは、パケットが一致する最初のエントリによって決まります。つまり、一致すると、それ以降のエントリは考慮されません。
- いずれの条件とも一致しなかった場合、パケットは廃棄されます。これは、各アクセスリストが暗黙の **deny** ステートメントで終了するためです。言い換えると、パケットが各ステートメントに対してテストされたときまでに許可されないと、このパケットは拒否されます。

13 を超えるエントリが含まれるアクセス リストは、trie ベースのルックアップ アルゴリズムを使用して処理されます。このプロセスは自動的に行われます。設定する必要はありません。

アクセス リストのルール

アクセス コントロール リスト（ACL）には、次のルールが適用されます。

- 1 つのインターフェイス、1 つのプロトコル、1 つの方向につき、許可されるアクセス リストは 1 つだけです。
- アクセスリストには少なくとも 1 つの **permit** ステートメントが含まれる必要があります。そうしないと、ネットワークに入るすべてのパケットが拒否されます。
- アクセスリスト条件または一致基準の構成順序は重要です。パケットを転送するかブロックするかを決定するときに、シスコソフトウェアは、それぞれの条件ステートメントに対してステートメントの作成順にパケットをテストします。一致が見つかると、条件ステートメントはそれ以上チェックされません。同じ **permit** ステートメントまたは **deny** ステー

トメントでも、順序が異なる場合、ある状況では通過し、別の状況では拒否されるパケットが生じる可能性があります。

- アクセス リストを名前によって参照したときに、そのアクセス リストが存在しない場合は、すべてのパケットが通過します。インターフェイスまたはコマンドに空のアクセス リストを適用すると、ネットワークに対するすべてのトラフィックが許可されます。
- 標準のアクセス リストと拡張のアクセス リストの名前は同じにできません。
- パケットがアウトバウンド インターフェイスに送信される前に、インバウンド アクセス リストがパケットを処理します。ネットワークへのパケットアクセスを拒否するフィルタ条件があるインバウンド アクセス リストは、ルート ルックアップ時のオーバーヘッドを削減します。構成されたフィルタ基準に基づいてネットワークへのアクセスを許可されたパケットはルーティング処理されます。インバウンド アクセス リストの場合、**permit** ステートメントを構成するとパケットは受信後に処理され、**deny** ステートメントを構成するとパケットは破棄されます。
- アウトバウンドアクセスリストの場合、パケットの処理後にデバイスから送信されます。着信パケットはアウトバウンドインターフェイスにルーティングされてから、アウトバウンドアクセス リストで処理されます。アウトバウンドアクセス リストの場合、**permit** ステートメントを構成するとパケットは出力バッファに送信され、**deny** ステートメントを構成するとパケットは破棄されます。
- アクセスリストで、デバイスに到達するトラフィック、またはデバイス経由で送信されるトラフィックは制御できますが、デバイスが送信元のトラフィックは制御できません。

IP アクセス リストを作成する際に役立つヒント

意図しない結果を回避し、より効率的なアクセスリストを作成するために役立つヒントを紹介します。

- アクセスリストを作成してから、インターフェイス（または別の対象）に適用します。その理由は、存在しないアクセス リストをインターフェイスに適用してから、アクセス リストを設定すると、最初のステートメントが有効になり、それに続く暗黙的な**deny**ステートメントによってアクセスに緊急の問題が発生するおそれがあるためです。
- アクセス リストを設定してから適用するもう 1 つの理由は、空のアクセス リストが適用されたインターフェイスはすべてのトラフィックを許可するためです。
- すべてのアクセス リストには、少なくとも 1 つの **permit** ステートメントが必要です。permit がないと、すべてのパケットは拒否され、トラフィックはまったく通過しません。
- 最初に（**permit** または **deny** ステートメントに対する）一致が見つかった後は条件のテストが終了するため、パケットが一致する可能性の高いステートメントをアクセス リストの先頭に配置すると処理にかかる時間とリソースが削減されます。最も頻繁に発生する条件を発生頻度の低い条件より前に配置します。
- ネットワークまたはサブネットのより具体的な参照が、より全般的な参照よりも前に出現するように、アクセス リストを構成します。

- まだ拒否されていないその他のパケットすべてを許可する場合、ステートメント **permit any any** を使用します。ステートメント **permit any any** を使用すると、実質的に、アクセス リストの末尾にある暗黙的な deny ステートメントでその他すべてのパケットが拒否されることを防ぎます。最初のアクセス リスト エントリは **permit any any** にしないでください。すべてのトラフィックが通過し、以降のテストに到達するパケットがなくなります。**permit any any** を指定すると、まだ拒否されていないすべてのトラフィックが通過します。
- すべてのアクセス リストは暗黙的な **deny** ステートメントで終了しますが、明示的な **deny** ステートメント（たとえば **deny ip any any**）の使用を推奨します。ほとんどのプラットフォームでは、**show access-list**コマンドを発行して拒否されるパケット数を表示し、アクセス リストが許可していないパケットに関する詳細情報を調査できます。明示的な **deny** ステートメントで拒否されたパケットのみがカウントされます。これは、明示的な **deny** ステートメントによって、より詳細なデータが生成されるためです。
- アクセス リストの作成中、または作成後に、エントリを削除する場合があります。
  - 番号付きアクセス リストからはエントリを削除できません。削除しようとすると、アクセス リスト全体が削除されます。エントリを削除する必要がある場合、アクセス リスト全体を削除してから最初から作り直す必要があります。
  - 名前付きアクセス リストからはエントリを削除できます。**no permit** または **no deny** コマンドを使用すると、適切なエントリが削除されます。
- 個々のステートメントの用途をひと目で確認および理解しやすくするために、**remark** コマンドを使用して、ステートメントの前後に役立つ注記を書き込むことができます。
- 特定のホストまたはネットワークに対するアクセスを拒否し、そのネットワークまたはホストの誰かがアクセスしようとしたかどうかを検出する場合、対応する **deny** ステートメントを指定した **log** キーワードを含めます。それによって、その送信元からの拒否されたパケットがログに記録されます。
- このヒントは、アクセス リストの配置に適用されます。リソースを保存しようとすると、インバウンドアクセスリストでは常にフィルタ条件を適用した後に、ルーティングテーブルの検索を行います。アウトバウンドアクセス リストではフィルタ条件を適用する前に、ルーティング テーブルの検索を行います。

アクセスを制御するためにフィルタできる IP パケット フィールド

拡張アクセスリストを使用すると、IPパケットに含まれる次の任意のフィールドについてフィルタできます。送信元アドレスおよび宛先アドレスは、アクセスリストの基礎として最もよく指定される 2 つのフィールドです。

- 送信元アドレス - 特定のネットワーキング デバイスまたはホストから送信されるパケットを制御するために、送信元アドレスを指定します。
- 宛先アドレス - 特定のネットワーキング デバイスまたはホストに対して送信されるパケットを制御するために、宛先アドレスを指定します。

送信元アドレスと宛先アドレス

IP パケットの送信元アドレスと宛先アドレスのフィールドは、アクセス リストの基礎となる典型的な2つのフィールドです。送信元アドレスを指定して、特定のネットワーキング デバイスまたはホストから送信されるパケットを制御します。宛先アドレスを指定して、特定のネットワーキング デバイスまたはホストに送信されるパケットを制御します。

アクセス リストのアドレスに対するワイルドカード マスク

アドレス フィルタリングでは、アクセス リスト エントリ内のアドレス ビットとアクセス リストに送信されるパケットを比較するときに、対応する IP アドレスを確認するか無視するかをソフトウェアに示すために、ワイルドカード マスクを使用します。注意してワイルドカード マスクを設定することで、許可または拒否テストのために 1 つまたは複数の IP アドレスを指定できます。

IP アドレス ビット用のワイルドカード マスクでは、数値 1 と数値 0 を使用して、対応する IP アドレス ビットをどのように扱うかを指定します。1 と 0 は、サブネット（ネットワーク）マスクで意味する内容が対照的なため、ワイルドカード マスクは逆マスクとも呼ばれます。

- ワイルドカード マスク ビット 0 は、対応するビット値を確認することを示します。ビット値は一致する必要があります。
- ワイルドカード マスク ビット 1 は、対応するビット値を無視することを示します。ビット値が一致する必要はありません。

アクセス リスト ステートメントの送信元アドレスまたは宛先アドレスでワイルドカード マスクを指定しない場合、0.0.0.0（すべての値が一致する必要があることを示します）という暗黙的なワイルドカード マスクが想定されます。

サブネット マスクでは、ネットワークとサブネットを示す隣接ビットをマスクにする必要がありますが、それとは異なり、ワイルドカード マスクではマスクに非隣接ビットを使用できます。

次の表に、アクセス リストの IP アドレスおよびマスクと、それに一致すると見なされる対応するアドレスの例を示します。

表 *17*：*IP* アドレス、ワイルドカード マスク、および一致する結果の例

アドレス	ワイルドカード マスク	一致する結果
0.0.0.0	255.255.255.255	すべてのアドレスはアクセス リスト条件に一致します
172.18.0.0/16	0.0.255.255	ネットワーク 172.18.0.0
172.18.5.2/16	0.0.0.0	ホスト 172.18.5.2 のみが一致します
172.18.8.0	0.0.0.7	サブネット 172.18.8.0/29 のみが一致します

アドレス	ワイルドカード マスク	一致する結果
172.18.8.8	0.0.0.7	サブネット 172.18.8.8/29 のみが一致します
172.18.8.15	0.0.0.3	サブネット 172.18.8.15/30 のみが一致します
10.1.2.0	0.0.254.255（マスクの非隣接ビット）	10.1.2.0 ～ 10.1.254.0 に含まれる偶数のネットワークに一致します

アクセス リストのシーケンス番号

IP アクセス リスト エントリにシーケンス番号を適用する機能によって、アクセス リストの変更が簡易になります。IP アクセス リスト エントリ シーケンス番号機能の前には、アクセス リスト内のエントリの位置を指定する方法はありませんでした。以前は、既存のリストの途中にエントリを挿入する場合、目的の位置の後にあるすべてのエントリを削除してから、新しいエントリを追加し、削除したすべてのエントリを再入力する必要がありました。これは手間がかかり、エラーが起こりやすい方法です。

この新しい機能を使用すると、アクセス リスト エントリにシーケンス番号を追加し、順序を変更することができます。新しいエントリを追加する場合、アクセスリストの目的の位置に挿入されるようにシーケンス番号を指定します。必要に応じて、アクセスリストの現在のエントリを並べ替えて、新しいエントリを挿入できる場所を作成できます。

ACL でサポートされるタイプ

スイッチは、IP ACL とイーサネット（MAC）ACL をサポートします。

- IP ACL は、TCP、ユーザ データグラム プロトコル（UDP）、インターネット グループ管理プロトコル（IGMP）、およびインターネット制御メッセージプロトコル（ICMP）などの IPv4 トラフィックをフィルタリングします。
- イーサネット ACL は非 IP トラフィックをフィルタリングします。

サポートされる ACL

スイッチは、トラフィックをフィルタリングする ACL の次のタイプをサポートします。

- ポート ACL は、レイヤ 2 インターフェイスに入るトラフィックをアクセス コントロールします。アクセスリストの各タイプ（IPv4 および MAC）のそれぞれの入力方向に、レイヤ 2 インターフェイスに対するポート ACL を適用できます。

ポート ACL

ポート ACL は、スイッチのレイヤ 2 インターフェイスに適用される ACL です。ポート ACL を使用できるのは、物理インターフェイスだけです。EtherChannel インターフェイスでは使用

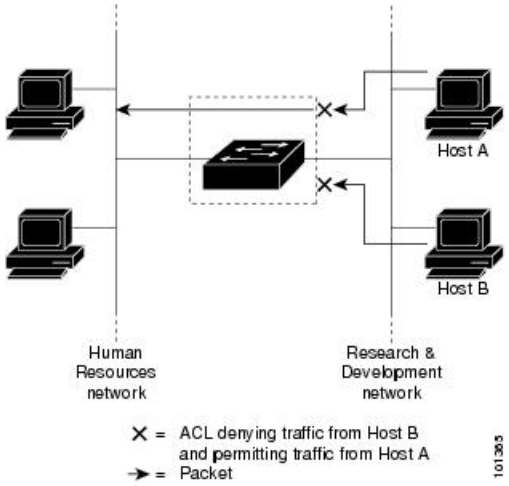
できません。ポート ACL は、着信方向のインターフェイスに適用できます。次のアクセス リストがサポートされています。

- 送信元アドレスを使用する IP アクセス リスト
- 送信元および宛先のアドレスと任意でプロトコル タイプ情報を使用できる拡張 IP アクセス リスト
- 送信元および宛先の MAC アドレスと任意でプロトコル タイプ情報を使用できる MAC 拡張アクセス リスト

スイッチは、インターフェイス上の ACL を調べ、パケットが ACL 内のエントリとどのように一致するかに基づいてパケットの転送を許可または拒否します。このように、ACL がネットワークまたはネットワークの部分へのアクセスを制御します。

図 *4 : ACL* によるネットワーク内のトラフィックの制御

次に、すべてのワークステーションが同じ VLAN にある場合にポート ACL を使用してネットワークへのアクセスを制御する例を示します。レイヤ 2 の着信方向に適用された ACL は、ホスト A がヒューマン リソース ネットワークにアクセスすることを許可しますが、ホスト B が同一のネットワークにアクセスすることは拒否します。ポート ACL は、着信方向のレイヤ 2



インターフェイスだけに適用できます。

ポート ACL をトランク ポートに適用すると、ACL はそのトランク ポート上のすべての VLAN でトラフィックをフィルタリングします。ポート ACL を音声 VLAN ポートに適用すると、ACL はデータ VLAN と音声 VLAN の両方でトラフィックをフィルタリングします。

ポート ACL では、IP アクセス リストを使用して IP トラフィックをフィルタリングでき、MAC アドレスを使用して非 IP トラフィックをフィルタリングできます。同じレイヤ 2 インターフェイス上で IP トラフィックと非 IP トラフィックの両方をフィルタリングするには、そのインターフェイスに IP アクセス リストと MAC アクセス リストの両方を適用します。

（注） レイヤ 2 インターフェイスに適用できるのは、IP アクセス リスト 1 つと MAC アクセス リスト 1 つだけです。すでに IP アクセス リストまたは MAC アクセス リストが設定されているレイヤ 2 インターフェイスに、新しい IP アクセス リストまたは MAC アクセス リストを適用すると、前に設定した ACL が新しい ACL に置き換わります。

アクセス コントロール エントリ

ACL には、アクセス コントロール エントリ（ACE）の順序付けられたリストが含まれています。各 ACE には、*permit* または *deny* と、パケットが ACE と一致するために満たす必要のある一連の条件を指定します。*permit* または *deny* の意味は、ACL が使用されるコンテキストによって変わります。

ACE およびフラグメント化されたトラフィックとフラグメント化されていないトラフィック

IP パケットは、ネットワークを通過するときにフラグメント化されることがあります。その場合、TCP または UDP ポート番号や ICMP タイプおよびコードなどのレイヤ 4 情報は、パケットの最初の部分があるフラグメントだけに含まれます。他のフラグメントには、この情報はありません。

アクセス コントロール エントリ（ACE）には、レイヤ 4 情報をチェックしないため、すべてのパケット フラグメントに適用されるものがあります。レイヤ 4 情報を調べる ACE は、フラグメント化された IP パケットのほとんどのフラグメントに標準的な方法では適用できません。フラグメントにレイヤ 4 情報が含まれておらず、ACE が一部のレイヤ 4 情報をチェックする場合、一致ルールは次のように変更されます。

- フラグメント内のレイヤ 3 情報（TCP や UDP などのプロトコル タイプを含む）をチェックする許可 ACE は、含まれていないレイヤ 4 情報の種類にかかわらず、フラグメントと一致すると見なされます。

  （注） L4 Ops をともなう ACE の TCP では、フラグメント化パケットは RFC 1858 ごとにドロップします。

- レイヤ 4 情報をチェックする拒否 ACE は、フラグメントにレイヤ 4 情報が含まれていない限り、フラグメントと一致しません。

例：ACE およびフラグメント化されたトラフィックとフラグメント化されていないトラフィック

次のコマンドで構成され、フラグメント化された 3 つのパケットに適用されるアクセス リスト 102 を例に取って説明します。

```
Device(config)# access-list 102 permit tcp any host 10.1.1.1 eq smtp
Device(config)# access-list 102 deny tcp any host 10.1.1.2 eq telnet
Device(config)# access-list 102 permit tcp any host 10.1.1.2
Device(config)# access-list 102 deny tcp any any
```

（注） 最初の 2 つの ACE には宛先アドレスの後に **eq** キーワードがありますが、これは既知の TCP 宛先ポート番号がそれぞれシンプルメール転送プロトコル（SMTP）および Telnet と一致するかどうかをチェックすることを意味します。

- パケット A は、ホスト 10.2.2.2 のポート 65000 からホスト 10.1.1.1 の SMTP ポートに送信される TCP パケットです。このパケットがフラグメント化された場合、レイヤ 4 情報がすべて揃っているため、完全なパケットである場合と同じように最初のフラグメントが最初の ACE（permit）と一致します。残りのフラグメントも最初の ACE と一致します。これは、それらのフラグメントに SMTP ポート情報が含まれていなくても、最初の ACE が適用されたときにレイヤ 3 情報だけをチェックするからです。この例の情報は、パケットが TCP であることと、宛先が 10.1.1.1 であることです。
- パケット B は、ホスト 10.2.2.2 のポート 65001 からホスト 10.1.1.2 の Telnet ポートに送信されます。このパケットがフラグメント化された場合、レイヤ 3 情報とレイヤ 4 情報がすべて揃っているため、最初のフラグメントが 2 つめの ACE（deny）と一致します。残りのフラグメントは、レイヤ 4 情報が含まれていないため、2 つめの ACE と一致しません。残りのフラグメントは 3 つめの ACE（permit）と一致します。

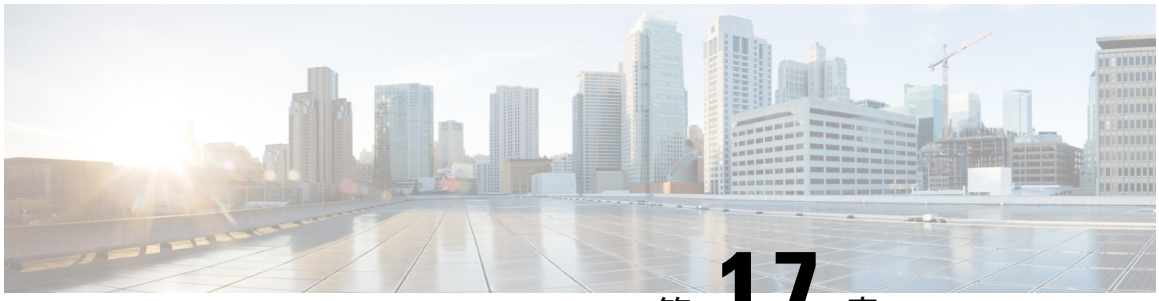
  最初のフラグメントが拒否されたため、ホスト 10.1.1.2 は完全なパケットを再構成できず、その結果、パケット B は拒否されます。ただし、以降の許可されたフラグメントがネットワークの帯域幅を使用し、ホスト 10.1.1.2 がパケットを再構成しようとするときにホストのリソースが消費されます。
- フラグメント化されたパケット C は、ホスト 10.2.2.2 のポート 65001 からホスト 10.1.1.3 のポート ftp に送信されます。このパケットがフラグメント化された場合、最初のフラグメントが 4 つめの ACE（deny）と一致します。ACE はレイヤ 4 情報をチェックせず、すべてのフラグメントのレイヤ 3 情報に宛先がホスト 10.1.1.3 であることが示され、前の permit ACE は異なるホストをチェックしていたため、他のフラグメントもすべて 4 つめの ACE と一致します。

アクセスコントロールリストの概要に関する追加情報

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*

関連項目	マニュアル タイトル
ACL	詳細については、以下を参照してください。 • 『Security Configuration Guide』の「Configuring IPv4 Access Control Lists」 • 『Security Configuration Guide』の「Configuring IPv6 Access Control Lists」



第 17 章

IPv4 アクセス コントロール リストの設定

IPv4アクセスコントロールリストの設定に関する制約事項

一般的なネットワーク セキュリティ

次は、ACL によるネットワーク セキュリティの設定の制約事項です。

- 出力 ACL はレイヤ 2 インターフェイスではサポートされていません。
- ルータ ACL と VLAN ACL はサポートされていません。
- 番号付き ACL で使用できるすべてのコマンドが名前付き ACL でも使用できるわけではありません。インターフェイスのパケット フィルタおよびルート フィルタ用の ACL では、名前を使用できます。
- 標準 ACL と拡張 ACL に同じ名前は使用できません。
- **appletalk** は、コマンドラインのヘルプ ストリングに表示されますが、**deny** および **permit** MAC アクセスリスト コンフィギュレーション モード コマンドの一致条件としてサポートされていません。
- ACL ワイルドカードは、ダウンストリーム クライアント ポリシーではサポートされていません。

IPv4 ACL ネットワーク インターフェイス

次の制限事項が、ネットワーク インターフェイスへの IPv4 ACL に適用されます。

- インターフェイスへのアクセスを制御する場合、名前付き ACL または番号付き ACL を使用できます。
- レイヤ 2 インターフェイスに ACL を適用する場合、ルーティングをイネーブルにする必要はありません。
- レイヤ 3 ポートおよび SVI では、ACL はサポートされていません。

レイヤ 2 インターフェイスの MAC ACL

MAC ACL を作成し、それをレイヤ 2 インターフェイスに適用すると、そのインターフェイスに着信する非 IP トラフィックをフィルタリングできます。MAC ACL を適用するときには、次の注意事項に留意してください。

- 同じレイヤ 2 インターフェイスには、IP アクセス リストと MAC アクセス リストを 1 つずつしか適用できません。IP アクセス リストは IP パケットだけをフィルタリングし、MAC アクセス リストは非 IP パケットをフィルタリングします。
- 1 つのレイヤ 2 インターフェイスに適用できる MAC アドレス リストは 1 つだけです。すでに MAC ACL が設定されているレイヤ 2 インターフェイスに MAC アクセス リストを適用すると、設定済みの ACL が新しい ACL に置き換えられます。

（注） **mac access-group** インターフェイス コンフィギュレーション コマンドは、物理レイヤ 2 インターフェイスに適用される場合のみ有効です。このコマンドは、EtherChannel ポート チャネルでは使用できません。

IP アクセス リスト エントリ シーケンス番号

- この機能は、ダイナミック アクセス リスト、再帰アクセス リスト、またはファイアウォール アクセス リストをサポートしていません。

IPv4 アクセスコントロールリストに関する情報

アクセス コントロール リスト（ACL）は、パケット フィルタリングを実行して、ネットワークを介して移動するパケットとその場所を制御します。このような制御によって、ネットワーク トラフィックを制限し、ユーザおよびデバイスのネットワークに対するアクセスを制限し、トラフィックがネットワークから外部に送信されるのを防ぐことで、セキュリティを実現します。IP アクセスリストによって、スプーフィングやサービス拒否攻撃の可能性を軽減し、ファイアウォールを介したダイナミックで一時的なユーザ アクセスが可能になります。

また、IP アクセス リストは、セキュリティ以外の用途にも使用できます。たとえば、帯域幅制御、デバッグ出力の制限、Quality of Service（QoS）機能のためのトラフィックの識別または分類などです。このモジュールでは、IP アクセス リストの概要について説明します。

ACL の概要

パケット フィルタリングは、ネットワーク トラフィックを限定し、特定のユーザまたはデバイスによるネットワークの使用を制限するうえで役立ちます。ACL はルータまたはスイッチを通過するトラフィックをフィルタリングし、特定のインターフェイスまたは VLAN（仮想 LAN）でパケットを許可、または拒否します。ACL は、パケットに適用される許可条件および拒否条件の順序付けられた集まりです。パケットがインターフェイスに着信すると、スイッチはパケット内のフィールドを適用される ACL と比較し、アクセス リストに指定された基準に基づいて、パケットが転送に必要な権限を持っているかどうかを確認します。アクセス リスト内の条件を 1 つずつ調べ、パケットをテストします。最初に一致した条件によって、スイッチがパケットを受け入れるか拒否するかが決定されます。スイッチは最初に一致した時点でテストを中止するので、リストに条件を指定する順序が重要です。一致する条件がない場合、スイッチはパケットを拒否します。スイッチは、制限条件がない場合はパケットを転送し、制限条件がある場合はパケットをドロップします。スイッチは、VLAN 内でブリッジングされるパケットを含めて、転送されるすべてのパケットに ACL を使用します。

ネットワークに基本的なセキュリティを導入する場合は、ルータにアクセスリストを設定します。ACL を設定しなければ、スイッチを通過するすべてのパケットがネットワークのあらゆる部分で許可される可能性があります。ACL を使用すると、ネットワークの場所ごとにアクセス可能なホストを制御したり、ルータインターフェイスで転送またはブロックされるトラフィックの種類を決定したりできます。たとえば、電子メール トラフィックの転送を許可し、Telnet トラフィックの転送を拒否することもできます。ACL を着信トラフィック、発信トラフィック、またはその両方をブロックするように設定することもできます。

標準 IPv4 ACL および拡張 IPv4 ACL

ここでは、IP ACL について説明します。

ACL は、許可条件と拒否条件の順序付けられた集まりです。スイッチは、アクセス リスト内の条件を 1 つずつ調べ、パケットをテストします。最初に一致した条件によって、スイッチがパケットを受け入れるか拒否するかが決定されます。スイッチは一致する最初の条件が見つかった時点でパケットのテストを停止するため、条件の順序が重要な意味を持ちます。一致する条件がない場合、スイッチはパケットを拒否します。

このソフトウェアは、IPv4 について次の ACL（アクセス リスト）をサポートします。

- 標準 IP アクセス リストでは、照合操作に送信元アドレスを使用します。
- 拡張 IP アクセス リストでは、照合操作に送信元アドレスと宛先アドレスを使用し、任意でプロトコル タイプ情報を使用して制御のきめ細かさを高めることもできます。

IPv4 ACL スイッチでサポートされていない機能

このスイッチで IPv4 ACL を設定する手順は、他の Cisco スイッチやルータで IPv4 ACL を設定する手順と同じです。

以下の ACL 関連の機能はサポートされていません。

- 非 IP プロトコル ACL またはブリッジ グループ ACL
- IP アカウンティング
- 着信および発信レート制限（QoS ACL によるレート制限を除く）
- 再帰 ACL およびダイナミック ACL はサポートされていません。（スイッチのクラスタリング機能で使用される特別なダイナミック ACL を除く）
- VLAN マップの ACL ロギング

アクセス リスト番号

ACL を識別するために使用する番号は、作成するアクセス リストのタイプを表します。

次の一覧に、アクセス リスト番号と対応するアクセス リスト タイプを挙げ、このスイッチでサポートされているかどうかを示します。このスイッチは、IPv4 標準アクセス リストおよび拡張アクセス リスト（1 ～ 199 および 1300 ～ 2699）をサポートします。

表 *18 :* アクセス リスト番号

アクセス リスト番号	タイプ	サポートあり
1 ～ 99	IP 標準アクセス リスト	あり
100 ～ 199	IP 拡張アクセス リスト	あり
200 ～ 299	プロトコル タイプコード アクセス リスト	なし
300 ～ 399	DECnet アクセス リスト	なし
400 ～ 499	XNS 標準アクセス リスト	なし
500 ～ 599	XNS 拡張アクセス リスト	なし
600 ～ 699	AppleTalk アクセス リスト	なし
700 ～ 799	48 ビット MAC アドレス アクセス リスト	なし
800 ～ 899	IPX 標準アクセス リスト	なし
900 ～ 999	IPX 拡張アクセス リスト	なし
1000 ～ 1099	IPX SAP アクセス リスト	なし

アクセス リスト番号	タイプ	サポートあり
1100 ～ 1199	拡張 48 ビット MAC サマリー アドレス アクセス リスト	なし
1200 ～ 1299	IPX サマリー アドレス アクセス リスト	なし
1300 ～ 1999	IP 標準アクセス リスト（拡張範囲）	あり
2000 ～ 2699	IP 拡張アクセス リスト（拡張範囲）	あり

番号付き標準 ACL および番号付き拡張 ACL に加え、サポートされる番号を使用して名前付き標準 ACL および名前付き拡張 ACL も作成できます。標準 IP ACL の名前は 1 ～ 99 で、拡張 IP ACL の名前は 100 ～ 199 です。番号付きリストの代わりに名前付き ACL を使用することには、エントリを個別に削除できるという利点があります。

番号付き標準 IPv4 ACL

ACL を作成するときには、ACL の末尾にデフォルトで暗黙的な deny ステートメントが追加され、ACL の終わりに到達するまで一致する条件が見つからなかったすべてのパケットに適用されることに注意してください。標準アクセス リストでは、関連付けられた IP ホスト アドレス ACL の指定からマスクを省略すると、0.0.0.0 がマスクと見なされます。

スイッチは、**host** 一致条件があるエントリと *don't care* マスク 0.0.0.0 を含む一致条件があるエントリがリストの先頭に移動し、0 以外の *don't care* マスクを含むエントリよりも前に位置するように、標準アクセス リストの順序を書き換えます。そのため、**show** コマンドの出力やコンフィギュレーション ファイルでは、ACE が必ずしも入力されたとおりの順序で配置されません。

番号付き拡張 IPv4 ACL

標準 ACL では照合に送信元アドレスだけを使用しますが、拡張 ACL では、照合操作に送信元アドレスと宛先アドレスを使用でき、任意でプロトコル タイプ情報を使用して制御のきめ細かさを高めることができます。番号付き拡張アクセス リストの ACE を作成するときには、作成した ACE がリストの末尾に追加されることに注意してください。番号付きリストでは、ACE の順序を変更したり、リスト内の特定の場所に対して ACE を追加または削除したりできません。

このスイッチは、ダイナミックまたは再帰アクセスリストをサポートしていません。また、タイプ オブ サービス（ToS）の minimize-monetary-cost ビットに基づくフィルタリングもサポートしていません。

一部のプロトコルには、特定のパラメータやキーワードも適用されます。

拡張 TCP、UDP、ICMP、IGMP、またはその他の IP ACL を定義できます。また、このスイッチはこれらの IP プロトコルをサポートします。

（注） ICMP エコー応答はフィルタリングできません。他の ICMP コードまたはタイプは、すべてフィルタリングできます。

これらの IP プロトコルがサポートされます。

- 認証ヘッダー プロトコル（**ahp**）
- カプセル化セキュリティペイロード（**esp**）
- Enhanced Interior Gateway Routing Protocol（**eigrp**）
- 総称ルーティング カプセル化（**gre**）
- インターネット制御メッセージ プロトコル（**icmp**）
- インターネット グループ管理プロトコル（**igmp**）
- すべての内部プロトコル（**ip**）
- IP-in-IP トンネリング（**ipinip**）
- KA9Q NOS 互換 IP over IP トンネリング（**nos**）
- Open Shortest Path First ルーティング（**ospf**）
- ペイロード圧縮プロトコル（**pcp**）
- プロトコル独立マルチキャスト（**pim**）
- 伝送制御プロトコル（**tcp**）
- ユーザ データグラム プロトコル（**udp**）

名前付き IPv4 ACL

IPv4 ACL を識別する手段として、番号ではなく英数字のストリング（名前）を使用できます。名前付き ACL を使用すると、ルータ上で番号付きアクセス リストの場合より多くの IPv4 アクセス リストを設定できます。アクセス リストの識別手段として名前を使用する場合のモードとコマンド構文は、番号を使用する場合とは多少異なります。ただし、IP アクセス リストを使用するすべてのコマンドを名前付きアクセス リストで使用できるわけではありません。

（注） 標準 ACL または拡張 ACL に指定する名前は、アクセス リスト番号のサポートされる範囲内の番号にすることもできます。つまり、標準の IP ACL の名前は 1～99 を指定できます。番号付きリストの代わりに名前付き ACL を使用することには、エントリを個別に削除できるという利点があります。

名前付き ACL を設定するときには、次の注意事項に留意してください。

- また、番号付き ACL も使用できます。

- 標準 ACL と拡張 ACL に同じ名前は使用できません。

アクセス コントロール エントリ機能での非隣接ポートに関する名前付き ACL サポートを使用する利点

アクセス コントロール エントリ機能での非隣接ポートに関する名前付き ACL サポートを使用すると、1 つのアクセス コントロール エントリで非隣接ポートを指定できるため、複数のエントリが同じ送信元アドレス、宛先アドレス、およびプロトコルを持ち、ポートのみが異なる場合に、アクセス コントロール リストで必要なエントリ数を大幅に減らすことができます。

この機能によって、同じ送信元アドレス、宛先アドレス、およびプロトコルに関して複数のエントリを処理するために、アクセス コントロール リストで必要なアクセス コントロール エントリ（ACE）の数が大幅に削減されます。大量の ACE を保守している場合、可能な限り、新しいアクセス リスト エントリを作成するときは、この機能を使用して既存のアクセス リスト エントリのグループを統合します。非隣接ポートを使用するアクセス リスト エントリを設定すると、保守するアクセス リスト エントリ数が少なくなります。

IP アクセス リスト エントリ シーケンス番号の利点

IP アクセス リスト エントリにシーケンス番号を適用する機能によって、アクセス リストの変更が簡易になります。IP アクセス リスト エントリ シーケンス番号機能の前には、アクセス リスト内のエントリの位置を指定する方法はありませんでした。以前は、既存のリストの途中にエントリ（ステートメント）を挿入する場合、目的の位置の後にあるすべてのエントリを削除してから、新しいエントリを追加し、削除したすべてのエントリを再入力する必要がありました。これは手間がかかり、エラーが起こりやすい方法です。

この新しい機能を使用すると、アクセス リスト エントリにシーケンス番号を追加し、順序を変更することができます。新しいエントリを追加するとき、アクセス リストの目的の位置に配置されるように、シーケンス番号を選択します。必要に応じて、アクセス リストの現在のエントリを並べ替えて、新しいエントリを挿入できる場所を作成できます。

シーケンス番号の動作

- 以前のリリースとの下位互換性を保つため、シーケンス番号のないエントリが適用された場合には、最初のエントリにはシーケンス番号 10 が割り当てられます。連続してエントリを追加すると、シーケンス番号は10ずつ増分されます。最大シーケンス番号は2147483647です。生成したシーケンス番号がこの最大値を超えると、次のメッセージが表示されます。

```
Exceeded maximum sequence number.
```

- シーケンス番号のないエントリを入力すると、アクセス リストの最後のシーケンス番号に 10 を加えたシーケンス番号が割り当てられ、リストの末尾に配置されます。
- （シーケンス番号以外が）既存のエントリに一致するエントリを入力すると、何も変更されません。
- 既存のシーケンス番号を入力すると、次のエラー メッセージが表示されます。

```
Duplicate sequence number.
```

- グローバル コンフィギュレーション モードで新しいアクセス リストを入力すると、そのアクセス リストのシーケンス番号が自動的に生成されます。
- 分散サポートが提供されます。ルート プロセッサ（RP）とライン カードにあるエントリのシーケンス番号は、常に同期されます。
- シーケンス番号が不揮発性生成（NVGEN）されることはありません。つまり、シーケンス番号自体は保存されません。システムのリロード時には、設定されたシーケンス番号はデフォルトのシーケンス開始番号と増分に戻されます。この機能は、シーケンス番号をサポートしないソフトウェア リリースとの下位互換性を保つために提供されています。
- この機能は、名前付きおよび番号付きの標準および拡張IPアクセス リストと連動します。

ACL へのコメントの挿入

remark キーワードを使用すると、任意の IP 標準または拡張 ACL にエントリに関するコメント（注釈）を追加できます。コメントを使用すると、ACL の理解とスキャンが容易になります。1 つのコメント行の最大長は 100 文字です。

コメントは、permit ステートメントまたは deny ステートメントの前後どちらにでも配置できます。コメントがどの permit ステートメントまたは deny ステートメントの説明であるのかが明確になるように、コメントの位置に関して一貫性を保つ必要があります。たとえば、あるコメントは対応する permit または deny ステートメントの前にあり、他のコメントは対応するステートメントの後ろにあると、混乱を招く可能性があります。

番号付き IP 標準または拡張 ACL にコメントを挿入するには、**access-list** *access-list number* **remark** *remark* 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。コメントを削除するには、このコマンドの **no** 形式を使用します。

後続の deny ステートメントの機能を説明する注釈の例を次に示します。

```
ip access-list extended telnetting
  remark Do not allow host1 subnet to telnet out
  deny tcp host 172.16.2.88 any eq telnet
```

ハードウェアおよびソフトウェアによる IP ACL の処理

ACL の処理は、ハードウェア側で実行されます。ハードウェアで ACL の設定を保存する容量が不足すると、パケットは CPU に送られ、ACL の処理はソフトウェア側で行われます。ACL をソフトウェアで処理するためにデータ パケットが転送される場合、転送速度はレート制限により、ライン レートよりもかなり低下します。

（注） スイッチのリソース不足が原因でハードウェアに ACL を設定できない場合、影響を受けるのは、スイッチに着信した該当 VLAN 内のトラフィックだけです。パケットのソフトウェア転送が発生すると、消費される CPU サイクル数に応じて、スイッチのパフォーマンスが低下することがあります。

トラフィック フローのロギングと転送の両方を行う場合、転送はハードウェアで処理されますが、ロギングはソフトウェアで処理される必要があります。ハードウェアとソフトウェアではパケット処理能力が異なるため、ロギング中であるすべてのフロー（許可フローと拒否フロー）の合計帯域幅が非常に大きい場合は、転送されたパケットの一部をロギングできません。

show ip access-lists 特権 EXEC コマンドを入力した場合、表示される一致カウントには、ハードウェアでアクセスが制御されるパケットは含まれません。ACL の機能は、次のとおりです。

- 標準 ACL および拡張 ACL（入力および出力）の許可アクションや拒否アクションをハードウェアで制御し、アクセス コントロールのセキュリティを強化します。
- *ip unreachables* が無効の場合、**log**を指定しないと、セキュリティ ACL の *deny* ステートメントと一致するフローがハードウェアによってドロップされます。許可ステートメントと一致するフローは、ハードウェアでスイッチングされます。
- ACL の ACE に **log** キーワードを追加すると、パケットのコピーが CPU に送信され、ロギングだけが行われます。ACE が *permit* ステートメントの場合も、パケットはハードウェアでスイッチングされます。

ACL の時間範囲

time-range グローバル コンフィギュレーション コマンドを使用することによって、時刻および曜日に基づいて拡張 ACL を選択的に適用できます。まず、時間範囲の名前を定義し、その時間範囲内の時刻および日付または曜日を設定します。次に、ACL を適用してアクセス リストに制限を設定するときに時間範囲を入力します。時間範囲を使用すると、ACL の許可ステートメントまたは拒否ステートメントの有効期間（指定期間内や指定曜日など）を定義できます。**time-range** キーワードおよび引数については、名前付きおよび番号付き拡張 ACL タスクの表を参照してください。

時間範囲を使用するいくつかの利点を次に示します。

- アプリケーションなどのリソース（IP アドレスとマスクのペア、およびポート番号で識別）へのユーザ アクセスをより厳密に許可または拒否できます。
- ログ メッセージを制御できます。ACL エントリを使用して特定の時刻に関してのみトラフィックをロギングできるため、ピーク時間に生成される多数のログを分析しなくても、簡単にアクセスを拒否できます。

時間ベースのアクセス リストを使用すると、CPU に負荷が生じます。これは、アクセス リストの新規設定を他の機能や、ハードウェア メモリにロードされた結合済みの設定とマージする

必要があるためです。そのため、複数のアクセス リストが短期間に連続して（互いに数分以内に）有効となるような設定とならないように注意する必要があります。

（注） 時間範囲は、スイッチのシステム クロックに基づきます。したがって、信頼できるクロック ソースが必要です。ネットワーク タイム プロトコル（NTP）を使用してスイッチ クロックを同期させることを推奨します。

IPv4 ACL のインターフェイスに関する注意事項

インバウンド ACL の場合、パケットの受信後スイッチはパケットを ACL と照合します。ACL がパケットを許可する場合、スイッチはパケットの処理を継続します。ACL がパケットを拒否する場合、スイッチはパケットを廃棄します。

未定義の ACL をインターフェイスに適用すると、スイッチは ACL がインターフェイスに適用されていないと判断し、すべてのパケットを許可します。ネットワーク セキュリティのために未定義の ACL を使用する場合は、このような結果が生じることに注意してください。

インターフェイスへのアクセス コントロール リストの適用

インバウンドアクセスリストの場合、デバイスがパケットを受信すると、シスコソフトウェアはアクセスリストの条件ステートメントをチェックして一致がないか確認します。パケットが許可されると、ソフトウェアはパケットの処理を継続します。パケットが拒否されると、ソフトウェアはパケットを廃棄します。

（注） デバイスのインターフェイスに適用されるアクセスリストは、そのデバイスから送信されたトラフィックにはフィルタ処理を行いません。

ACL ロギング

標準 IP アクセス リストによって許可または拒否されたパケットに関するログ メッセージが、スイッチのソフトウェアによって表示されます。つまり、ACL と一致するパケットがあった場合は、そのパケットに関するログ通知メッセージがコンソールに送信されます。コンソールに表示されるメッセージのレベルは、syslog メッセージを管理する **logging console** コマンドで管理されます。

（注） ルーティングはハードウェアで、ロギングはソフトウェアで実行されます。したがって、**log** キーワードを含む許可（*permit*）または拒否（*deny*）ACE と一致するパケットが多数存在する場合、ソフトウェアはハードウェアの処理速度に追いつくことができないため、一部のパケットはロギングされない場合があります。

ACL を起動した最初のパケットについては、ログ メッセージがすぐに表示されますが、それ以降のパケットについては、5 分間の収集時間が経過してから表示またはロギングされます。ログ メッセージにはアクセス リスト番号、パケットの許可または拒否に関する状況、パケットの送信元 IP アドレス、および直前の 5 分間に許可または拒否された送信元からのパケット数が示されます。

（注） ロギング メッセージが多すぎて処理できない場合、または1秒以内に処理する必要があるロギング メッセージが複数ある場合、ロギング設備ではロギング メッセージ パケットの一部をドロップすることがあります。この動作によって、ロギング パケットが多すぎてルータがクラッシュすることを回避します。そのため、課金ツールや、アクセス リストと一致する数の正確な情報源としてロギング設備をを使用しないでください。

ACL の設定方法

この項では、ACL の設定方法について説明します。

IPv4 ACL の設定

スイッチで IP ACL を使用するには、次の手順に従います。

手順

ステップ 1 アクセス リストの番号または名前とアクセス条件を指定して、ACL を作成します。

ステップ 2 ACL をインターフェイスに適用します。

番号付き標準 ACL の作成（CLI）

番号付き標準 ACL を作成するには、次の手順に従います。

手順

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

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**access-list** *access-list-number* {**deny** | **permit**} *source source-wildcard* [**log**] 例： `Device(config)# access-list 2 deny your_host`	送信元アドレスとワイルドカードを使用して標準 IPv4 アクセス リストを定義します。 *access-list-number* には、1 ～ 99 または 1300 ～ 1999 の 10 進数を指定します。 条件が一致した場合にアクセスを拒否する場合は **deny** を指定し、許可する場合は **permit** を指定します。 *source* には、パケットの送信元となるネットワークまたはホストのアドレスを次の形式で指定します。 • ドット付き 10 進表記による 32 ビット長の値。 • キーワード **any** は 0.0.0.0 255.255.255.255 という *source* および *source-wildcard* の省略形です。source-wildcard を入力する必要はありません。 • キーワード **host** は送信元および *source* 0.0.0.0 の *source-wildcard* の省略形です。 （任意）*source-wildcard* は、ワイルドカード ビットを送信元アドレスに適用します。 （任意）**log** を指定すると、エントリと一致するパケットに関するログ通知メッセージがコンソールに送信されます。 （注） ロギングは、レイヤ 3 インターフェイスに割り当てられた ACL でだけサポートされます。

	コマンドまたはアクション	目的
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

番号付き拡張 ACL の作成（CLI）

番号付き拡張 ACL を作成するには、次の手順に従います。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** tos] [**fragments**] [**time-range** *time-range-name*] [**dscp** *dscp*] 例：	拡張 IPv4 アクセス リストおよびアクセス条件を定義します。 *access-list-number* には、100 ～ 199 または 2000 ～ 2699 の 10 進数を指定します。

	コマンドまたはアクション	目的
	`Device(config)# access-list 101 permit ip host 10.1.1.2 any precedence 0 tos 0 log`	条件が一致した場合にパケットを拒否する場合は **deny** を指定し、許可する場合は **permit** を指定します。 *protocol* には、インターネットプロトコルの名前または番号を入力します。**ahp**、**eigrp**、**esp**、**gre**、**icmp**、**igmp**、**igrp**、**ip**、**ipinip**、**nos**、**ospf**、**pcp**、**pim**、**tcp**、**udp**、または IP プロトコル番号を表す 0 ～ 255 の整数を使用できます。一致条件としてインターネット プロトコル（ICMP、TCP、UDP など）を指定するには、キーワード **ip** を使用します。 （注） この手順には、ほとんどの IP プロトコルのオプションが含まれています。TCP、UDP、ICMP、および IGMP の追加の特定パラメータについては、次のステップを参照してください。 *source* には、パラメータの送信元であるネットワークまたはホストの番号を指定します。 *source-wildcard* は、ワイルドカードビットを送信元アドレスに適用します。 *destination* には、パラメータの宛先であるネットワークまたはホストの番号を指定します。 *destination-wildcard* は、ワイルドカードビットを宛先アドレスに適用します。 source、source-wildcard、destination、および destination-wildcard の値は、次の形式で指定します。 • ドット付き 10 進表記による 32 ビット長の値。 • 0.0.0.0 255.255.255.255（任意のホスト）を表すキーワード **any**。 • 単一のホスト 0.0.0.0 を表すキーワード **host**。

	コマンドまたはアクション	目的
		その他のキーワードはオプションであり、次の意味を持ちます。 • **precedence**：パケットを 0 ～ 7 の番号または名前で指定する優先度と一致させる場合に入力します。指定できる値は、**routine**（0）、**priority**（1）、**immediate**（2）、**flash**（3）、**flash-override**（4）、**critical**（5）、**internet**（6）、**network**（7）です。 • **fragments**：2 つ目以降のフラグメントをチェックする場合に入力します。 • **tos**：パケットを 0 ～ 15 の番号または名前で指定するサービス タイプ レベルと一致させる場合に入力します。指定できる値は、**normal**（0）、**max-reliability**（2）、**max-throughput**（4）、**min-delay**（8）です。 • **time-range**：時間範囲の名前を指定します。 • **dscp**：パケットを 0 ～ 63 の番号で指定する DSCP 値と一致させる場合に入力します。または、指定できる値のリストを表示するには、疑問符（?）を使用します。 （注） **dscp** 値を入力する場合は、**tos** または **precedence** を入力できません。**dscp** を入力せずに **tos** と **precedence** の両方の値を入力できます。
ステップ 4	**access-list** *access-list-number* {**deny** | **permit**} **tcp** *source source-wildcard* [*operator port*] *destination destination-wildcard* [*operator port*] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**time-range** *time-range-name*] [**dscp** *dscp*] [*flag*]	拡張 TCP アクセス リストおよびアクセス条件を定義します。 次に示す例外を除き、拡張 IPv4 ACL に対して説明するパラメータと同じパラメータを使用します。

	コマンドまたはアクション	目的
	例： `Device(config)# access-list 101 permit tcp any any eq 500`	（任意）*operator* および *port* を入力すると、送信元ポート（*source source-wildcard* の後に入力した場合）または宛先ポート（*destination destination-wildcard* の後に入力した場合）が比較されます。演算子の候補には、**eq**（次の値に等しい）、**gt**（次の値より大きい）、**lt**（次の値より小さい）、**neq**（次の値に等しくない）、および **range**（次の範囲）があります。演算子にはポート番号を指定する必要があります（range の場合は 2 つのポート番号をスペースで区切って指定する必要があります）。 *port* には、10 進数（0 ～ 65535）のポート番号または TCP ポート名を入力します。TCP をフィルタリングするときには、TCP ポートの番号または名前だけを使用します。 他のオプションのキーワードの意味は次のとおりです。 • *flag*：指定された TCP ヘッダー ビットを基準にして照合します。入力できるフラグは、**ack**（確認応答）、**fin**（終了）、**psh**（プッシュ）、**rst**（リセット）、**syn**（同期）、または **urg**（緊急）です。
ステップ 5	**access-list** *access-list-number* {**deny** | **permit**} **udp** *source source-wildcard* [*operator port*] *destination destination-wildcard* [*operator port*] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**time-range** *time-range-name*] [**dscp** *dscp*] 例： `Device(config)# access-list 101 permit udp any any eq 100`	（任意）拡張 UDP アクセス リストおよびアクセス条件を定義します。 UDP パラメータは TCP の説明にあるパラメータと同じです。ただし、[operator [port]] ポートの番号または名前は、UDP ポートの番号または名前でなければなりません。また、UDP では、**flag** キーワードは無効です。
ステップ 6	**access-list** *access-list-number* {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* [*icmp-type*	拡張 ICMP アクセス リストおよびアクセス条件を定義します。

	コマンドまたはアクション	目的
	| [[*icmp-type icmp-code*] | [*icmp-message*]] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**time-range** *time-range-name*] [**dscp** *dscp*] 例： `Device(config)# access-list 101 permit icmp any any 200`	ICMP パラメータは拡張 IPv4 ACL の IP プロトコルの説明にあるパラメータとほとんど同じですが、ICMP メッセージタイプおよびコード パラメータが追加されています。オプションのキーワードの意味は次のとおりです。 • *icmp-type*：ICMP メッセージ タイプでフィルタリングする場合に入力します。指定できる値の範囲は、0～255 です。 • *icmp-code*：ICMP パケットを ICMP メッセージ コード タイプでフィルタリングする場合に入力します。指定できる値の範囲は、0 ～ 255 です。 • *icmp-message*：ICMP パケットを ICMP メッセージ タイプ名または ICMP メッセージタイプとコード名でフィルタリングする場合に入力します。
ステップ 7	**access-list** *access-list-number* {**deny** | **permit**} **igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**time-range** *time-range-name*] [**dscp** *dscp*] 例： `Device(config)# access-list 101 permit igmp any any 14`	（任意）拡張 IGMP アクセス リストおよびアクセス条件を定義します。 IGMP パラメータは拡張 IPv4 ACL の IP プロトコルの説明にあるパラメータとほとんど同じですが、次に示すオプションのパラメータが追加されています。 *igmp-type*IGMP メッセージ タイプと比較するには、0～15 の番号またはメッセージ名（**dvmrp**、**host-query**、**host-report**、**pim**、または **trace**）を入力します。
ステップ 8	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

名前付き標準 ACL の作成

名前を使用して標準 ACL を作成するには、次の手順に従います。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip access-list standard** *name* 例： `Device(config)# ip access-list standard 20`	名前を使用して標準 IPv4 アクセス リストを定義し、アクセス リスト コンフィギュレーション モードを開始します。 名前には、1 ～ 99 の番号を使用できます。

ステップ 4	次のいずれかを使用します。
	• **deny** {*source* [*source-wildcard*] | **host** *source* | **any**} [**log**]
	• **permit** {*source* [*source-wildcard*] | **host** *source* | **any**} [**log**]
	例：
	`Device(config-std-nacl)# deny 192.168.0.0 0.0.255.255 255.255.0.0 0.0.255.255`
	または
	`Device(config-std-nacl)# permit 10.108.0.0 0.0.0.0 255.255.255.0 0.0.0.0`
		アクセス リスト コンフィギュレーション モードで、パケットを転送するのかドロップするのかを決定する 1 つ以上の拒否条件または許可条件を指定します。
		• **host** *source*：送信元および送信元ワイルドカードの値である *source* 0.0.0.0。
		• **any**：送信元および送信元ワイルドカードの値である 0.0.0.0 255.255.255.255

ステップ 5	**end** 例： `Device(config-std-nacl)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例：	入力を確認します。

	コマンドまたはアクション	目的
	`Device# show running-config`	
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

名前付き拡張 ACL の作成

名前を使用して拡張 ACL を作成するには、次の手順に従います。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip access-list extended** *name* 例： `Device(config)# ip access-list extended 150`	名前を使用して拡張 IPv4 アクセス リストを定義し、アクセス リスト コンフィギュレーション モードを開始します。 名前には、100 ～ 199 の番号を使用できます。
ステップ 4	{**deny** | **permit**} *protocol* {*source* [*source-wildcard*] | **host** *source* | **any**} {*destination* [*destination-wildcard*] | host *destination* | **any**} [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] 例： `Device(config-ext-nacl)# permit 0 any`	アクセス リスト コンフィギュレーション モードで、許可条件または拒否条件を指定します。**log** キーワードを使用すると、違反を含むアクセス リストのログ メッセージを取得できます。 • **host** *source*：送信元および送信元ワイルドカードの値である *source* 0.0.0.0。

	コマンドまたはアクション	目的
	any	• **host** *destination*：接続先および接続先ワイルドカードの値である *destination* 0.0.0.0。 • **any**：source および source wildcard の値または destination および destination wildcard の値である 0.0.0.0 255.255.255.255
ステップ 5	**end** 例： `Device(config-ext-nacl)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

拡張 ACL を作成するときには、ACL の末尾にデフォルトで暗黙的な deny ステートメントが追加され、ACL の終わりに到達するまで一致する条件が見つからなかったすべてのパケットに適用されることに注意してください。標準 ACL では、関連付けられた IP ホスト アドレス アクセス リストの指定からマスクを省略すると、0.0.0.0 がマスクと見なされます。

ACL の作成後に追加したエントリは、リストの末尾に追加されます。ACL エントリを特定の ACL に選択的に追加できません。ただし、**no permit** および **no deny** アクセスリスト コンフィギュレーション モード コマンドを使用すると、名前付き ACL からエントリを削除できます。

番号付き ACL ではなく名前付き ACL を使用する理由の 1 つとして、名前付き ACL では行を選択して削除できることがあります。

次のタスク

作成した名前付き ACL をインターフェイスに適用できます。

アクセス リスト エントリの順序付けとアクセス リストの変更

ここでは、名前付き IP アクセス リストのエントリにシーケンス番号を割り当てる方法と、アクセス リストに対するエントリの追加または削除を行う方法を説明します。この作業を実行する場合は、次の点に注意してください。

- アクセス リスト エントリの並べ替えは任意です。この作業での並べ替えのステップは、機能の目的の 1 つであり、またその機能の説明が必要と思われることから、必要に応じて説明します。
- 次の手順で、**permit** コマンドはステップ 5 に、**deny** コマンドはステップ 6 に記載されています。ただし、その順番を入れ替えることもできます。設定のニーズに合わせた順番を使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip access-list resequence** *access-list-name starting-sequence-number increment* 例： `Device(config)# ip access-list resequence kmd1 100 15`	開始シーケンス番号と、シーケンス番号の増分を使用して、指定した IP アクセス リストを並べ替えます。
ステップ 4	**ip access-list** {**standard**| **extended**} *access-list-name* 例： `Device(config)# ip access-list standard kmd1`	名前で IP アクセス リストを指定し、名前付きアクセス リストのコンフィギュレーション モードを開始します。 • **standard** を指定する場合は、その後に、標準アクセス リスト構文を使用して **permit** ステートメントまたは **deny** ステートメントを指定します。 • **extended** を指定する場合は、その後に、拡張アクセス リスト構文を使用して **permit** ステートメントま

	コマンドまたはアクション	目的
		たは **deny** ステートメントを指定します。
ステップ 5	次のいずれかを実行します。 • *sequence-number* **permit** *source source-wildcard* • *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] 例： `Device(config-std-nacl)# 105 permit 10.5.5.5 0.0.0 255`	名前付き IP アクセス リスト モードで permit ステートメントを指定します。 • このアクセス リストでは **permit** ステートメントを最初に使用していますが、必要なステートメントの順序に応じて、**deny** ステートメントが最初に使用される可能性もあります。 • プロンプトに示されるとおり、このアクセス リストは標準アクセス リストでした。ステップ 4 で **extended** を指定した場合は、このステップのプロンプトは Device(config-ext-nacl) となり、拡張 **permit** コマンドシンタックスを使用します。
ステップ 6	次のいずれかを実行します。 • *sequence-number* **deny** *source source-wildcard* • *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] 例： `Device(config-std-nacl)# 105 deny 10.6.6.7 0.0.0 255`	（任意）名前付き IP アクセス リスト モードで deny ステートメントを指定します。 • このアクセス リストでは **permit** ステートメントを最初に使用していますが、必要なステートメントの順序に応じて、**deny** ステートメントが最初に使用される可能性もあります。 • プロンプトに示されるとおり、このアクセス リストは標準アクセス リストでした。ステップ 4 で **extended** を指定した場合は、このステップのプロンプトは Device(config-ext-nacl) となり、拡張 **deny** コマンドシンタックスを使用します。
ステップ 7	次のいずれかを実行します。 • *sequence-number* **permit** *source source-wildcard*	名前付き IP アクセス リスト モードで permit ステートメントを指定します。

	コマンドまたはアクション	目的
	• *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] 例： `Device(config-ext-nacl)# 150 permit tcp any any log`	• このアクセス リストでは **permit** ステートメントを最初に使用していますが、必要なステートメントの順序に応じて、**deny** ステートメントが最初に使用される可能性もあります。 • 上位層プロトコル（ICMP、IGMP、TCP、およびUDP）を許可するその他のコマンドシンタックスについては、permit（IP）コマンドを参照してください。 • エントリを削除するには、**no** *sequence-number* コマンドを使用します。
ステップ 8	次のいずれかを実行します。 • *sequence-number* **deny** *source source-wildcard* • *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] 例： `Device(config-ext-nacl)# 150 deny tcp any any log`	（任意）名前付き IP アクセス リスト モードで deny ステートメントを指定します。 • このアクセス リストでは **permit** ステートメントを最初に使用していますが、必要なステートメントの順序に応じて、**deny** ステートメントが最初に使用される可能性もあります。 • 上位層プロトコル（ICMP、IGMP、TCP、およびUDP）を許可するその他のコマンドシンタックスについては、deny（IP）コマンドを参照してください。 • エントリを削除するには、**no** *sequence-number* コマンドを使用します。
ステップ 9	必要に応じてシーケンス番号ステートメントを追加するには、ステップ 5 とステップ 6 を繰り返します。	アクセス リストは変更できます。
ステップ 10	**end** 例： `Device(config-std-nacl)# end`	（任意）コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 11	**show ip access-lists** *access-list-name* 例： `Device# show ip access-lists kmd1`	（任意）IPアクセスリストの内容を表示します。

例

アクセス リストに新しいエントリが含まれていることを確認するには、**show ip access-lists** コマンドの出力を確認します。

```
Device# show ip access-lists kmd1

Standard IP access list kmd1
100 permit 10.4.4.0, wildcard bits 0.0.0.255
105 permit 10.5.5.0, wildcard bits 0.0.0.255
115 permit 10.0.0.0, wildcard bits 0.0.0.255
130 permit 10.5.5.0, wildcard bits 0.0.0.255
145 permit 10.0.0.0, wildcard bits 0.0.0.255
```

コメント付き IP ACL エントリの設定

名前付きまたは番号付きアクセス リスト設定を使用します。作業する設定用にアクセス リストを作成したら、アクセスリストをインターフェイスまたは端末回線に適用する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip access-list** {**standard** | **extended**} {*name* | *number*} 例： `Device(config)# ip access-list extended telnetting`	名前または番号でアクセス リストを特定し、拡張名前付きアクセス リスト コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**remark** *remark* 例： `Device(config-ext-nacl)# remark Do not allow host1 subnet to telnet out`	名前付き IP アクセス リストのエントリに注釈を追加します。 • 注釈は、**permit** または **deny** ステートメントの目的を示します。
ステップ 5	**deny** *protocol* **host** *host-address* **any eq** *port* 例： `Device(config-ext-nacl)# deny tcp host 172.16.2.88 any eq telnet`	パケットを拒否する名前付き IP アクセス リストの条件を設定します。
ステップ 6	**end** 例： `Device(config-ext-nacl)# end`	拡張名前付きアクセス リスト コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。

ACL の時間範囲の設定

ACL の時間範囲パラメータを設定するには、次の手順に従ってください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device(config)# enable`	特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**time-range** *time-range-name* 例： `Device(config)# time-range workhours`	作成する時間範囲には意味のある名前（*workhours* など）を割り当て、時間範囲コンフィギュレーション モードを開始します。名前にスペースや疑問符を含めることはできません。また、文字から始める必要があります。
ステップ 4	次のいずれかを使用します。 • **absolute** [**start** *time date*] [**end** *time date*]	適用対象の機能がいつ動作可能になるかを指定します。

	コマンドまたはアクション	目的

	• **periodic** *day-of-the-week hh:mm to* [*day-of-the-week*] *hh:mm*
	• **periodic** {**weekdays** | **weekend** | **daily**} *hh:mm to hh:mm*
	例：
	`Device(config-time-range)# absolute start 00:00 1 Jan 2006 end 23:59 1 Jan 2006`
	または
	`Device(config-time-range)# periodic weekdays 8:00 to 12:00`
		• 時間範囲には、**absolute** ステートメントを 1 つだけ使用できます。複数の absolute ステートメントを設定した場合は、最後に設定したステートメントだけが実行されます。
		• 複数の **periodic** ステートメントを入力できます。たとえば、平日と週末に異なる時間を設定できます。
		設定例を参照してください。

ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

複数の項目をそれぞれ異なる時間に有効にする場合は、上記の手順を繰り返してください。

端末回線への IPv4 ACL の適用

番号付き ACL を使用して、1 つまたは複数の端末回線へのアクセスを制御できます。端末回線には名前付き ACL を適用できません。すべての仮想端末回線にユーザが接続する可能性があるため、すべてに同じ制限を設定する必要があります。

仮想端末回線と ACL に指定されたアドレス間の着信接続および発信接続を制限するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device(config)# enable`	特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**line** [**console** | **vty**] *line-number* 例： `Devices(config)# line console 0`	設定する回線を指定し、インライン コンフィギュレーション モードを開始します。 • **console**：コンソール端末回線を指定します。コンソール ポートは DCE です。 • **vty**：リモート コンソール アクセス用の仮想端末を指定します。 *line-number* は、回線タイプを指定する場合に、設定する連続グループ内で最初の回線番号です。指定できる範囲は 0 ～ 16 です。
ステップ 4	**access-class** *access-list-number* **in** 例： `Device(config-line)# access-class 10 in`	（デバイスへの）特定の仮想端末回線とアクセスリストに指定されたアドレス間の着信接続を制限します。
ステップ 5	**end** 例： `Device(config-line)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。

	コマンドまたはアクション	目的
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

インターフェイスへの IPv4 ACL の適用（CLI）

ここでは、IPv4 ACL をネットワーク インターフェイスへ適用する方法について説明します。

インターフェイスへのアクセスを制御するには、特権 EXEC モードで次の手順に従います。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	設定するインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。 インターフェイスには、レイヤ 2 インターフェイス（ポート ACL）を指定できます。
ステップ 3	**ip access-group** {*access-list-number* | *name*} {**in**} 例： `Device(config-if)# ip access-group 2 in`	指定されたインターフェイスへのアクセスを制御します。
ステップ 4	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例：	アクセス リストの設定を表示します。

	コマンドまたはアクション	目的
	`Device# show running-config`	
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IPv4 ACL のモニタリング

スイッチに設定されている ACL、およびインターフェイスに適用済みの ACL を表示することで、IPv4 ACL をモニタできます。

ip access-group インターフェイス コンフィギュレーション コマンドを使用して、レイヤ 2 またはレイヤ 3 インターフェイスに ACL を適用した場合は、そのインターフェイスのアクセス グループを表示できます。また、レイヤ 2 インターフェイスに適用された MAC ACL も表示できます。この情報を表示するには、次の表に記載された特権 EXEC コマンドを使用します。

表 *19*：アクセス リストおよびアクセス グループを表示するコマンド

コマンド	目的
show access-lists [*number* | *name*]	最新の IP および MAC アドレス アクセス リストの全体やその一部、または特定のアクセス リスト（番号付きまたは名前付き）の内容を表示します。
show ip access-lists [*number* | *name*]	最新の IP アクセス リスト全体、または特定の IP アクセス リスト（番号付きまたは名前付き）を表示します。
show ip interface *interface-id*	インターフェイスの詳細設定およびステータスを表示します。IP がイネーブルになっているインターフェイスに、**ip access-group** インターフェイス コンフィギュレーション コマンドを使用して ACL を適用した場合は、アクセス グループも表示に含まれます。
show running-config [**interface** *interface-id*]	スイッチまたは指定されたインターフェイスのコンフィギュレーション ファイルの内容（設定されたすべての MAC および IP アクセス リストや、どのアクセス グループがインターフェイスに適用されたかなど）を表示します。

コマンド	目的
show mac access-group [**interface** *interface-id*]	すべてのレイヤ 2 インターフェイスまたは指定されたレイヤ 2 インターフェイスに適用されている MAC アクセス リスト を表示します。

ACL の設定例

この項では、IPv4 ACL の設定例を示します。

例：番号付き ACL

次の例のネットワーク 10.0.0.0 は、2 番目のオクテットがサブネットを指定するクラス A ネットワークです。つまり、サブネットマスクは255.255.0.0 です。ネットワーク 10.0.0.0 アドレスの 3 番目および 4 番目のオクテットで特定のホストを指定します。アクセス リスト 2 を使用して、サブネット 48 のアドレスを 1 つ許可し、同じサブネットの他のアドレスはすべて拒否します。このアクセス リストの最終行は、ネットワーク 10.0.0.0 の他のすべてのサブネット上のアドレスが許可されることを示します。この ACL は、ポートに着信するパケットに適用されます。

```
Device(config)# access-list 2 permit 10.48.0.3
Device(config)# access-list 2 deny 10.48.0.0 0.0.255.255
Device(config)# access-list 2 permit 10.0.0.0 0.255.255.255
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# ip access-group 2 in
```

例：拡張 ACL

次の例の先頭行は、1023 よりも大きい宛先ポートへの着信 TCP 接続を許可します。2 番目の行は、ホスト 128.88.1.2 のシンプルメール転送プロトコル（SMTP）ポートへの着信 TCP 接続を許可します。3 番めの行は、エラー フィードバック用の着信 ICMP メッセージを許可します。

```
Device(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 gt 1023
Device(config)# access-list 102 permit tcp any host 128.88.1.2 eq 25
Device(config)# access-list 102 permit icmp any any
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# ip access-group 102 in
```

次の例では、インターネットに接続されたネットワークがあり、そのネットワーク上の任意のホストがインターネット上の任意のホストと TCP 接続を確立できるようにする場合を想定しています。ただし、IP ホストからは、専用メール ホストのメール（SMTP）ポートを除き、ネットワーク上のホストと TCP 接続を確立できないようにします。

SMTP は、接続の一端では TCP ポート 25、もう一端ではランダムなポート番号を使用します。接続している間は、同じポート番号が使用されます。インターネットから着信するメール パケットの宛先ポートは 25 です。安全なネットワークシステムは、ポート 25 で常にメール接続を受け入れます。

```
Device(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 eq 23
Device(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 eq 25
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# ip access-group 102 in
```

次の例では、ネットワークはアドレスが 128.88.0.0 のクラス B ネットワークで、メール ホストのアドレスは 128.88.1.2 です。**ACK** または **RST** キーワードを使用して、ACK または RST ビット セットを照合します。これで、パケットが既存の接続に属していることが判明します。

```
Device(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 RST
Device(config)# access-list 102 permit tcp any host 128.88.1.2 eq 25
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# ip access-group 102 in
```

例：名前付き ACL

名前付き標準 ACL および名前付き拡張 ACL の作成

次に、*Internet_filter* という名前の標準 ACL および *marketing_group* という名前の拡張 ACL を作成する例を示します。*Internet_filter* ACL は、送信元アドレス 1.2.3.4 から送信されるすべてのトラフィックを許可します。

```
Device(config)# ip access-list standard Internet_filter
Device(config-ext-nacl)# permit 1.2.3.4
Device(config-ext-nacl)# exit
```

marketing_group ACL は、宛先アドレスとワイルドカードの値 171.69.0.0 0.0.255.255 への任意の TCP Telnet トラフィックを許可し、その他の TCP トラフィックを拒否します。ICMP トラフィックを許可し、任意の送信元から、宛先ポートが 1024 より小さい 171.69.0.0 ～ 179.69.255.255 の宛先アドレスへ送信される UDP トラフィックを拒否します。それ以外のすべての IP トラフィックを拒否して、結果を示すログが表示されます。

```
Device(config)# ip access-list extended marketing_group
Device(config-ext-nacl)# permit tcp any 171.69.0.0 0.0.255.255 eq telnet
Device(config-ext-nacl)# deny tcp any any
Device(config-ext-nacl)# permit icmp any any
Device(config-ext-nacl)# deny udp any 171.69.0.0 0.0.255.255 lt 1024
Device(config-ext-nacl)# deny ip any any log
Device(config-ext-nacl)# exit
```

名前付き ACL からの個別 ACE の削除

次に、名前付きアクセス リスト *border-list* から ACE を個別に削除する例を示します。

```
Device(config)# ip access-list extended border-list
Device(config-ext-nacl)# no permit ip host 10.1.1.3 any
```

例：アクセス リストのエントリの並べ替え

次に、並べ替える前と後のアクセス リストの例を示します。開始値は 1、増分値は 2 です。後続のエントリはユーザ指定の増分値に基づいて並べられています。範囲は 1 ～ 2147483647 です。

シーケンス番号のないエントリが入力されると、デフォルトで、アクセス リストの最後のエントリのシーケンス番号に 10 を加えたシーケンス番号が割り当てられます。

```
Router# show access-list carls
Extended IP access list carls
    10 permit ip host 10.3.3.3 host 172.16.5.34
    20 permit icmp any any
    30 permit tcp any host 10.3.3.3
    40 permit ip host 10.4.4.4 any
    50 Dynamic test permit ip any any
    60 permit ip host 172.16.2.2 host 10.3.3.12
    70 permit ip host 10.3.3.3 any log
    80 permit tcp host 10.3.3.3 host 10.1.2.2
    90 permit ip host 10.3.3.3 any
    100 permit ip any any
Router(config)# ip access-list extended carls
Router(config)# ip access-list resequence carls 1 2
Router(config)# end
Router# show access-list carls
Extended IP access list carls
    1 permit ip host 10.3.3.3 host 172.16.5.34
    3 permit icmp any any
    5 permit tcp any host 10.3.3.3
    7 permit ip host 10.4.4.4 any
    9 Dynamic test permit ip any any
    11 permit ip host 172.16.2.2 host 10.3.3.12
    13 permit ip host 10.3.3.3 any log
    15 permit tcp host 10.3.3.3 host 10.1.2.2
    17 permit ip host 10.3.3.3 any
    19 permit ip any any
```

例：シーケンス番号を指定したエントリの追加

次の例では、新しいエントリ（シーケンス番号 15）がアクセス リストに追加されます。

```
Router# show ip access-list
Standard IP access list tryon
2 permit 10.4.4.2, wildcard bits 0.0.255.255
5 permit 10.0.0.44, wildcard bits 0.0.0.255
10 permit 10.0.0.1, wildcard bits 0.0.0.255
20 permit 10.0.0.2, wildcard bits 0.0.0.255
Router(config)# ip access-list standard tryon
Router(config-std-nacl)# 15 permit 10.5.5.5 0.0.0.255
Router# show ip access-list
Standard IP access list tryon
2 permit 10.4.0.0, wildcard bits 0.0.255.255
```

```
5 permit 10.0.0.0, wildcard bits 0.0.0.255
10 permit 10.0.0.0, wildcard bits 0.0.0.255
15 permit 10.5.5.0, wildcard bits 0.0.0.255
20 permit 10.0.0.0, wildcard bits 0.0.0.255
```

例：シーケンス番号を指定しないエントリの追加

次に、シーケンス番号が指定されていないエントリをアクセス リストの末尾に追加する方法を示します。シーケンス番号のないエントリを追加すると、自動的にシーケンス番号が割り当てられ、アクセス リストの末尾に配置されます。デフォルトの増分値は 10 であるため、エントリには、既存のアクセス リストの最後のエントリのシーケンス番号に 10 を加えたシーケンス番号が割り当てられます。

```
Router(config)# ip access-list standard resources
Router(config-std-nacl)# permit 10.1.1.1 0.0.0.255
Router(config-std-nacl)# permit 10.2.2.2 0.0.0.255
Router(config-std-nacl)# permit 10.3.3.3 0.0.0.255
Router# show access-list
Standard IP access list resources
10 permit 10.1.1.1, wildcard bits 0.0.0.255
20 permit 10.2.2.2, wildcard bits 0.0.0.255
30 permit 10.3.3.3, wildcard bits 0.0.0.255
Router(config)# ip access-list standard resources
Router(config-std-nacl)# permit 10.4.4.4 0.0.0.255
Router(config-std-nacl)# end
Router# show access-list
Standard IP access list resources
10 permit 10.1.1.1, wildcard bits 0.0.0.255
20 permit 10.2.2.2, wildcard bits 0.0.0.255
30 permit 10.3.3.3, wildcard bits 0.0.0.255
40 permit 10.4.4.4, wildcard bits 0.0.0.255
```

例：コメント付き IP ACL エントリの設定

次に示す番号付き ACL の例では、Jones が所有するワークステーションにはアクセスを許可し、Smith が所有するワークステーションにはアクセスを許可しません。

```
Device(config)# access-list 1 remark Permit only Jones workstation through
Device(config)# access-list 1 permit 171.69.2.88
Device(config)# access-list 1 remark Do not allow Smith workstation through
Device(config)# access-list 1 deny 171.69.3.13
```

次に示す番号付き ACL の例では、Winter および Smith のワークステーションに Web 閲覧を許可しません。

```
Device(config)# access-list 100 remark Do not allow Winter to browse the web
Device(config)# access-list 100 deny host 171.69.3.85 any eq www
Device(config)# access-list 100 remark Do not allow Smith to browse the web
Device(config)# access-list 100 deny host 171.69.3.13 any eq www
```

次に示す名前付き ACL の例では、Jones のサブネットにアクセスを許可しません。

```
Device(config)# ip access-list standard prevention
Device(config-std-nacl)# remark Do not allow Jones subnet through
Device(config-std-nacl)# deny 171.69.0.0 0.0.255.255
```

次に示す名前付き ACL の例では、Jones のサブネットに発信 Telnet の使用を許可しません。

```
Device(config)# ip access-list extended telnetting
Device(config-ext-nacl)# remark Do not allow Jones subnet to telnet out
Device(config-ext-nacl)# deny tcp 171.69.0.0 0.0.255.255 any eq telnet
```

例：ACL での時間範囲を使用

次の例に、*workhours*（営業時間）の時間範囲および会社の休日（2006 年 1 月 1 日）を設定し、設定を確認する例を示します。

```
Device# show time-range
time-range entry: new_year_day_2003 (inactive)
   absolute start 00:00 01 January 2006 end 23:59 01 January 2006
time-range entry: workhours (inactive)
   periodic weekdays 8:00 to 12:00
   periodic weekdays 13:00 to 17:00
```

時間範囲を適用するには、時間範囲を実装できる拡張 ACL 内に時間範囲名を入力します。次に、拡張アクセス リスト 188 を作成して確認する例を示します。このアクセス リストでは、定義された休業時間中はすべての送信元からすべての宛先への TCP トラフィックを拒否し、営業時間中はすべての TCP トラフィックを許可します。

```
Device(config)# access-list 188 deny tcp any any time-range new_year_day_2006
Device(config)# access-list 188 permit tcp any any time-range workhours
Device(config)# end
Device# show access-lists
Extended IP access list 188
   10 deny tcp any any time-range new_year_day_2006 (inactive)
   20 permit tcp any any time-range workhours (inactive)
```

次に、名前付き ACL を使用して同じトラフィックを許可および拒否する例を示します。

```
Device(config)# ip access-list extended deny_access
Device(config-ext-nacl)# deny tcp any any time-range new_year_day_2006
Device(config-ext-nacl)# exit
Device(config)# ip access-list extended may_access
Device(config-ext-nacl)# permit tcp any any time-range workhours
Device(config-ext-nacl)# end
Device# show ip access-lists
Extended IP access list lpip_default
    10 permit ip any any
Extended IP access list deny_access
    10 deny tcp any any time-range new_year_day_2006 (inactive)
Extended IP access list may_access
    10 permit tcp any any time-range workhours (inactive)
```

例：IP ACL に適用される時間範囲

次に、月曜日から金曜日の午前 8 時 ～午後 6 時（18 時）の間、IP の HTTP トラフィックを拒否する例を示します。UDP トラフィックは、土曜日および日曜日の正午～午後 8 時（20 時）の間だけ許可されます。

```
Device(config)# time-range no-http
Device(config)# periodic weekdays 8:00 to 18:00
!
Device(config)# time-range udp-yes
Device(config)# periodic weekend 12:00 to 20:00
!
Device(config)# ip access-list extended strict
Device(config-ext-nacl)# deny tcp any any eq www time-range no-http
Device(config-ext-nacl)# permit udp any any time-range udp-yes
!
Device(config-ext-nacl)# exit
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# ip access-group strict in
```

例：ACL ロギング

ACL では 2 種類のロギングがサポートされています。**log** キーワードを指定すると、エントリと一致するパケットに関するログ通知メッセージがコンソールに送信されます。**log-input** キーワードを指定すると、ログ エントリに入力インターフェイスが追加されます。

次の例では、名前付き標準アクセス リスト *stan1* は 10.1.1.0 0.0.0.255 からのトラフィックを拒否し、その他のすべての送信元からのトラフィックを許可します。**log** キーワードも指定されています。

```
Device(config)# ip access-list standard stan1
Device(config-std-nacl)# deny 10.1.1.0 0.0.0.255 log
Device(config-std-nacl)# permit any log
Device(config-std-nacl)# exit
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# ip access-group stan1 in
Device(config-if)# end
Device# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: level debugging, 37 messages logged
    Monitor logging: level debugging, 0 messages logged
    Buffer logging: level debugging, 37 messages logged
    File logging: disabled
    Trap logging: level debugging, 39 message lines logged

Log Buffer (4096 bytes):

00:00:48: NTP: authentication delay calculation problems

<output truncated>

00:09:34:%SEC-6-IPACCESSLOGS:list stan1 permitted 0.0.0.0 1 packet
00:09:59:%SEC-6-IPACCESSLOGS:list stan1 denied 10.1.1.15 1 packet
00:10:11:%SEC-6-IPACCESSLOGS:list stan1 permitted 0.0.0.0 1 packet
```

次に、名前付き拡張アクセス リスト *ext1* によって、任意の送信元から 10.1.1.0 0.0.0.255 への ICMP パケットを許可し、すべての UDP パケットを拒否する例を示します。

```
Device(config)# ip access-list extended ext1
Device(config-ext-nacl)# permit icmp any 10.1.1.0 0.0.0.255 log
Device(config-ext-nacl)# deny udp any any log
Device(config-std-nacl)# exit
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# ip access-group ext1 in
```

次に、拡張 ACL のログの例を示します。

```
01:24:23:%SEC-6-IPACCESSLOGDP:list ext1 permitted icmp 10.1.1.15 -> 10.1.1.61 (0/0), 1
packet
01:25:14:%SEC-6-IPACCESSLOGDP:list ext1 permitted icmp 10.1.1.15 -> 10.1.1.61 (0/0), 7
packets
01:26:12:%SEC-6-IPACCESSLOGP:list ext1 denied udp 0.0.0.0(0) -> 255.255.255.255(0), 1
packet
01:31:33:%SEC-6-IPACCESSLOGP:list ext1 denied udp 0.0.0.0(0) -> 255.255.255.255(0), 8
packets
```

IP ACL のすべてのロギング エントリは %SEC-6-IPACCESSLOG で開始します。エントリの形式は、一致した ACL やアクセス エントリの種類に応じて若干異なります。

次に、**log-input** キーワードを指定した場合の出力メッセージの例を示します。

```
00:04:21:%SEC-6-IPACCESSLOGDP:list inputlog permitted icmp 10.1.1.10 (Vlan1 0001.42ef.a400)
 ->
10.1.1.61 (0/0), 1 packet
```

log キーワードを指定した場合、同様のパケットに関するログ メッセージには入力インターフェイス情報が含まれません。

```
00:05:47:%SEC-6-IPACCESSLOGDP:list inputlog permitted icmp 10.1.1.10 -> 10.1.1.61 (0/0),
 1 packet
```

例：ACL のトラブルシューティング

次の ACL マネージャ メッセージが表示されて [chars] がアクセス リスト名である場合は次のようにします。

```
ACLMGR-2-NOVMR: Cannot generate hardware representation of access list [chars]
```

スイッチには、ACL のハードウェア表現を作成するのに使用可能なリソースが不足しています。このリソースには、ハードウェア メモリおよびラベル スペースが含まれますが、CPU メモリは含まれません。この問題の原因は、使用可能な論理演算ユニットまたは専用のハードウェア リソースの不足です。論理演算ユニットは、TCP フラグの一致、または TCP、UDP、SCTP ポート番号での **eq**（**ne**、**gt**、**lt**、または **range**）のテストで必要です。

次のいずれかの回避策を使用します。

- ACL の設定を変更して使用するリソースを減らします。
- 他の ACL 名または番号よりも英数字順で先に表示される名前または番号に ACL の名前を変更します。

十分なリソースがない ACL の設定の詳細については、Bug Toolkit の CSCsq63926 を参照してください。

たとえば、次の ACL をインターフェイスに適用します。

```
permit tcp source source-wildcard destination destination-wildcard range 5 60
permit tcp source source-wildcard destination destination-wildcard range 15 160
permit tcp source source-wildcard destination destination-wildcard range 115 1660
permit tcp source source-wildcard destination destination-wildcard
```

なおかつ次のメッセージが表示される場合は次のようにします。

```
ACLMGR-2-NOVMR: Cannot generate hardware representation of access list [chars]
```

フラグ関連の演算子は使用できません。この問題を回避するには、

- **ip access-list resequence** グローバル コンフィギュレーション コマンドを使用することによって、4 つ目の ACE を 1 つ目の ACE の前に移動させます。

  ```
  permit tcp source source-wildcard destination destination-wildcard
  permit tcp source source-wildcard destination destination-wildcard range 5 60
  permit tcp source source-wildcard destination destination-wildcard range 15 160
  permit tcp source source-wildcard destination destination-wildcard range 115 1660
  ```

  または

- 他の ACL 名または番号よりも英数字順で先に表示される名前または番号に ACL の名前を変更します（たとえば、ACL *79* を ACL *1* に変更します）。

これで、ACL 内の 1 つ目の ACE をインターフェイスに適用できます。スイッチによって、ACE が、Opselect インデックス内の利用可能なマッピング ビットに割り当てられ、次に、ハードウェア メモリ内の同じビットを使用するフラグ関連の演算子が割り当てられます。

IPv4 アクセスコントロールリストに関する追加情報

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*

関連項目	マニュアル タイトル
ACL	詳細については、以下を参照してください。 • 『Security Configuration Guide』の「Access Control Lists Overview」 • 『Security Configuration Guide』の「Configuring IPv6 Access Control Lists」

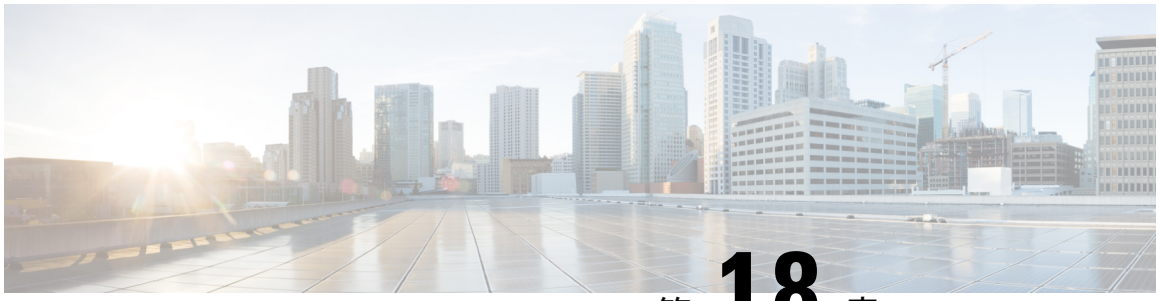
IPv4 アクセスコントロールリストの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	IPv4 アクセスコントロールリスト	この章では、ACL を使用して、スイッチのネットワーク セキュリティを設定する方法について説明します。パケット フィルタリングは、ネットワーク トラフィックを限定し、特定のユーザまたはデバイスによるネットワークの使用を制限するうえで役立ちます。ACL は、トラフィックをデバイスの通過時にフィルタリングし、パケットが指定されたインターフェイスを通過することを許可または拒否します。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 18 章

IPv6 アクセス コントロール リストの設定

IPv6 ACL の制限

IPv4 では、番号制の標準 IP ACL および拡張 IP ACL、名前付き IP ACL、および MAC ACL を設定できます。IPv6 がサポートするのは名前付き ACL だけです。

スイッチは Cisco IOS がサポートする IPv6 ACL の大部分をサポートしますが、一部例外もあります。

- スイッチは、**routing header**、および **undetermined-transport** というキーワードの照合をサポートしません。
- スイッチは、再帰 ACL（**reflect** キーワード）をサポートしません。
- このリリースは、IPv6 のルータ ACL および VLAN ACL（VLAN マップ）をサポートしています。
- スイッチは、IPv6 フレームに MAC ベース ACL を適用しません。
- ACL を設定する場合、ACL に入力されるキーワードには、それがプラットフォームでサポートされるかどうかにかかわらず、制限事項はありません。ハードウェア転送が必要なインターフェイス（物理ポート）に ACL を適用する場合、スイッチはインターフェイスで ACL がサポートされるかどうか判別します。サポートされない場合、ACL の付加は拒否されます。
- インターフェイスに適用される ACL に、サポートされないキーワードを持つアクセス コントロールエントリ（ACE）を追加しようとする場合、スイッチは現在インターフェイスに適用されている ACL に ACE が追加されるのを許可しません。

スイッチの IPv6 ACL には、次の特性があります。

- 分割フレーム（IPv6 では **fragments** キーワード）がサポートされません。
- IPv6 ACL では、IPv4 と同じ統計情報がサポートされます。
- スイッチのハードウェアスペースが不足している場合、ACLに関連付けられたパケットはCPU で処理され、ACL はソフトウェアで適用されます。
- スイッチは、プレフィックス長の最大範囲の IPv6 アドレス一致をサポートしません。

IPv6 ACL の設定に関する情報

アクセス リストによって、デバイス インターフェイスでブロックされるトラフィックおよび転送されるトラフィックが決定され、送信元アドレスと宛先アドレスに基づくトラフィックのフィルタリング、および特定のインターフェイスへの着信および発信トラフィックのフィルタリングを行うことができます。標準のIPv6 ACL機能が拡張されて、IPv6オプションヘッダー、および任意でより細かい制御を行うための上位層プロトコルタイプ情報に基づくトラフィックフィルタリングがサポートされています。標準の IPv6 ACL 機能が拡張されて、IPv6 オプション ヘッダー、および任意でより細かい制御を行うための上位層プロトコル タイプ情報に基づくトラフィック フィルタリングがサポートされています。

このモジュールは、仮想端末回線へのアクセスを制御する IPv6 トラフィック フィルタリングの設定方法について説明します。

ACL の概要

パケット フィルタリングは、ネットワーク トラフィックを限定し、特定のユーザまたはデバイスによるネットワークの使用を制限するうえで役立ちます。ACLはルータまたはスイッチを通過するトラフィックをフィルタリングし、特定のインターフェイスまたは VLAN（仮想 LAN）でパケットを許可、または拒否します。ACL は、パケットに適用される許可条件および拒否条件の順序付けられた集まりです。パケットがインターフェイスに着信すると、スイッチはパケット内のフィールドを適用される ACL と比較し、アクセス リストに指定された基準に基づいて、パケットが転送に必要な権限を持っているかどうかを確認します。アクセスリスト内の条件を1つずつ調べ、パケットをテストします。最初に一致した条件によって、スイッチがパケットを受け入れるか拒否するかが決定されます。スイッチは最初に一致した時点でテストを中止するので、リストに条件を指定する順序が重要です。一致する条件がない場合、スイッチはパケットを拒否します。スイッチは、制限条件がない場合はパケットを転送し、制限条件がある場合はパケットをドロップします。スイッチは、VLAN内でブリッジングされるパケットを含めて、転送されるすべてのパケットに ACL を使用します。

ネットワークに基本的なセキュリティを導入する場合は、ルータにアクセスリストを設定します。ACLを設定しなければ、スイッチを通過するすべてのパケットがネットワークのあらゆる部分で許可される可能性があります。ACLを使用すると、ネットワークの場所ごとにアクセス可能なホストを制御したり、ルータインターフェイスで転送またはブロックされるトラフィックの種類を決定したりできます。たとえば、電子メール トラフィックの転送を許可し、Telnet

トラフィックの転送を拒否することもできます。ACL を着信トラフィック、発信トラフィック、またはその両方をブロックするように設定することもできます。

IPv6 ACL の概要

IP Version 6（IPv6）アクセス コントロール リスト（ACL）を作成し、それをインターフェイスに適用することによって、IPv6 トラフィックをフィルタリングできます。これは、IP Version 4（IPv4）の名前付き ACL を作成し、適用する方法と同じです。

1 つのインターフェイスに、IPv4 ACL および IPv6 ACL の両方を適用できます。

スイッチは、IPv6 トラフィックの Virtual LAN（VLAN）ACL（VLAN マップ）をサポートしません。

他の機能およびスイッチとの相互作用

- ブリッジド フレームがポート ACL によってドロップされる場合、このフレームはブリッジングされません。
- IPv4 ACL および IPv6 ACL の両方を 1 つのスイッチに作成したり、同一のインターフェイスに適用したりできます。各 ACL には一意の名前が必要です。設定済みの名前を使用しようとすると、エラー メッセージが表示されます。

  IPv4 ACL と IPv6 ACL の作成、および同一のレイヤ 2 インターフェイスへの IPv4 ACL または IPv6 ACL の適用には、異なるコマンドを使用します。ACL を付加するのに誤ったコマンドを使用すると（例えば、IPv6 ACL の付加に IPv4 コマンドを使用するなど）、エラー メッセージが表示されます。
- MAC ACL を使用して、IPv6 フレームをフィルタリングできません。MAC ACL は非 IP フレームだけをフィルタリングできます。
- ハードウェアメモリが満杯の場合、ACL に関連付けられたパケットは CPU に向けて処理され、ACL はソフトウェアで適用されます。

IPv6 ACL のデフォルト設定

デフォルトの IPv6 ACL 設定は次のとおりです。

```
Device# show access-lists preauth_ipv6_acl
IPv6 access list preauth_ipv6_acl (per-user)
permit udp any any eq domain sequence 10
permit tcp any any eq domain sequence 20
permit icmp any any nd-ns sequence 30
permit icmp any any nd-na sequence 40
permit icmp any any router-solicitation sequence 50
permit icmp any any router-advertisement sequence 60
permit icmp any any redirect sequence 70
permit udp any eq 547 any eq 546 sequence 80
permit udp any eq 546 any eq 547 sequence 90
deny ipv6 any any sequence 100
```

サポートされる ACL 機能

スイッチの IPv6 ACL には、次の特性があります。

- 分割フレーム（IPv4 では fragments キーワード）がサポートされます。
- IPv6 ACL では、IPv4 と同じ統計情報がサポートされます。
- スイッチの Ternary CAM（TCAM）スペースが不足している場合、ACL ラベルに対応付けられたパケットは CPU に転送され、ACL はソフトウェアで適用されます。

IPv6 ポートベースのアクセス コントロール リスト サポート

IPv6 PACL 機能は、IPv6 トラフィック用のレイヤ 2 スイッチ ポートでアクセス コントロール（許可または拒否）を提供する機能を備えています。IPv6 PACL は、IPv4 トラフィック用のレイヤ 2 スイッチ ポートでアクセス コントロールを提供する IPv4 PACL と似ています。これらは、入力方向とハードウェアだけでサポートされます。

ACL およびトラフィック転送

ホップ バイ ホップ フィルタリングに対応するための IPv6 ACL の拡張機能により、ホップ バイ ホップ拡張ヘッダーを含む可能性がある IPv6 トラフィックを制御することができます。アクセス コントロール リスト（ACL）を設定して、すべてのホップ バイ ホップ トラフィックを拒否するか、またはプロトコルに基づいて選択的にトラフィックを許可することができます。

IPv6 アクセス コントロール リスト（ACL）は、デバイス インターフェイスでブロックされるトラフィックと転送されるトラフィックを決定します。ACL を使用すると、特定のインターフェイスへの着信および発信を、送信元アドレスと宛先アドレスに基づいてフィルタリングできます。**ipv6 access-list** コマンドを使用して IPv6 ACL を定義し、**deny** および **permit** コマンドを使用してその条件を構成します。

ホップ バイ ホップ フィルタリングに対応するための IPv6 ACL の拡張機能は、上位層プロトコル タイプでのトラフィック フィルタリングをサポートするために RFC 2460 を実装します。

IPv6 ACL の設定方法

この項では、IPv6 ACL の設定方法について説明します。

IPv6 ACL の設定

IPv6 トラフィックをフィルタリングする場合は、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	{ **ipv6 access-list** *list-name* 例： `Device(config)# ipv6 access-list example_acl_list`	IPv6 ACL 名を定義し、IPv6 アクセス リスト コンフィギュレーション モードを開始します。
ステップ 4	{**deny** | **permit**} protocol {*source-ipv6-prefix/|prefix-length*|**any**| **host** *source-ipv6-address*} [operator [*port-number*]] { *destination-ipv6-prefix/ prefix-length* | **any** | **host** *destination-ipv6-address*} [operator [*port-number*]][**dscp** *value*] [**fragments**] [**log**] [**log-input**] [**sequence** *value*] [**time-range** *name*]	条件が一致した場合にパケットを拒否する場合は deny、許可する場合は permit を指定します。次に、条件について説明します。 • protocol には、インターネット プロトコルの名前または番号を入力します。**ahp**、**esp**、**icmp**、**ipv6**、**pcp**、**stcp**、**tcp**、**udp**、または IPv6 プロトコル番号を表す 0 ～ 255 の整数を使用できます。 • *source-ipv6-prefix/prefix-length* または *destination-ipv6-prefix/ prefix-length* は、拒否条件または許可条件を設定する送信元または宛先 IPv6 ネットワークあるいはネットワーク クラスで、コロン区切りの 16 ビット値を使用した 16 進形式で指定します（RFC 2373 を参照）。 • IPv6 プレフィックス ::/0 の短縮形として、any を入力します。 • **host** *source-ipv6-address* または *destination-ipv6-address* には、拒否条件または許可条件を設定する送

	コマンドまたはアクション	目的
		信元または宛先 IPv6 ホスト アドレスを入力します。アドレスはコロン区切りの 16 ビット値を使用した 16 進形式で指定します。 • （任意）operator には、指定のプロトコルの送信元ポートまたは宛先ポートを比較するオペランドを指定します。オペランドには、**lt**（より小さい）、**gt**（より大きい）、**eq**（等しい）、**neq**（等しくない）、**range**（包含範囲）があります。 *source-ipv6-prefix/prefix-length* 引数のあとの operator は、送信元ポートに一致する必要があります。*destination-ipv6- prefix/prefix-length* 引数のあとの operator は、宛先ポートに一致する必要があります。 • （任意）**port-number** は、0 ～ 65535 の 10 進数または TCP あるいは UDP ポートの名前です。TCP ポート名を使用できるのは、TCP のフィルタリング時だけです。UDP ポート名を使用できるのは、UDP のフィルタリング時だけです。 • （任意）**dscp** value を入力して、各 IPv6 パケット ヘッダーの Traffic Class フィールド内のトラフィック クラス値と DiffServ コード ポイント値を照合します。指定できる範囲は 0 ～ 63 です。 • （任意）**fragments** を入力して、先頭ではないフラグメントを確認します。このキーワードが表示されるのは、プロトコルが ipv6 の場合だけです。 • （任意）**log** を指定すると、エントリと一致するパケットに関するログ メッセージがコンソールに送信

	コマンドまたはアクション	目的
		されます。**log-input** を指定すると、ログエントリに入力インターフェイスが追加されます。 • （任意）**sequence** *value* を入力して、アクセスリスト ステートメントのシーケンス番号を指定します。指定できる範囲は 1 ～ 4,294,967,295 です。 • （任意）**time-range** name を入力して、deny または permit ステートメントに適用される時間の範囲を指定します。
ステップ 5	{**deny** | **permit**} **tcp** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [**operator** [**port-number**]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [operator [*port-number*]] [**ack**] [**dscp** *value*] [**fin**] [**log**] [**log-input**] [**neq** {**port** | protocol}] [**psh**] [**range** {**port** | protocol}] [**rst**] [**sequence** *value*] [**syn**] [**time-range** *name*] [**urg**]	（任意）TCP アクセス リストおよびアクセス条件を定義します。 TCP の場合は **tcp** を入力します。パラメータはステップ 3a で説明されているパラメータと同じですが、次に示すオプションのパラメータが追加されています。 • **ack**：確認応答ビット セット • **fin**：終了ビット セット。送信元からのデータはそれ以上ありません。 • **neq** {*port* | protocol}：所定のポート番号上にないパケットだけを照合します。 • **psh**：プッシュ機能ビット セット • **range** {*port* | protocol}：ポート番号の範囲内のパケットだけを照合します。 • **rst**：リセット ビット セット • **syn**：同期ビット セット • **urg**：緊急ポインタ ビット セット
ステップ 6	{**deny** | **permit**} **udp** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [operator	（任意）UDP アクセス リストおよびアクセス条件を定義します。

	コマンドまたはアクション	目的
	[*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [operator [*port-number*]] [**dscp** *value*] [**log**] [**log-input**] [**neq** {*port* | *protocol*}] [**range** {*port* | *protocol*}] [**sequence** *value*] [**time-range** *name*]]	ユーザデータグラムプロトコルの場合は、**udp** を入力します。UDP パラメータは TCP の説明にあるパラメータと同じです。ただし、[operator [*port*]] ポートの番号または名前は、UDP ポートの番号または名前でなければなりません。
ステップ 7	{**deny** | **permit**} **icmp** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [operator [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [operator [*port-number*]] [*icmp-type* [*icmp-code*] | icmp-message] [**dscp** *value*] [**log**] [**log-input**] [**sequence** *value*] [**time-range** *name*]	（任意）ICMP アクセス リストおよびアクセス条件を定義します。 インターネット制御メッセージプロトコルの場合は、**icmp** を入力します。ICMP パラメータはステップ 1 の IP プロトコルの説明にあるパラメータとほとんど同じですが、ICMP メッセージ タイプおよびコードパラメータが追加されています。オプションのキーワードの意味は次のとおりです。 • *icmp-type*：ICMP メッセージ タイプでフィルタリングする場合に入力します。指定できる値の範囲は、0 ～ 255 です。 • *icmp-code*：ICMP パケットを ICMP メッセージコードタイプでフィルタリングする場合に入力します。指定できる値の範囲は、0 ～ 255 です。 • *icmp-message*：ICMP パケットを ICMP メッセージ タイプ名または ICMP メッセージ タイプとコード名でフィルタリングする場合に入力します。ICMP メッセージのタイプ名およびコード名のリストについては、? キーを使用するか、またはこのリリースのコマンドリファレンスを参照してください。
ステップ 8	**end**	特権 EXEC モードに戻ります。
ステップ 9	**show ipv6 access-list**	アクセスリストの設定を確認します。

	コマンドまたはアクション	目的
ステップ 10	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

インターフェイスに IPv6 ACL をアタッチします。

IPv6 ACL のモニタリング

次の表に示された 1 つまたは複数の特権 EXEC コマンドを使用して、設定済みのすべてのアクセス リスト、すべての IPv6 アクセス リスト、または特定のアクセス リストに関する情報を表示できます。

コマンド	目的
show access-lists	スイッチに設定されたすべてのアクセス リストを表示します。
show ipv6 access-list [*access-list-name*]	設定済みのすべての IPv6 アクセス リストまたは名前で指定されたアクセス リストを表示します。

次に、show access-lists 特権 EXEC コマンドの出力例を示します。スイッチに設定されているすべてのアクセス リストが表示されます。

```
Device# show access-lists
Extended IP access list hello
    10 permit ip any any
IPv6 access list ipv6
    permit ipv6 any any sequence 10
```

次に、**show ipv6 access-lists** 特権 EXEC コマンドの出力例を示します。スイッチに設定されている IPv6 アクセス リストだけが表示されます。

```
Device# show ipv6 access-list
IPv6 access list inbound
    permit tcp any any eq bgp (8 matches) sequence 10
    permit tcp any any eq telnet (15 matches) sequence 20
    permit udp any any sequence 30
```

```
IPv6 access list outbound
    deny udp any any sequence 10
    deny tcp any any eq telnet sequence 20
```

インターフェイスでの PACL モードの設定および IPv6 PACL の適用

始める前に

IPv6 PACL 機能を設定する前に、IPv6 アクセス リストを設定する必要があります。IPv6 アクセス リストを設定した後、指定された IPv6 レイヤ 2 インターフェイスでポートベース アクセス コントロール リスト（PACL）モードを設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 access-list** *access-list-name* 例： `Device(config)# ipv6 access-list list1`	IPv6 ACL を定義し、IPv6 アクセス リスト コンフィギュレーション モードを開始します。
ステップ 4	**exit** 例： `Device(config-ipv6-acl)# exit`	IPv6 アクセス リスト コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードを開始します。
ステップ 5	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイスのタイプと番号を指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 6	**ipv6 traffic-filter** *access-list-name* **in** 例： `Device(config-if)# ipv6 traffic-filter list1 in`	インターフェイス上の着信 IPv6 トラフィックをフィルタリングします。

	コマンドまたはアクション	目的
ステップ 7	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。

ホップ バイ ホップ フィルタリングに対応するための IPv6 ACL の拡張の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 access-list** *access-list-name* 例： `Device(config)# ipv6 access-list hbh-acl`	IPv6 ACL を定義し、IPv6 アクセス リスト コンフィギュレーション モードを開始します。
ステップ 4	**permit** *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address* } [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* } [*operator* [*port-number*]] [**dscp** *value*] [**hbh**] [**log**] [**log-input**] [**reflect** *name* [**timeout** *value*]] [**sequence** *value*] [**time-range** *name*] 例： `Device(config-ipv6-acl)# permit icmp any any`	IPv6 ACL の許可条件を設定します。
ステップ 5	**deny** *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address* } [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* } [*operator* [*port-number*]] [**dscp** *value*] [**hbh**] [**log**]	IPv6 ACL の拒否条件を設定します。

	コマンドまたはアクション	目的
	[**log-input**] [**sequence** *value*] [**time-range** *name*] 例： `Device(config-ipv6-acl)# deny icmp any any`	
ステップ 6	**end** 例： `Device (config-ipv6-acl)# end`	特権EXECコンフィギュレーションモードに戻ります。

IPv6 ACL の設定例

この項では、IPv6 ACL の設定例を示します。

例：IPv6 ACL の設定

次に、CISCO と名前が付けられた IPv6 アクセス リストを設定する例を示します。リスト内の最初の拒否エントリは、宛先 TCP ポート番号が 5000 より大きいパケットをすべて拒否します。2 番めの拒否エントリは、送信元 UDP ポート番号が 5000 未満のパケットを拒否します。また、この2番めの拒否エントリは、すべての一致をコンソールに表示します。リスト内の最初の許可エントリは、すべての ICMP パケットを許可します。リスト内の 2 番めの許可エントリは、その他のすべてのトラフィックを許可します。暗黙の全否定の条件が各 IPv6 アクセス リストの末尾にあるため、2 番めの許可エントリは必要です。

```
Device(config)# ipv6 access-list CISCO
Device(config-ipv6-acl)# deny tcp any any gt 5000
Device(config-ipv6-acl)# deny ::/0 lt 5000 ::/0 log
Device(config-ipv6-acl)# permit icmp any any
Device(config-ipv6-acl)# permit any any
```

例：インターフェイスでの PACL モードの設定および IPv6 PACL の適用

```
Device# configure terminal
Device(config)# ipv6 access-list list1
Device(config-ipv6-acl)# exit
Device(config-if)# ipv6 traffic-filter list1 in
```

例：ホップ バイ ホップ フィルタリングに対応するための IPv6 ACL の拡張

```
Device(config)# ipv6 access-list hbh_acl
```

```
Device(config-ipv6-acl)# permit tcp any any hbh
Device(config-ipv6-acl)# permit tcp any any
Device(config-ipv6-acl)# permit udp any any
Device(config-ipv6-acl)# permit udp any any hbh
Device(config-ipv6-acl)# permit hbh any any
Device(config-ipv6-acl)# permit any any
Device(config-ipv6-acl)# exit

! Assign an IP address and add the ACL on the interface.

Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# ipv6 address 1001::1/64
Device(config-if)# ipv6 traffic-filter hbh_acl in
Device(config-if)# exit
Device(config)# exit
Device# clear counters
Clear "show interface" counters on all interfaces [confirm]
Device#

! Verify the configurations.

Device# show running-config interface gigabitethernet 1/0/1

Building configuration...

Current configuration : 114 bytes
!
interface gigabitethernet 1/0/1
no switchport
ipv6 address 1001::1/64
ipv6 traffic-filter hbh_acl
end
```

IPv6 アクセスコントロールリストに関する追加情報

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*
ACL	詳細については、以下を参照してください。 • 『Security Configuration Guide』の「Access Control Lists Overview」 • 『Security Configuration Guide』の「Configuring IPv4 Access Control Lists」

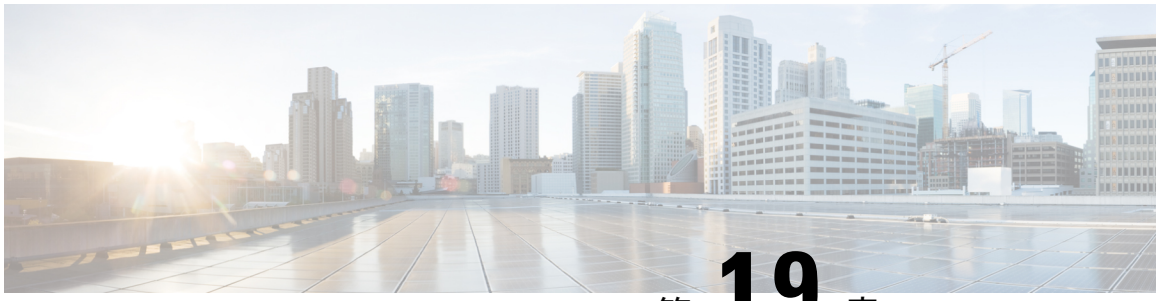
IPv6 アクセスコントロールリストの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	IPv6 アクセス コントロール リスト	IPv6 ACL を作成して、インターフェイスに適用することによって、IPv6 トラフィックをフィルタリングできます。これは、IPv4 の名前付き ACL を作成し、適用する方法と類似しています。レイヤ 3 管理トラフィックをフィルタリングするために、入力ルータ ACL を作成し、適用することもできます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 19 章

IEEE 802.1x ポートベースの認証の設定

802.1x ポートベース認証の前提条件

次のタスクは、IEEE 802.1X ポートベース認証機能を実装する前に完了する必要があります。

- IEEE 802.1X をデバイス ポートで有効にする必要があります。
- デバイスが RADIUS 設定されていること、および Cisco Secure アクセス コントロール サーバ（ACS）に接続されていることが必要です。RADIUS プロトコルの概念とアクセス コントロール リスト（ACL）の作成および適用方法を理解しておく必要があります。
- EAP サポートを RADIUS サーバで有効にする必要があります。
- ユーザがログオフしたときに EAP-Logoff（Stop）メッセージがデバイスに送信されるよう、IEEE 802.1X サプリカントを設定する必要があります。IEEE 802.1X サプリカントをこのように設定しないと、EAP-Logoff メッセージはデバイスに送信されず、付随するアカウンティング Stop メッセージが認証サーバに送信されません。
- すべてのネットワーク関連のサービス要求について、ポートで認証、許可、およびアカウンティング（AAA）を設定する必要があります。認証方式リストを有効化および指定する必要があります。方式リストは、ユーザ認証のためにクエリ送信を行う手順と認証方式を記述したものです。
- ポートの認証に成功する必要があります。

IEEE 802.1x ポートベースの認証に関する情報

802.1x ポートベース認証の概要

802.1x 規格では、一般の人がアクセス可能なポートから不正なクライアントが LAN に接続しないように規制する（適切に認証されている場合を除く）、クライアント/サーバ型のアクセスコントロールおよび認証プロトコルを定めています。認証サーバがスイッチポートに接続する各クライアントを認証したうえで、デバイスまたは LAN が提供するサービスを利用できるようにします。

（注） TACACS は、802.1x 認証ではサポートされていません。

802.1x アクセスコントロールでは、クライアントを認証するまでの間、そのクライアントが接続しているポート経由では Extensible Authentication Protocol over LAN（EAPOL）、Cisco Discovery Protocol、およびスパニングツリープロトコル（STP）トラフィックしか許可されません。認証に成功すると、通常のトラフィックはポートを通過できるようになります。

次の表は、サポートされる各クライアントセッションの最大数を示しています。

クライアント セッション	サポートされる最大セッション数
dot1x または MAB クライアント セッションの最大数	2000
Web ベース認証セッションの最大数	2000
クリティカル認証 VLAN を有効にしてサーバを再初期化した dot1x セッションの最大数	2000
さまざまなセッション機能が適用される MAB セッションの最大数	2000
サービス テンプレートまたはセッション機能が適用される dot1x セッションの最大数	2000

ポートベース認証プロセス

IEEE 802.1X ポートベース認証を設定するには、認証、認可、およびアカウンティング（AAA）を有効にし、認証方式リストを指定する必要があります。方式リストは、ユーザ認証のためにクエリー送信を行う手順と認証方式を記述したものです。

AAA プロセスは認証から始まります。802.1x ポートベース認証がイネーブルであり、クライアントが 802.1x 準拠のクライアント ソフトウェアをサポートしている場合、次のイベントが発生します。

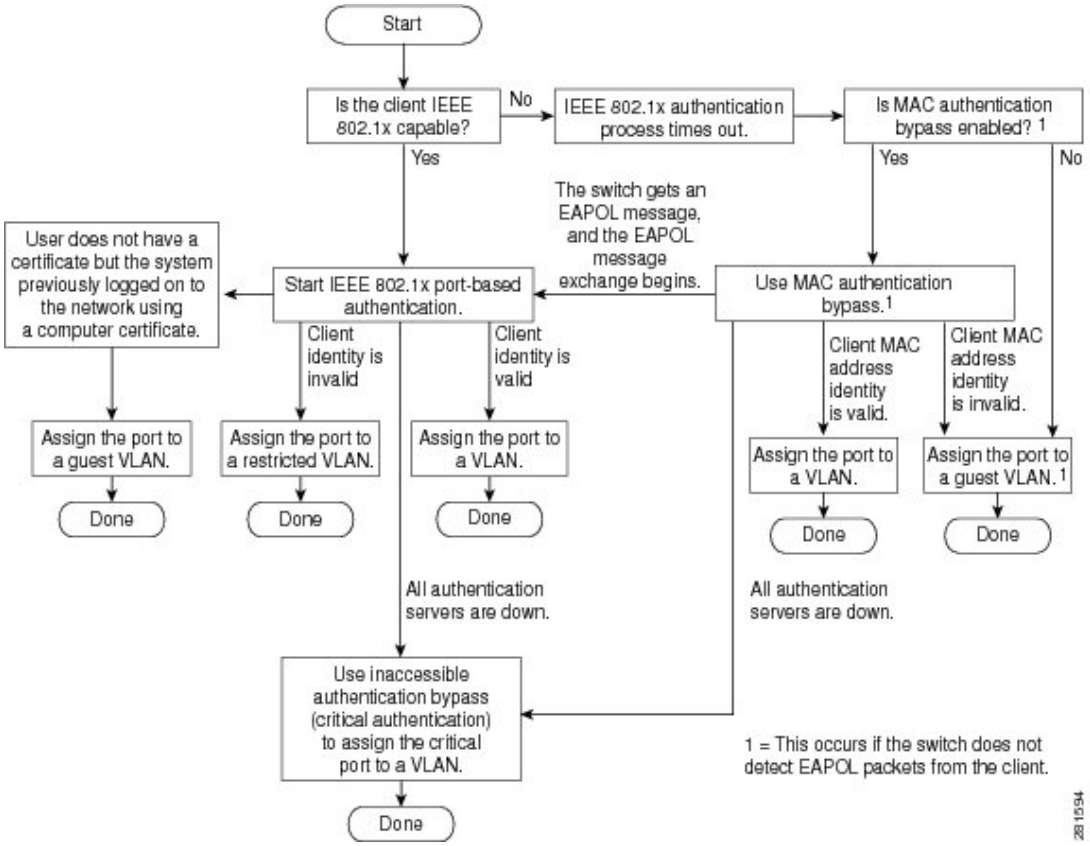
- クライアント ID が有効で 802.1x 認証に成功した場合、デバイスはクライアントにネットワークへのアクセスを許可します。
- EAPOL メッセージ交換の待機中に 802.1x 認証がタイムアウトし、MAC 認証バイパスがイネーブルの場合、デバイスはクライアント MAC アドレスを認証用に使用します。このクライアント MAC アドレスが有効で認証に成功した場合、デバイスはクライアントにネットワークへのアクセスを許可します。クライアント MAC アドレスが無効で認証に失敗した場合、ゲスト VLAN が設定されていれば、デバイスはクライアントに限定的なサービスを提供するゲスト VLAN を割り当てます。
- デバイスが 802.1x 対応クライアントから無効な ID を取得し、制限付き VLAN が指定されている場合、デバイスはクライアントに限定的なサービスを提供する制限付き VLAN を割り当てることができます。
- RADIUS 認証サーバが使用できず（ダウンしていて）アクセスできない認証バイパスがイネーブルの場合、デバイスは、RADIUS 設定済み VLAN またはユーザ指定のアクセス VLAN で、ポートをクリティカル認証ステートにして、クライアントにネットワークへのアクセスを許可します。

注　アクセスできない認証バイパスは、クリティカル認証、または AAA 失敗ポリシーとも呼ばれます。

ポートで Multi Domain Authentication（MDA）が有効になっている場合、音声許可に該当する例外をいくつか伴ったフローを使用できます。

図 *5:* 認証フローチャート

次の図は認証プロセスを示します。



次の状況のいずれかが発生すると、デバイスはクライアントを再認証します。

- 定期的な再認証がイネーブルで、再認証タイマーの期限が切れている場合。

  デバイス固有の値を使用するか、RADIUS サーバからの値に基づいて再認証タイマーを設定できます。

  RADIUS サーバを使用した 802.1x 認証の後で、デバイスは Session-Timeout RADIUS 属性（Attribute[27]）、および Termination-Action RADIUS 属性（Attribute[29]）に基づいてタイマーを使用します。

  Session-Timeout RADIUS 属性（Attribute[27]）には再認証が行われるまでの時間を指定します。

  Termination-Action RADIUS 属性（Attribute[29]）には、再認証中に行われるアクションを指定します。アクションは *Initialize* および *ReAuthenticate* に設定できます。アクションに *Initialize*（属性値は *DEFAULT*）を設定した場合、802.1x セッションは終了し、認証中、接続は失われます。アクションに *ReAuthenticate*（属性値は RADIUS-Request）を設定した場合、セッションは再認証による影響を受けません。

- クライアントを手動で再認証するには、**dot1x re-authenticate interface** *interface-id* 特権 EXEC コマンドを入力します。

ポートベース認証の開始およびメッセージ交換

802.1x認証中に、デバイスまたはクライアントは認証を開始できます。**authentication port-control auto** インターフェイス コンフィギュレーション コマンドを使用してポート上で認証をイネーブルにすると、デバイスは、リンクステートがダウンからアップに移行したときに認証を開始し、ポートがアップしていて認証されていない場合は定期的に認証を開始します。デバイスはクライアントに EAP-Request/Identity フレームを送信し、識別情報を要求します。クライアントはフレームを受信すると、EAP-Response/Identity フレームで応答します。

ただし、クライアントが起動時にデバイスからの EAP-Request/Identity フレームを受信しなかった場合、クライアントは EAPOL-Start フレームを送信して認証を開始できます。このフレームはデバイスに対し、クライアントの識別情報を要求するように指示します。

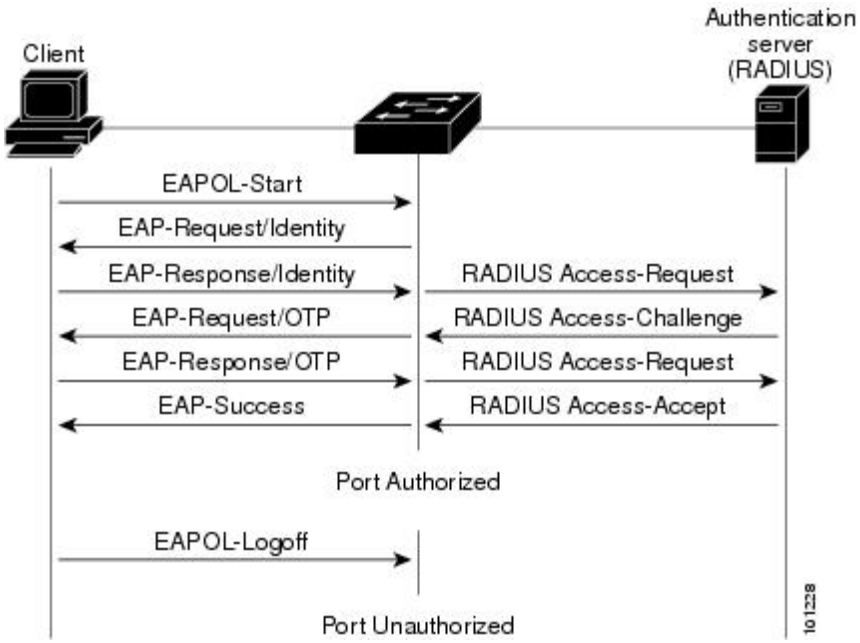
（注） ネットワーク アクセス デバイスで 802.1x 認証がイネーブルに設定されていない、またはサポートされていない場合には、クライアントからの EAPOL フレームはすべて廃棄されます。クライアントが認証の開始を 3 回試みても EAP-Request/Identity フレームを受信しなかった場合、クライアントはポートが許可ステートであるものとしてフレームを送信します。ポートが許可ステートであるということは、クライアントの認証が成功したことを実質的に意味します。

クライアントが自らの識別情報を提示すると、デバイスは仲介デバイスとしての役割を開始し、認証が成功または失敗するまで、クライアントと認証サーバの間で EAP フレームを送受信します。認証が成功すると、ポートは許可ステートになります。認証に失敗した場合、認証が再試行されるか、ポートが限定的なサービスを提供する VLAN に割り当てられるか、あるいはネットワーク アクセスが許可されないかのいずれかになります。

実際に行われる EAP フレーム交換は、使用する認証方式によって異なります。

図 *6:* メッセージ交換

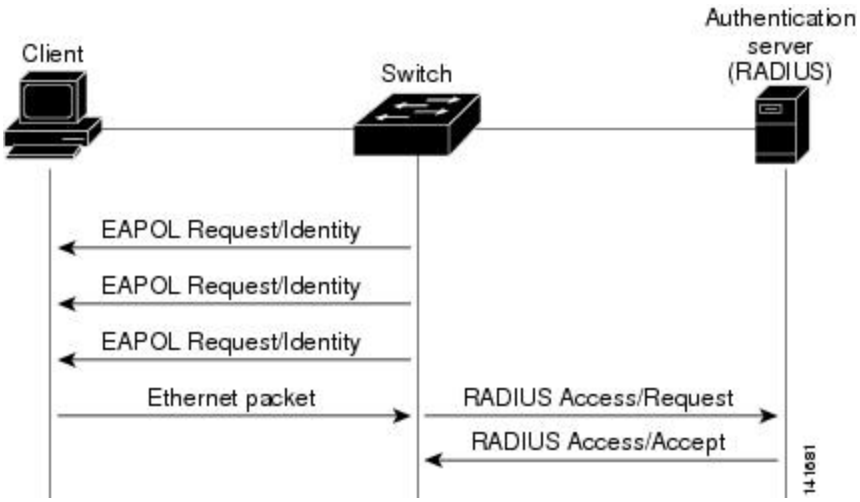
次の図に、クライアントが RADIUS サーバとの間で OTP（ワンタイム パスワード）認証方式を使用する際に行われるメッセージ交換を示します。



EAPOL メッセージ交換の待機中に 802.1x 認証がタイムアウトし、MAC 認証バイパスがイネーブルの場合、デバイスはクライアントからイーサネットパケットを検出するとそのクライアントを許可できます。デバイスは、クライアントの MAC アドレスを識別情報として使用し、RADIUS サーバに送信される RADIUS-Access/Request フレームにこの情報を保存します。サーバがデバイスに RADIUS-Access/Accept フレームを送信（許可が成功）すると、ポートが許可されます。許可に失敗してゲスト VLAN が指定されている場合、デバイスはポートをゲスト VLAN に割り当てます。イーサネットパケットの待機中にデバイスが EAPOL パケットを検出すると、デバイスは MAC 認証バイパスプロセスを停止して、802.1x 認証を開始します。

図 *7 : MAC* 認証バイパス中のメッセージ交換

次の図に、MAC 認証バイパス中のメッセージ交換を示します。



ポートベース認証方法

表 *20 : 802.1x* 機能

認証方法	モード			
	シングル ホスト	マルチ ホスト	MDA	複数認証
802.1x	VLAN 割り当て ユーザ単位 ACL Filter-ID 属性 ダウンロード可能 ACL リダイレクト URL	VLAN 割り当て	VLAN 割り当て ユーザ単位 ACL Filter-ID 属性 ダウンロード可能 ACL リダイレクト URL	VLAN 割り当て ユーザ単位 ACL Filter-ID 属性 ダウンロード可能 ACL リダイレクト URL
MAC 認証バイパス	VLAN 割り当て ユーザ単位 ACL Filter-ID 属性 ダウンロード可能 ACL リダイレクト URL	VLAN 割り当て	VLAN 割り当て ユーザ単位 ACL Filter-ID 属性 ダウンロード可能 ACL リダイレクト URL	VLAN 割り当て ユーザ単位 ACL Filter-ID 属性 ダウンロード可能 ACL リダイレクト URL

認証方法	モード			
	シングル ホスト	マルチ ホスト	MDA	複数認証
スタンドアロン Web 認証	プロキシ ACL、Filter-ID 属性、ダウンロード可能 ACL			
NAC レイヤ 2 IP 検証	Filter-ID 属性 ダウンロード可能 ACL リダイレクト URL	Filter-ID 属性 ダウンロード可能 ACL リダイレクト URL	Filter-ID 属性 ダウンロード可能 ACL リダイレクト URL	Filter-ID 属性 ダウンロード可能 ACL リダイレクト URL
フォールバック メソッドとしての Web 認証 （注） 802.1x 認証をサポートしていないクライアントの場合。	プロキシ ACL Filter-ID 属性 ダウンロード可能 ACL	プロキシ ACL Filter-ID 属性 ダウンロード可能 ACL	プロキシ ACL Filter-ID 属性 ダウンロード可能 ACL	プロキシ ACL Filter-ID 属性 ダウンロード可能 ACL

ポートベース認証マネージャ CLI コマンド

認証マネージャ インターフェイス コンフィギュレーション コマンドは、802.1x、MAC 認証バイパスおよび Web 認証など、すべての認証方法を制御します。認証マネージャ コマンドは、接続ホストに適用される認証方法のプライオリティと順序を決定します。

これらのコマンドは、ホストモード、違反モード、および認証タイマーなど、一般的な認証機能を制御します。一般的な認証コマンドには、**authentication host-mode**、**authentication violation**、および **authentication timer** インターフェイス コンフィギュレーション コマンドが含まれます。

デバイスで dot1x をディセーブルにするには、**no dot1x system-auth-control** コマンドを使用して設定をグローバルに削除し、設定されているすべてのインターフェイスからも削除します。

（注） 802.1x 認証がグローバルにディセーブル化されても、Web 認証など他の認証方法はそのポートでイネーブルのままです。

authentication manager コマンドは、以前の 802.1x コマンドと同じ機能を提供します。

認証マネージャが生成する冗長なシステムメッセージをフィルタリングすると、通常は、フィルタリングされた内容が認証の成功に結びつきます。802.1x 認証および MAB 認証の冗長なメッセージをフィルタリングすることもできます。認証方式ごとに異なるコマンドが用意されています。

- **no authentication logging verbose** グローバル コンフィギュレーション コマンドは、認証マネージャからの詳細メッセージをフィルタリングします。
- **no dot1x logging verbose** グローバル コンフィギュレーション コマンドは、802.1x 認証の詳細メッセージをフィルタリングします。
- **no mab logging verbose** グローバル コンフィギュレーション コマンドは、MAC 認証バイパス（MAB）の詳細メッセージをフィルタリングします。

ユーザ単位 ACL および Filter-ID

（注） **any** は、ACL の発信元としてだけ設定できます。

（注） マルチホストモードで設定された ACL では、ステートメントの発信元部分は *any* でなければなりません。（たとえば、 **permit icmp** *any* **host 10.10.1.1**）。

（注） filter-ID としてロールベース ACL を使用することは推奨されません。

定義された ACL の発信元ポートには **any** を指定する必要があります。指定しない場合、ACL は適用できず、認証は失敗します。シングル ホストは唯一例外的に後方互換性をサポートします。

MDA 対応ポートおよびマルチ認証ポートでは、複数のホストを認証できます。ホストに適用される ACL ポリシーは、別のホストのトラフィックには影響を与えません。マルチ ホスト ポートで認証されるホストが 1 つだけで、他のホストが認証なしでネットワーク アクセスを取得する場合、発信元アドレスに any を指定することで、最初のホストの ACL ポリシーを他の接続ホストに適用できます。

許可ステートおよび無許可ステートのポート

802.1x 認証中に、スイッチのポート ステートによって、スイッチはネットワークへのクライアント アクセスを許可します。ポートは最初、無許可ステートです。このステートでは、音声 VLAN ポートとして設定されていないポートは 802.1x 認証、Cisco Discovery Protocol、および STP パケットを除くすべての入力および出力トラフィックを禁止します。クライアントの認証が成功すると、ポートは許可ステートに変更し、クライアントのトラフィック送受信を通常ど

おりに許可します。ポートが音声 VLAN ポートとして設定されている場合、VoIP トラフィックおよび 802.1x プロトコル パケットが許可された後クライアントが正常に認証されます。

（注） Cisco Discovery Protocol バイパスはサポートされていないため、ポートが err-disabled ステートになる場合があります。

802.1x をサポートしていないクライアントが、無許可ステートの 802.1x ポートに接続すると、スイッチはそのクライアントの識別情報を要求します。この状況では、クライアントは要求に応答せず、ポートは引き続き無許可ステートとなり、クライアントはネットワーク アクセスを許可されません。

反対に、802.1x 対応のクライアントが、802.1x 標準が稼働していないポートに接続すると、クライアントは EAPOL-Start フレームを送信して認証プロセスを開始します。応答がなければ、クライアントは同じ要求を所定の回数だけ送信します。応答がないので、クライアントはポートが許可ステートであるものとしてフレーム送信を開始します。

authentication port-control インターフェイス コンフィギュレーション コマンドおよび次のキーワードを使用して、ポートの許可ステートを制御できます。

- **force-authorized**：802.1x 認証を無効にし、認証情報の交換を必要とせずに、ポートを許可ステートに変更します。ポートはクライアントとの 802.1x ベース認証を行わずに、通常のトラフィックを送受信します。これがデフォルト設定です。
- **force-unauthorized**：ポートが無許可ステートのままになり、クライアントからの認証の試みをすべて無視します。スイッチはポートを介してクライアントに認証サービスを提供できません。
- **auto**：802.1x 認証をイネーブルにします。ポートは最初、無許可ステートであり、ポート経由で送受信できるのは EAPOL フレームだけです。ポートのリンク ステートがダウンからアップに変更した際、または EAPOL-Start フレームを受信した際に、認証プロセスが開始されます。スイッチはクライアントの識別情報を要求し、クライアントと認証サーバとの間で認証メッセージのリレーを開始します。スイッチはクライアントの MAC アドレスを使用して、ネットワーク アクセスを試みる各クライアントを一意に識別します。

クライアントが認証に成功すると（認証サーバから Accept フレームを受信すると）、ポートが許可ステートに変わり、認証されたクライアントからの全フレームがポート経由での送受信を許可されます。認証が失敗すると、ポートは無許可ステートのままですが、認証を再試行することはできます。認証サーバに到達できない場合、スイッチは要求を再送信します。所定の回数だけ試行してもサーバから応答が得られない場合には、認証が失敗し、ネットワーク アクセスは許可されません。

クライアントはログオフするとき、EAPOL-Logoff メッセージを送信します。このメッセージによって、スイッチ ポートが無許可ステートになります。

ポートのリンク ステートがアップからダウンに変更した場合、または EAPOL-Logoff フレームを受信した場合に、ポートは無許可ステートに戻ります。

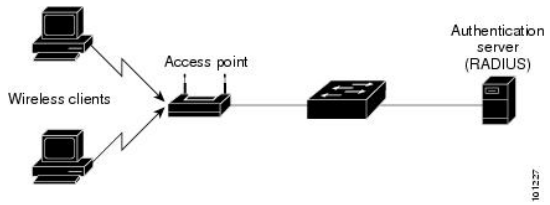
802.1X のホスト モード

802.1x ポートは、シングル ホスト モードまたはマルチ ホスト モードで設定できます。シングルホストモードでは、802.1x 対応ポートに接続できるのはクライアント 1 つだけです。デバイスは、ポートのリンクステートがアップに変化したときに、EAPOL フレームを送信することでクライアントを検出します。クライアントがログオフしたとき、または別のクライアントに代わったときには、デバイスはポートのリンクステートをダウンに変更し、ポートは無許可ステートに戻ります。

マルチホストモードでは、複数のホストを単一の 802.1x 対応ポートに接続できます。このモードでは、接続されたクライアントのうち 1 つが許可されれば、クライアントすべてのネットワーク アクセスが許可されます。ポートが無許可ステートになると（再認証が失敗するか、または EAPOL-Logoff メッセージを受信した場合）、デバイスは接続しているクライアントのネットワークアクセスをすべて禁止します。

このトポロジでは、ワイヤレスアクセスポイントが接続しているクライアントの認証を処理し、デバイスに対してクライアントとしての役割を果たします。

図 *8 :* マルチ ホスト モードの例



（注） すべてのホスト モードで、ポートベース認証が設定されている場合、ライン プロトコルは許可の前にアップのままです。

デバイスはマルチドメイン認証（MDA）をサポートしています。これにより、データ装置と IP Phone などの音声装置（シスコ製品またはシスコ以外の製品）の両方を同じデバイスポートに接続できます。

802.1x マルチ認証モード

マルチ認証（multi-auth）モードでは、データ VLAN および音声 VLAN で複数のクライアントを認証できます。各ホストは個別に認証されます。マルチ認証ポートで認証できるデータデバイスまたは音声デバイスの数には制限はありません。

ハブまたはアクセス ポイントが 802.1x 対応ポートに接続されている場合、接続されている各クライアントを認証する必要があります。802.1x 以外のデバイスでは、MAC 認証バイパスまたは Web 認証をホスト単位認証フォールバック メソッドとして使用し、単一のポートで異なる方法で異なるホストを認証できます。

（注） ポートがマルチ認証モードの場合、認証失敗 VLAN 機能はアクティブになりません。

次の条件で、RADIUS サーバから提供された VLAN をマルチ認証モードで割り当てることができます。

- ホストがポートで最初に許可されたホストであり、RADIUS サーバが VLAN 情報を提供している。
- 後続のホストが、動作 VLAN に一致する VLAN を使用して許可される。
- ホストは VLAN が割り当てられていないポートで許可され、後続のホストでは VLAN 割り当てが設定されていないか、VLAN 情報が動作 VLAN と一致している。
- ポートで最初に許可されたホストにはグループ VLAN が割り当てられ、後続のホストでは VLAN 割り当てが設定されていないか、グループ VLAN がポート上のグループ VLAN と一致している。後続のホストが、最初のホストと同じ VLAN グループの VLAN を使用する必要がある。VLAN リストが使用されている場合、すべてのホストは VLAN リストで指定された条件に従う。
- VLAN がポート上のホストに割り当てられると、後続のホストは一致する VLAN 情報を持つ必要があり、この情報がなければポートへのアクセスを拒否される。
- ゲスト VLAN または認証失敗 VLAN をマルチ認証モードに設定できない。
- クリティカル認証 VLAN の動作が、マルチ認証モード用に変更されない。ホストが認証を試みたときにサーバに到達できない場合、許可されたすべてのホストは、設定された VLAN で再初期化される。

MAC 移動

あるスイッチ ポートで MAC アドレスが認証されると、そのアドレスは同じスイッチの別の認証マネージャ対応ポートでは許可されません。スイッチが同じ MAC アドレスを別の認証マネージャ対応ポートで検出すると、そのアドレスは許可されなくなります。

場合によっては、MAC アドレスを同じスイッチ上のポート間で移動する必要があります。たとえば、認証ホストとスイッチ ポート間に別のデバイス（ハブまたは IP Phone など）がある場合、ホストをデバイスから接続して、同じスイッチの別のポートに直接接続する必要があります。

デバイスが新しいポートで再認証されるように、MAC 移動をグローバルにイネーブルにできます。ホストが 2 番目のポートに移動すると、最初のポートのセッションが削除され、ホストは新しいポートで再認証されます。MAC 移動はすべてのホスト モードでサポートされます（認証ホストは、ポートでイネーブルにされているホスト モードに関係なく、スイッチの任意のポートに移動できます）。MAC アドレスがあるポートから別のポートに移動すると、スイッチは元のポートで認証済みセッションを終了し、新しいポートで新しい認証シーケンスを開始します。MAC 移動の機能は、音声およびデータ ホストの両方に適用されます。

（注） オープン認証モードでは、MACアドレスは、新しいポートでの許可を必要とせずに、元のポートから新しいポートへただちに移動します。

MAC 置換

MAC 置換機能は、ホストが別のホストがすでに認証済みであるポートに接続しようとすると発生する違反に対処するように設定できます。

（注） 違反はマルチ認証モードでは発生しないため、マルチ認証モードのポートにこの機能は適用されません。マルチホストモードで認証が必要なのは最初のホストだけなので、この機能はこのモードのポートには適用されません。

replace キーワードを指定して **authentication violation** インターフェイス コンフィギュレーション コマンドを設定すると、マルチドメインモードのポートでの認証プロセスは、次のようになります。

- 既存の認証済みMACアドレスを使用するポートで新しいMACアドレスが受信されます。
- 認証マネージャは、ポート上の現在のデータ ホストの MAC アドレスを、新しい MAC アドレスで置き換えます。
- 認証マネージャは、新しい MAC アドレスに対する認証プロセスを開始します。
- 認証マネージャによって新しいホストが音声ホストであると判断された場合、元の音声ホストは削除されます。

ポートがオープン認証モードになっている場合、MACアドレスはただちにMACアドレステーブルに追加されます。

802.1x アカウンティング

802.1x 標準では、ユーザの認証およびユーザのネットワーク アクセスに対する許可方法を定義しています。ただし、ネットワークの使用法についてはトラッキングしません。802.1xアカウンティングは、デフォルトでディセーブルです。802.1xアカウンティングをイネーブルにすると、次の処理を 802.1x 対応のポート上でモニタできます。

- 正常にユーザを認証します。
- ユーザがログ オフします。
- リンクダウンが発生します。
- 再認証が正常に行われます。
- 再認証が失敗します。

デバイスは802.1xアカウンティング情報を記録しません。その代わり、スイッチはこの情報をRADIUSサーバに送信します。RADIUSサーバは、アカウンティング メッセージを記録するように設定する必要があります。

802.1x アカウンティング属性値ペア

RADIUS サーバに送信された情報は、属性値（AV）ペアの形式で表示されます。これらの AV ペアのデータは、各種アプリケーションによって使用されます（たとえば課金アプリケーションの場合、RADIUS パケットの Acct-Input-Octets または Acct-Output-Octets 属性の情報が必要です）。

AV ペアは、802.1x アカウンティングが設定されているデバイスによって自動的に送信されます。次の種類の RADIUS アカウンティングパケットがデバイスによって送信されます。

- START：新規ユーザセッションの開始時に送信されます
- INTERIM：既存のセッション中にアップデートのために送信されます
- STOP：セッションが終了すると送信されます

デバイスによって送信された AV ペアは、**debug radius accounting** 特権 EXEC コマンドを入力して表示できます。

次の表に、AV ペアおよびデバイスによって送信される AV ペアの条件を示します。

表 *21 :* アカウンティング *AV* ペア

属性番号	AV ペア名	START	INTERIM	STOP
属性 [1]	User-Name	送信	送信	送信
属性 [4]	NAS-IP-Address	送信	送信	送信
属性 [5]	NAS-Port	送信	送信	送信
属性 [8]	Framed-IP-Address	非送信	条件に応じて送信[1]	条件に応じて送信
属性 [25]	Class	送信	送信	送信
属性 [30]	Called-Station-ID	送信	送信	送信
属性 [31]	Calling-Station-ID	送信	送信	送信
属性 [40]	Acct-Status-Type	送信	送信	送信
属性 [41]	Acct-Delay-Time	送信	送信	送信
属性 [42]	Acct-Input-Octets	非送信	送信	送信
属性 [43]	Acct-Output-Octets	非送信	送信	送信

属性番号	AV ペア名	START	INTERIM	STOP
属性 [47]	Acct-Input-Packets	非送信	送信	送信
属性 [48]	Acct-Output-Packets	非送信	送信	送信
属性 [44]	Acct-Session-ID	送信	送信	送信
属性 [45]	Acct-Authentic	送信	送信	送信
属性 [46]	Acct-Session-Time	非送信	送信	送信
属性 [49]	Acct-Terminate-Cause	非送信	非送信	送信
属性 [61]	NAS-Port-Type	送信	送信	送信

[1] 有効な静的 IP アドレスが設定されているか、ホストに対する Dynamic Host Control Protocol（DHCP）バインディングが DHCP スヌーピング バインディング テーブルに存在している場合に、Framed-IP-Address の AV ペアが送信されます。

デバイスと RADIUS サーバの通信

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 UDP ポート番号、または IP アドレスと特定の UDP ポート番号によって識別します。IP アドレスと UDP ポート番号の組み合わせによって、一意の ID が作成され、同一 IP アドレスのサーバ上にある複数の UDP ポートに RADIUS 要求を送信できるようになります。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（たとえば認証）を設定した場合、2 番めに設定されたホスト エントリは、最初に設定されたホスト エントリのフェールオーバー バックアップとして動作します。RADIUS ホスト エントリは、設定した順序に従って試行されます。

802.1X 認証

802.1x 認証を設定する場合の注意事項は、次のとおりです。

- 802.1x 認証をイネーブルにすると、他のレイヤ 2 またはレイヤ 3 機能がイネーブルになる前に、ポートが認証されます。
- 802.1x 対応ポートが割り当てられている VLAN が変更された場合、この変更は透過的でデバイスには影響しません。たとえば、ポートが RADIUS サーバに割り当ててられた VLAN に割り当てられ、再認証後に別の VLAN に割り当てられた場合に、この変更が発生します。

  802.1x ポートが割り当てられている VLAN がシャットダウン、ディセーブル、または削除される場合、ポートは無許可になります。たとえば、ポートが割り当てられたアクセス VLAN がシャットダウンまたは削除された後、ポートは無許可になります。

- 802.1x プロトコルは、レイヤ 2 スタティックアクセス ポート、音声 VLAN ポート、およびレイヤ 3 ルーテッド ポートでサポートされますが、次のポート タイプではサポートされません。
  - ダイナミックポート：ダイナミックモードのポートは、トランクポートへの変更を、ネイバーとネゴシエートする場合があります。ダイナミック ポートで 802.1x 認証をイネーブルにしようとすると、エラー メッセージが表示され、802.1x 認証はイネーブルになりません。802.1x 対応ポートのモードをダイナミックに変更しようとしても、エラー メッセージが表示され、ポート モードは変更されません。
  - EtherChannel ポート：アクティブまたはアクティブでない EtherChannel メンバを 802.1x ポートとして設定しないでください。EtherChannel ポートで 802.1x 認証をイネーブルにしようとすると、エラー メッセージが表示され、802.1x 認証はイネーブルになりません。
  - スイッチドポートアナライザ（SPAN）宛先ポート：SPAN 宛先ポートであるポートの 802.1x 認証をイネーブルにすることができます。ただし、ポートを SPAN 宛先ポートとして削除するまでは、802.1x 認証はディセーブルになります。SPAN 送信元ポートでは 802.1x 認証をイネーブルにすることができます。
- デバイス上で、**dot1x system-auth-control** グローバル コンフィギュレーション コマンドを入力して 802.1x 認証をグローバルにイネーブルにする前に、802.1x 認証と EtherChannel が設定されているインターフェイスから、EtherChannel の設定を削除してください。

802.1x 認証のデフォルト設定

表 *22 : 802.1x* 認証のデフォルト設定

機能	デフォルト設定
デバイスの 802.1x イネーブルステート	ディセーブル
ポート単位の 802.1x イネーブル ステート	ディセーブル（force-authorized） ポートはクライアントとの 802.1x ベース認証を行わずに、通常のトラフィックを送受信します。
AAA	ディセーブル
RADIUS サーバ • IP アドレス • UDP 認証ポート • デフォルトのアカウンティング ポート • キー	• 指定なし • 1645 • 1646 • 指定なし

機能	デフォルト設定
ホスト モード	シングル ホスト モード
制御方向	双方向制御
定期的な再認証	ディセーブル
再認証の間隔（秒）	3600 秒
再認証回数	2回（ポートが無許可ステートに変わる前に、デバイスが認証プロセスを再開する回数）
待機時間	60 秒（デバイスがクライアントとの認証情報の交換に失敗した後、待機状態を続ける秒数）
再送信時間	30 秒（デバイスが EAP-Request/Identity フレームに対するクライアントからの応答を待ち、要求を再送信するまでの秒数）
最大再送信回数	2回（デバイスが認証プロセスを再開するまでに、EAP-Request/Identity フレームを送信する回数）
クライアント タイムアウト時間	30 秒（認証サーバからの要求をクライアントにリレーするとき、デバイスが返答を待ち、クライアントに要求を再送信するまでの時間）
認証サーバ タイムアウト時間	30 秒（クライアントからの応答を認証サーバにリレーするとき、デバイスが応答を待ち、サーバに応答を再送信するまでの時間） dot1x timeout server-timeout インターフェイス コンフィギュレーション コマンドを使用して、このタイムアウト時間を変更できます。
無活動タイムアウト	ディセーブル
ゲスト VLAN	指定なし
アクセス不能認証バイパス	ディセーブル
制限付き VLAN	指定なし
オーセンティケータ（スイッチ）モード	指定なし
MAC 認証バイパス	ディセーブル
音声認識セキュリティ	ディセーブル

柔軟な認証の順序設定

柔軟な認証の順序設定を使用して、ポートが新しいホストを認証するときに使用する方法の順序を設定できます。The IEEE 802.1X の柔軟な認証機能では、以下の 3 つの認証方法をサポートしています。

- dot1X：IEEE 802.1X 認証はレイヤ 2 の認証方式です。
- mab：MAC 認証バイパスはレイヤ 2 の認証方式です。
- webauth：Web 認証はレイヤ 3 の認証方式です。

これらの機能を使用すると、各ポートでどの認証方式を使用するかを制御できます。また、そのポートの方式についてフェールオーバー順も制御できます。たとえば、MAC 認証バイパスおよび 802.1x は、プライマリまたはセカンダリ認証方法として使用し、Web 認証は、これらの認証のいずれか、または両方が失敗した場合のフォールバック方法として使用できます。

The IEEE 802.1X の柔軟な認証機能では、以下のホスト モードをサポートしています。

- multi-auth：マルチ認証では、音声 VLAN に 1 つの認証、データ VLAN に複数の認証を使用できます。
- multi-domain：マルチドメイン認証では、音声 VLAN に 1 つ、データ VLAN に 1 つの、2 つの認証を使用できます。

VLAN 割り当てを使用した 802.1x 認証

デバイスは、VLAN 割り当てを使用した 802.1x 認証をサポートしています。ポートの 802.1x 認証が成功すると、RADIUS サーバは VLAN 割り当てを送信してデバイスポートを設定します。RADIUS サーバデータベースは、ユーザ名と VLAN のマッピングを維持し、デバイスポートに接続するクライアントのユーザ名に基づいて VLAN を割り当てます。この機能を使用して、特定のユーザのネットワーク アクセスを制限できます。

音声デバイス認証は、マルチドメイン ホスト モードでサポートされます。音声デバイスが許可されているときに、RADIUS サーバから許可された VLAN が返された場合、このポートの音声 VLAN は、割り当てられた音声 VLAN でパケットを送受信するように設定されています。音声 VLAN 割り当ては、マルチドメイン認証（MDA）対応のポートでのデータ VLAN 割り当てと同じように機能します。

デバイスと RADIUS サーバ上で設定された場合、VLAN 割り当てを使用した 802.1x 認証には次の特性があります。

- RADIUS サーバから VLAN が提供されない場合、または 802.1x 認証がディセーブルの場合、認証が成功するとポートはアクセス VLAN に設定されます。アクセス VLAN とは、アクセス ポートに割り当てられた VLAN です。このポート上で送受信されるパケットはすべて、この VLAN に所属します。

- 802.1x 認証がイネーブルで、RADIUS サーバからの VLAN 情報が有効でない場合、認証に失敗して、設定済みの VLAN が引き続き使用されます。これにより、設定エラーによって不適切な VLAN に予期せぬポートが現れることを防ぎます。

  設定エラーには、ルーテッド ポートの VLAN、間違った VLAN ID、存在しないまたは内部（ルーテッド ポート）の VLAN ID、RSPAN VLAN、シャットダウンしている VLAN、あるいは一時停止している VLAN ID の指定などがあります。マルチドメイン ホスト ポートの場合、設定エラーには、設定済みまたは割り当て済み VLAN ID と一致するデータ VLAN の割り当て試行（またはその逆）のために発生するものもあります。

- 802.1x 認証がイネーブルで、RADIUS サーバからのすべての情報が有効の場合、許可されたデバイスは認証後、指定した VLAN に配置されます。

- 802.1x ポートでマルチ ホスト モードがイネーブルの場合、すべてのホストは最初に認証されたホストと同じ VLAN（RADIUS サーバにより指定）に配置されます。

- ポート セキュリティをイネーブル化しても、RADIUS サーバが割り当てられた VLAN の動作には影響しません。

- 802.1x 認証がポートでディセーブルの場合、設定済みのアクセス VLAN と設定済みの音声 VLAN に戻ります。

- 802.1x ポートが認証され、RADIUS サーバによって割り当てられた VLAN に配置されると、そのポートのアクセス VLAN 設定への変更は有効になりません。マルチドメイン ホストの場合、ポートが完全にこれらの例外で許可されている場合、同じことが音声デバイスに適用されます。

  - あるデバイスで VLAN 設定を変更したことにより、他のデバイスに設定済みまたは割り当て済みの VLAN と一致した場合、ポート上の全デバイスの認証が中断して、データおよび音声デバイスに設定済みの VLAN が一致しなくなるような有効な設定が復元されるまで、マルチドメイン ホスト モードがディセーブルになります。

  - 音声デバイスが許可されて、ダウンロードされた音声 VLAN を使用している場合、音声 VLAN 設定を削除したり設定値を dot1p または untagged に修正したりすると、音声デバイスが未許可になり、マルチドメインホストモードがディセーブルになります。

ポートが、強制許可（force-authorized）ステート、強制無許可（force-unauthorized）ステート、無許可ステート、またはシャットダウン ステートの場合、ポートは設定済みのアクセス VLAN に配置されます。

802.1x ポートが認証され、RADIUS サーバによって割り当てられた VLAN に配置されると、そのポートのアクセス VLAN 設定への変更は有効になりません。マルチドメイン ホストの場合、ポートが完全にこれらの例外で許可されている場合、同じことが音声デバイスに適用されます。

- あるデバイスで VLAN 設定を変更したことにより、他のデバイスに設定済みまたは割り当て済みの VLAN と一致した場合、ポート上の全デバイスの認証が中断して、データおよび音声デバイスに設定済みの VLAN が一致しなくなるような有効な設定が復元されるまで、マルチドメイン ホスト モードがディセーブルになります。

- 音声デバイスが許可されて、ダウンロードされた音声 VLAN を使用している場合、音声 VLAN 設定を削除したり設定値を *dot1p* または *untagged* に修正したりすると、音声デバイスが未許可になり、マルチドメインホストモードがディセーブルになります。

ポートが、強制許可（force-authorized）ステート、強制無許可（force-unauthorized）ステート、無許可ステート、またはシャットダウン ステートの場合、ポートは設定済みのアクセス VLAN に配置されます。

VLAN 割り当てを設定するには、次の作業を実行する必要があります。

- **network** キーワードを使用して AAA 認証をイネーブルにし、RADIUS サーバからのインターフェイス設定を可能にします。
- 802.1x 認証をイネーブルにします。（アクセス ポートで 802.1x 認証を設定すると、VLAN 割り当て機能は自動的にイネーブルになります）。
- RADIUS サーバにベンダー固有のトンネル属性を割り当てます。RADIUS サーバは次の属性をデバイスに返す必要があります。
  - [64] Tunnel-Type = VLAN
  - [65] Tunnel-Medium-Type = 802
  - [81] Tunnel-Private-Group-ID = VLAN 名または VLAN ID
  - [83] Tunnel-Preference

  属性 [64] は、値 *VLAN*（タイプ 13）でなければなりません。属性 [65] は、値*802*（タイプ 6）でなければなりません。属性 [81] は、IEEE 802.1x 認証ユーザに割り当てられた *VLAN* 名または *VLAN ID* を指定します。

ゲスト VLAN を使用した 802.1x 認証

デバイス上の各 802.1x ポートにゲスト VLAN を設定し、クライアントに対して限定的なサービスを提供できます（802.1x クライアントのダウンロードなど）。これらのクライアントは 802.1x 認証用にシステムをアップグレードできる場合がありますが、一部のホスト（Windows 98 システムなど）は IEEE 802.1x 対応ではありません。

デバイスが EAP Request/Identity フレームに対する応答を受信していない場合、または EAPOL パケットがクライアントによって送信されない場合に、802.1x ポート上でゲスト VLAN をイネーブルにすると、デバイスはクライアントにゲスト VLAN を割り当てます。

デバイスはEAPOLパケット履歴を保持します。EAPOLパケットがリンクの存続時間中にインターフェイスで検出された場合、デバイスはそのインターフェイスに接続されているデバイスが IEEE 802.1x 対応のサプリカントであると判断します。インターフェイスはゲスト VLAN ステートにはなりません。インターフェイスのリンク ステータスがダウンした場合、EAPOL 履歴はクリアされます。EAPOLパケットがインターフェイスで検出されない場合、そのインターフェイスはゲスト VLAN のステートになります。

デバイスが 802.1x 対応の音声デバイスを許可しようとしたが、AAA サーバが使用できない場合、許可は失敗します。ただし、EAPOLパケットの検出はEAPOL履歴に保存されます。この音声デバイスは、AAA サーバが使用可能になると許可されます。ただし、他のデバイスによるゲスト VLAN へのアクセスは許可されなくなります。この状況を防ぐには、次のいずれかのコマンド シーケンスを使用します。

- **authentication event no-response action authorize vlan** *vlan-id* インターフェイス コンフィギュレーション コマンドを入力し、ゲスト VLAN へのアクセスを許可します。
- **shutdown** インターフェイス コンフィギュレーション コマンドを入力し、さらに **no shutdown** インターフェイス コンフィギュレーション コマンドを入力してポートを再起動します。

リンクの存続時間中にデバイスが EAPOL パケットを送信した場合、デバイスはゲスト VLAN への認証アクセスに失敗したクライアントを許可しません。

（注） インターフェイスがゲスト VLAN に変わってから EAPOL パケットが検出された場合、無許可ステートに戻って 802.1x 認証を再起動します。

デバイスポートがゲスト VLAN に変わると、802.1x 非対応クライアントはすべてアクセスを許可されます。ゲスト VLAN が設定されているポートに 802.1x 対応クライアントが加入すると、ポートは、ユーザ設定によるアクセス VLAN で無許可ステートになり、認証が再起動されます。

ゲスト VLAN は、単一のホスト、複数のホスト、複数認証、またはマルチドメイン モードにおける 802.1x ポートでサポートされています。

デバイスはMAC認証バイパスをサポートします。MAC認証バイパスが802.1xポートでイネーブルの場合、デバイスは、IEEE 802.1x 認証のタイムアウト時にEAPOL メッセージ交換を待機している間、クライアント MAC アドレスに基づいてクライアントを許可できます。デバイスは、802.1x ポート上のクライアントを検出したあとで、クライアントからのイーサネットパケットを待機します。デバイスは、MAC アドレスに基づいたユーザ名およびパスワードを持つRADIUS-Access/Requestフレームを認証サーバに送信します。許可に成功した場合、デバイスはクライアントにネットワークへのアクセスを許可します。許可に失敗した場合、ゲスト VLAN が指定されていれば、デバイスはポートをゲスト VLAN に割り当てます。

制限付き VLAN を使用した 802.1x 認証

ゲスト VLAN にアクセスできないクライアント向けに、限定されたサービスを提供するために、デバイスの各 IEEE 802.1x ポートに対して制限付き VLAN（認証失敗 VLAN と呼ばれることもあります）を設定できます。これらのクライアントは、認証プロセスに失敗したため他の VLAN にアクセスできない 802.1x 対応クライアントです。制限付き VLAN を使用すると、認証サーバの有効なクレデンシャルを持っていないユーザ（通常、企業にアクセスするユーザ）に、サービスを制限したアクセスを提供できます。管理者は制限付き VLAN のサービスを制御できます。

（注） 両方のタイプのユーザに同じサービスを提供する場合、ゲスト VLAN と制限付き VLAN の両方を同じに設定できます。

この機能がないと、クライアントは認証失敗を永遠に繰り返すことになるため、デバイスポートがスパニングツリーのブロッキングステートから変わることができなくなります。この機能を使用することで、クライアントの認証試行回数を指定し（デフォルト値は3回）、一定回数後にデバイスポートを制限付き VLAN の状態に移行させることができます。

認証サーバはクライアントの認証試行回数をカウントします。このカウントが設定した認証試行回数を超えると、ポートが制限付き VLAN の状態に変わります。失敗した試行回数は、RADIUS サーバが *EAP failure* で応答したときや、EAP パケットなしの空の応答を返したときからカウントされます。ポートが制限付き VLAN に変わったら、このカウント数はリセットされます。

認証に失敗したユーザのVLANは、もう一度認証を実行するまで制限された状態が続きます。VLAN 内のポートは設定された間隔に従って再認証を試みます（デフォルトは 60 秒）。再認証に失敗している間は、ポートの VLAN は制限された状態が続きます。再認証に成功した場合、ポートは設定された VLAN もしくは RADIUS サーバによって送信された VLAN に移行します。再認証はディセーブルにすることもできますが、ディセーブルにすると、*link down* または *EAP logoff* イベントを受信しない限り、ポートの認証プロセスを再起動できません。クライアントがハブを介して接続している場合、再認証機能はイネーブルにしておくことを推奨します。クライアントの接続をハブから切り離すと、ポートに *link down* や *EAP logoff* イベントが送信されない場合があります。

ポートが制限付き VLAN に移行すると、EAP 成功の疑似メッセージがクライアントに送信されます。このメッセージによって、繰り返し実行している再認証を停止させることができます。クライアントによっては（Windows XP が稼働しているデバイスなど）、EAP なしで DHCP を実装できません。

制限付き VLAN は、すべてのホスト モードでの 802.1x ポート上、およびレイヤ 2 ポート上でサポートされます。

ダイナミック ARP インスペクション、DHCP スヌーピング、IP 送信元ガードなどの他のセキュリティ ポート機能は、制限付き VLAN に対して個別に設定できます。

802.1X 認証失敗 VLAN

ゲスト VLAN にアクセスできないクライアント向けに、限定されたサービスを提供するために、デバイスの各 802.1X ポートに対して認証失敗 VLAN を設定できます。これらのクライアントは 802.1X 準拠で、認証プロセスに失敗しているため別の VLAN にアクセスすることができません。認証失敗 VLAN を使用すると、認証サーバの有効なクレデンシャルを持っていないユーザ（通常、企業にアクセスするユーザ）に、サービスを制限したアクセスを提供できます。管理者は認証失敗 VLAN のサービスを制御できます。

（注） 両方のタイプのユーザに同じサービスを提供する場合、ゲスト VLAN と認証失敗 VLAN の両方を同じに設定できます。

この機能がないと、クライアントは認証失敗を永遠に繰り返すことになるため、デバイスポートがスパニングツリーのブロッキングステートから変わることができなくなります。この機能を使用することで、クライアントの認証試行回数を指定し（デフォルト値は3回）、一定回数後にデバイスポートを認証失敗 VLAN の状態に移行させることができます。

認証サーバはクライアントの認証試行回数をカウントします。このカウントが設定した認証試行回数を超えると、ポートが認証失敗 VLAN の状態に変わります。失敗した試行回数は、RADIUS サーバが EAP failure で応答したときや、EAP パケットなしの空の応答を返したときからカウントされます。ポートが認証失敗 VLAN に変わったら、このカウント数はリセットされます。

認証に失敗したユーザのVLANは、もう一度認証を実行するまで認証失敗の状態が続きます。認証失敗 VLAN 内のポートは設定された間隔に従って再認証を試みます（デフォルトは 60 秒）。再認証に失敗している間は、ポートの VLAN は認証失敗の状態が続きます。再認証に成功した場合、ポートは設定されたVLANもしくはRADIUSサーバによって送信されたVLANに移行します。再認証はディセーブルにすることもできますが、ディセーブルにした場合、認証プロセスを再開する唯一の方法は、ポートでリンク ダウンまたは EAP ログオフ イベントを受信することです。クライアントがハブを介して接続している場合、再認証機能はイネーブルにしておくことを推奨します。クライアントがハブから切断されると、ポートがリンク ダウンや EAP ログオフ イベントを受信しない可能性があります。

ポートが認証失敗 VLAN に移行すると、EAP 成功の疑似メッセージがクライアントに送信されます。このメッセージによって、繰り返し実行している再認証を停止させることができます。

前提条件として、デバイスを Cisco Secure Access Control System（ACS）に接続し、RADIUS 認証、許可、およびアカウンティング（AAA）をWeb認証用に設定する必要があります。また、必要に応じて、ACL ダウンロードを有効にします。

Open1x 認証

Open1x 認証によって、デバイスが認証される前に、そのデバイスがポートにアクセスできるようになります。オープン認証が設定されている場合、新しいホストはポートに定義されているアクセス コントロール リスト（ACL）に基づいてトラフィックを渡します。ホストが認証されると、RADIUS サーバに設定されているポリシーがそのホストに適用されます。

オープン認証を次の状況で設定できます。

- シングルホストモードでのオープン認証：1 人のユーザだけが認証の前後にネットワークにアクセスできます。
- MDA モードでのオープン認証：音声ドメインの 1 人のユーザだけ、およびデータドメインの 1 人のユーザだけが許可されます。

- マルチホストモードでのオープン認証：任意のホストがネットワークにアクセスできます。
- 複数認証モードでのオープン認証：MDA の場合と似ていますが、複数のホストを認証できます。

注 オープン認証が設定されている場合は、他の認証制御よりも優先されます。これは、**authentication open** インターフェイス コンフィギュレーション コマンドを使用した場合、**authentication port-control** インターフェイス コンフィギュレーション コマンドに関係なく、ポートがホストにアクセス権を付与することを意味します。

ユーザのログイン制限

ログイン制限機能では、ネットワーク管理者が、ユーザによるネットワークへのログイン試行を制限することができます。ユーザによるネットワークへのログインの試行が、設定可能な時間制限内かつ設定可能な回数以内に成功しなかった場合、そのユーザをブロックできます。この機能は、ローカル ユーザに対してだけ有効であり、リモート ユーザは利用できません。この機能を有効にするには、グローバル コンフィギュレーション モードで **aaa authentication rejected** コマンドを設定する必要があります。

アクセス不能認証バイパスを使用した 802.1x 認証

デバイスが設定されたRADIUSサーバに到達できず、新しいホストを認証できない場合、アクセス不能認証バイパス機能を使用します。この機能は、クリティカル認証またはAAA失敗ポリシーとも呼ばれます。これらのホストをクリティカルポートに接続するようにデバイスを設定できます。

新しいホストがクリティカルポートに接続しようとすると、そのホストはユーザ指定のアクセスVLAN、クリティカル*VLAN*に移動されます。管理者はこれらのホストに制限付き認証を付与します。

デバイスは、クリティカルポートに接続されているホストを認証しようとする場合、設定されているRADIUSサーバのステータスをチェックします。利用可能なサーバが1つあれば、デバイスはホストを認証できます。ただし、すべてのRADIUSサーバが利用不可能な場合は、デバイスはホストへのネットワークアクセスを許可して、ポートを認証ステートの特別なケースであるクリティカル認証ステートにします。

（注） クリティカル認証をインターフェイスで設定する場合は、クリティカル承認（クリティカル vlan）に使用する vlan をデバイスでアクティブにする必要があります。クリティカル vlan が非アクティブまたはダウンしていると、クリティカル認証セッションは非アクティブな VLAN のイネーブル化を試行し続け、繰り返し失敗します。これは大量のメモリ保持の原因となる可能性があります。

アクセス不能認証バイパスの認証結果

アクセス不能認証バイパス機能の動作は、ポートの許可ステートにより異なります。

- クリティカルポートに接続されているホストが認証しようとする際にポートが無許可ですべてのサーバが利用できない場合、デバイスは RADIUS 設定済み VLAN またはユーザ指定のアクセス VLAN にあるポートをクリティカル認証ステートにします。
- ポートが許可済みで、再認証が行われた場合、デバイスは現在の VLAN（事前に RADIUS サーバにより割り当てられた可能性があるもの）でクリティカルポートをクリティカル認証ステートにします。
- 認証交換中に RADIUS サーバが使用不可能となった場合、現在の交換はタイムアウトになり、デバイスは次の認証試行の間にクリティカルポートをクリティカル認証ステートにします。

RADIUS サーバが再び使用可能になったときにホストを再初期化し、クリティカル VLAN から移動するように、クリティカルポートを設定できます。このように設定した場合、クリティカル認証ステートのすべてのクリティカル ポートは自動的に再認証されます。

アクセス不能認証バイパス機能の相互作用

アクセス不能認証バイパスは、次の機能と相互に作用します。

- ゲスト VLAN：アクセス不能認証バイパスは、ゲスト VLAN と互換性があります。ゲスト VLAN が 8021.x ポートでイネーブルの場合、この機能は次のように相互に作用します。
  - デバイスが EAP Request/Identity フレームへの応答を受信しないとき、または EAPOL パケットがクライアントによって送信されないときに、少なくとも 1 つの RADIUS サーバが使用できれば、デバイスはクライアントにゲスト VLAN を割り当てます。
  - すべての RADIUS サーバが使用できず、クライアントがクリティカルポートに接続されている場合、デバイスはクライアントを認証して、クリティカルポートを RADIUS 設定済み VLAN またはユーザ指定のアクセス VLAN でクリティカル認証ステートにします。
  - すべての RADIUS サーバが使用できず、クライアントがクリティカルポートに接続されていない場合、ゲスト VLAN が設定されていても、デバイスはクライアントにゲスト VLAN を割り当てられません。

- すべてのRADIUSサーバが使用できず、クライアントがクリティカルポートに接続されていて、すでにゲストVLANが割り当てられている場合、デバイスはそのポートをゲスト VLAN に保持します。

- 制限付き VLAN：ポートがすでに制限付き VLAN で許可されていて RADIUS サーバが使用できない場合、デバイスはクリティカルポートを制限付き VLAN でクリティカル認証ステートにします。
- 802.1xアカウンティング：RADIUSサーバが使用できない場合、アカウンティングは影響を受けません。
- 音声VLAN：アクセス不能認証バイパスは音声VLANと互換性がありますが、RADIUS設定済み VLAN またはユーザ指定のアクセス VLAN は、音声 VLAN と異なっていなければなりません。
- Remote Switched Port Analyzer（RSPAN）：アクセス不能認証バイパスの RADIUS 設定済み VLAN またはユーザ指定のアクセス VLAN として RSPAN VLAN を指定しないでください。

複数認証ポートのアクセス不能認証バイパスのサポート

ポートが任意のホストモードで設定されていて、AAA サーバを使用できない場合、ポートはマルチホストモードに設定され、クリティカルVLANに移動されます。複数認証（multi-auth）ポートで、このアクセス不能バイパスをサポートするには、**authentication event server dead action reinitialize vlan** *vlan-id* コマンドを使用します。新しいホストがクリティカル ポートに接続しようとすると、そのポートは再初期化され、接続されているすべてのホストがユーザ指定のアクセス VLAN に移動されます。

このコマンドは、すべてのホスト モードでサポートされます。

VLAN 割り当て、ゲスト VLAN、制限付き VLAN、アクセス不能認証バイパス

VLAN割り当て、ゲストVLAN、制限付きVLAN、およびアクセス不能認証バイパス設定時の注意事項は、次のとおりです。

- 802.1x 認証をポート上でイネーブルにすると、音声 VLAN の機能を持つポート VLAN は設定できません。
- トランクポートまたはダイナミックポートの場合、VLAN割り当て機能を使用した802.1x認証はサポートされません。
- 音声 VLAN を除くあらゆる VLAN を、802.1x ゲスト VLAN として設定できます。ゲスト VLAN 機能は、内部 VLAN（ルーテッド ポート）またはトランク ポートではサポートされていません。アクセス ポート上でだけサポートされます。
- DHCP クライアントが接続されている 802.1x ポートのゲスト VLAN を設定した後、DHCP サーバからホスト IP アドレスを取得する必要があります。クライアント上の DHCP プロ

セスが時間切れとなり DHCP サーバからホスト IP アドレスを取得しようとする前に、デバイス上の 802.1x 認証プロセスを再起動する設定を変更できます。802.1x 認証プロセスの設定を減らします（**authentication timer reauthenticate** インターフェイス コンフィギュレーション コマンド）。設定の減少量は、接続された 802.1x クライアントのタイプによって異なります。

- アクセス不能認証バイパス機能を設定する際には、次の注意事項に従ってください。
  - この機能はシングル ホスト モードおよびマルチホスト モードの 802.1x ポートでサポートされます。
  - アクセス不能認証バイパス機能および制限付き VLAN を 802.1x ポート上に設定できます。デバイスが制限付き VLAN 内でクリティカルポートを再認証しようとし、すべての RADIUS サーバが利用不可能な場合、デバイスはポートステートをクリティカル認証ステートに変更し、制限付き VLAN に残ります。
- 音声 VLAN を除くあらゆる VLAN を、802.1x 制限付き VLAN として設定できます。制限付き VLAN 機能は、内部 VLAN（ルーテッド ポート）またはトランク ポートではサポートされていません。アクセス ポート上でだけサポートされます。

MAC 認証バイパスを使用した IEEE 802.1x 認証

MAC 認証バイパス機能を使用し、クライアント MAC アドレスに基づいてクライアントを許可するようにデバイスを設定できます。たとえば、プリンタなどのデバイスに接続された IEEE 802.1x ポートでこの機能をイネーブルにできます。

クライアントからの EAPOL 応答の待機中に IEEE 802.1x 認証がタイムアウトした場合、デバイスは MAC 認証バイパスを使用してクライアントを許可しようとします。

MAC 認証バイパス機能が IEEE 802.1x ポートでイネーブルの場合、デバイスはクライアント ID として MAC アドレスを使用します。認証サーバには、ネットワーク アクセスを許可されたクライアント MAC アドレスのデータベースがあります。デバイスは、IEEE 802.1x ポート上のクライアントを検出した後で、クライアントからのイーサネットパケットを待機します。デバイスは、MAC アドレスに基づいたユーザ名およびパスワードを持つ RADIUS-Access/Request フレームを認証サーバに送信します。許可に成功した場合、デバイスはクライアントにネットワークへのアクセスを許可します。許可に失敗した場合、ゲスト VLAN が設定されていればデバイスはポートにゲスト VLAN を割り当てます。このプロセスは、ほとんどのクライアントデバイスで動作します。ただし、代替の MAC アドレス形式を使用しているクライアントでは動作しません。標準の形式とは異なる MAC アドレスを持つクライアントに対して MAB 認証をどのように実行するかや、RADIUS の設定のどこでユーザ名とパスワードが異なることが要求されるかを設定できます。

リンクのライフタイム中に EAPOL パケットがインターフェイス上で検出された場合、デバイスは、そのインターフェイスに接続されているデバイスが 802.1x 対応サプリカントであることを確認し、（MAC 認証バイパス機能ではなく）802.1x 認証を使用してインターフェイスを許可します。インターフェイスのリンク ステータスがダウンした場合、EAPOL 履歴はクリアされます。

デバイスがすでにMAC認証バイパスを使用してポートを許可し、IEEE 802.1xサプリカントを検出している場合、デバイスはポートに接続されているクライアントを許可します。再認証が発生するときに、Termination-Action RADIUS 属性値が DEFAULT であるために前のセッションが終了した場合、デバイスはポートに設定されている認証または再認証方式を使用します。

MAC 認証バイパスで認証されたクライアントは再認証できます。再認証プロセスは、IEEE 802.1xを使用して認証されたクライアントに対するプロセスと同じです。再認証中は、ポートは前に割り当てられた VLAN のままです。再認証に成功すると、デバイスはポートを同じ VLAN に保持します。再認証に失敗した場合、ゲスト VLAN が設定されていれば、デバイスはポートをゲスト VLAN に割り当てます。

再認証が Session-Timeout RADIUS 属性（Attribute[27]）、および Termination-Action RADIUS 属性（Attribute[29]）に基づいて行われるときに、Termination-Action RADIUS 属性（Attribute[29]）のアクションが *Initialize*（属性値は *DEFAULT*）である場合、MAC 認証バイパス セッションは終了し、再認証の間の接続は失われます。MAC 認証バイパス機能がイネーブルで IEEE 802.1x 認証がタイムアウトした場合、デバイスは MAC 認証バイパス機能を使用して再認証を開始します。これらの AV ペアの詳細については、RFC 3580『IEEE 802.1X Remote Authentication Dial In User Service (RADIUS)』を参照してください。

MAC 認証バイパスは、次の機能と相互に作用します。

- IEEE 802.1x 認証：802.1x 認証がポートで有効の場合にのみ MAC 認証バイパスを有効にできます。
- ゲスト VLAN：クライアントの MAC アドレス ID が無効な場合、ゲスト VLAN が設定されていれば、デバイスは VLAN にクライアントを割り当てます。
- 制限付き VLAN：IEEE 802.1x ポートに接続されているクライアントが MAC 認証バイパスで認証されている場合には、この機能はサポートされません。
- ポート セキュリティ
- 音声 VLAN
- Network Edge Access Topology（NEAT）：MAB と NEAT は相互に排他的です。インターフェイス上で NEAT が有効の場合は、MAB を有効にすることはできません。また、インターフェイス上で MAB が有効の場合は、NEAT を有効にすることはできません。

MAC 認証バイパスの注意事項

この項では、MAC 認証バイパス設定時の注意事項について説明します。

- 特に明記していないかぎり、MAC 認証バイパスの注意事項は 802.1x 認証のものと同じです。
- ポートが MAC アドレスで許可された後に、ポートから MAC 認証バイパスをディセーブルにしても、ポート ステートに影響はありません。
- ポートが未許可ステートであり、クライアント MAC アドレスが認証サーバ データベースにない場合、ポートは未許可ステートのままです。ただし、クライアント MAC アドレス

がデータベースに追加されると、デバイスは MAC 認証バイパス機能を使用してポートを再認証できます。

- ポートが認証ステートにない場合、再認証が行われるまでポートはこのステートを維持します。
- MAC 認証バイパスにより接続されているが、非アクティブなホストのタイムアウト時間を設定できます。指定できる範囲は 1 ～ 65535 秒です。

ポートあたりのデバイスの最大数

802.1x 対応のポートに接続できるデバイスの最大数は次のとおりです。

- シングル ホスト モードの場合、アクセス VLAN で接続できるデバイスは 1 台だけです。ポートが音声 VLAN でも設定されている場合、音声 VLAN を介して送受信できる Cisco IP Phone の数には制限はありません。
- マルチドメイン認証（MDA）モードの場合、アクセス VLAN で 1 台のデバイス、音声 VLAN で 1 台の IP Phone が許可されます。
- マルチホスト モードでは、1 つの 802.1x サプリカントだけがポートで許可されますが、非 802.1x ホストは数に制限なく、アクセス VLAN で許可されます。音声 VLAN で許可されるデバイスの数には制限はありません。

音声 VLAN ポートを使用した IEEE 802.1x 認証

音声 VLAN ポートは特殊なアクセス ポートで、次の 2 つの VLAN ID が対応付けられています。

- IP Phone との間で音声トラフィックを伝送する VVID。VVID は、ポートに接続された IP Phone を設定するために使用されます。
- IP Phone を通じて、デバイスと接続しているワークステーションとの間でデータトラフィックを伝送する PVID。PVID は、ポートのネイティブ VLAN です。

ポートの許可ステートにかかわらず、IP Phone は音声トラフィックに対して VVID を使用します。これにより、IP Phone は IEEE 802.1x 認証とは独立して動作できます。

シングル ホスト モードでは、IP Phone だけが音声 VLAN で許可されます。マルチ ホスト モードでは、サプリカントが PVID で認証された後、追加のクライアントがトラフィックを音声 VLAN 上で送信できます。マルチ ホスト モードがイネーブルの場合、サプリカント認証は PVID と VVID の両方に影響します。

リンクがあるとき、音声 VLAN ポートはアクティブになり、IP Phone からの最初の Cisco Discovery Protocol メッセージを受け取るとデバイスの MAC アドレスが表示されます。Cisco IP Phone は、他のデバイスから受け取った Cisco Discovery Protocol メッセージをリレーしません。その結果、複数の IP Phone が直列で接続されている場合、デバイスは直接接続されている 1 台だけを認識します。音声 VLAN ポートで IEEE 802.1x 認証がイネーブルの場合、デバイスは 2 ホップ以上離れた認識されない IP Phone からのパケットをドロップします。

IEEE 802.1x 認証をデバイスポート上でイネーブルにすると、音声 VLAN でもあるアクセスポート VLAN を設定できます。

IP Phone がシングルホストモードで 802.1x 対応のデバイスポートに接続されている場合、デバイスは認証を行わずに電話ネットワークアクセスを承認します。ポートでマルチドメイン認証（MDA）を使用して、データ デバイスと IP Phone などの音声デバイスの両方を認証することを推奨します。

（注） 音声 VLAN が設定され、Cisco IP Phone が接続されているアクセスポートで IEEE 802.1x 認証をイネーブルにした場合、Cisco IP Phone のデバイスへの接続が最大 30 秒間失われます。

ポート セキュリティを使用した IEEE 802.1x 認証

IEEE 802.1x ではポート単位（IP テレフォニーに MDA が設定されている場合は VLAN 単位）で単一の MAC アドレスが適用され、ポートセキュリティは冗長であり、場合によっては期待される IEEE 802.1x の動作と干渉することがあります。

IEEE 802.1x がイネーブルの場合に、ポートセキュリティをイネーブルにすることは推奨されません。

ポートベース認証プロセス

IEEE 802.1X ポートベース認証を設定するには、認証、認可、およびアカウンティング（AAA）を有効にし、認証方式リストを指定する必要があります。方式リストは、ユーザ認証のためにクエリー送信を行う手順と認証方式を記述したものです。

AAA プロセスは認証から始まります。802.1x ポートベース認証がイネーブルであり、クライアントが 802.1x 準拠のクライアント ソフトウェアをサポートしている場合、次のイベントが発生します。

- クライアント ID が有効で 802.1x 認証に成功した場合、デバイスはクライアントにネットワークへのアクセスを許可します。
- EAPOL メッセージ交換の待機中に 802.1x 認証がタイムアウトし、MAC 認証バイパスがイネーブルの場合、デバイスはクライアント MAC アドレスを認証用に使用します。このクライアント MAC アドレスが有効で認証に成功した場合、デバイスはクライアントにネットワークへのアクセスを許可します。クライアント MAC アドレスが無効で認証に失敗した場合、ゲスト VLAN が設定されていれば、デバイスはクライアントに限定的なサービスを提供するゲスト VLAN を割り当てます。
- デバイスが 802.1x 対応クライアントから無効な ID を取得し、制限付き VLAN が指定されている場合、デバイスはクライアントに限定的なサービスを提供する制限付き VLAN を割り当てることができます。
- RADIUS 認証サーバが使用できず（ダウンしていて）アクセスできない認証バイパスがイネーブルの場合、デバイスは、RADIUS 設定済み VLAN またはユーザ指定のアクセス

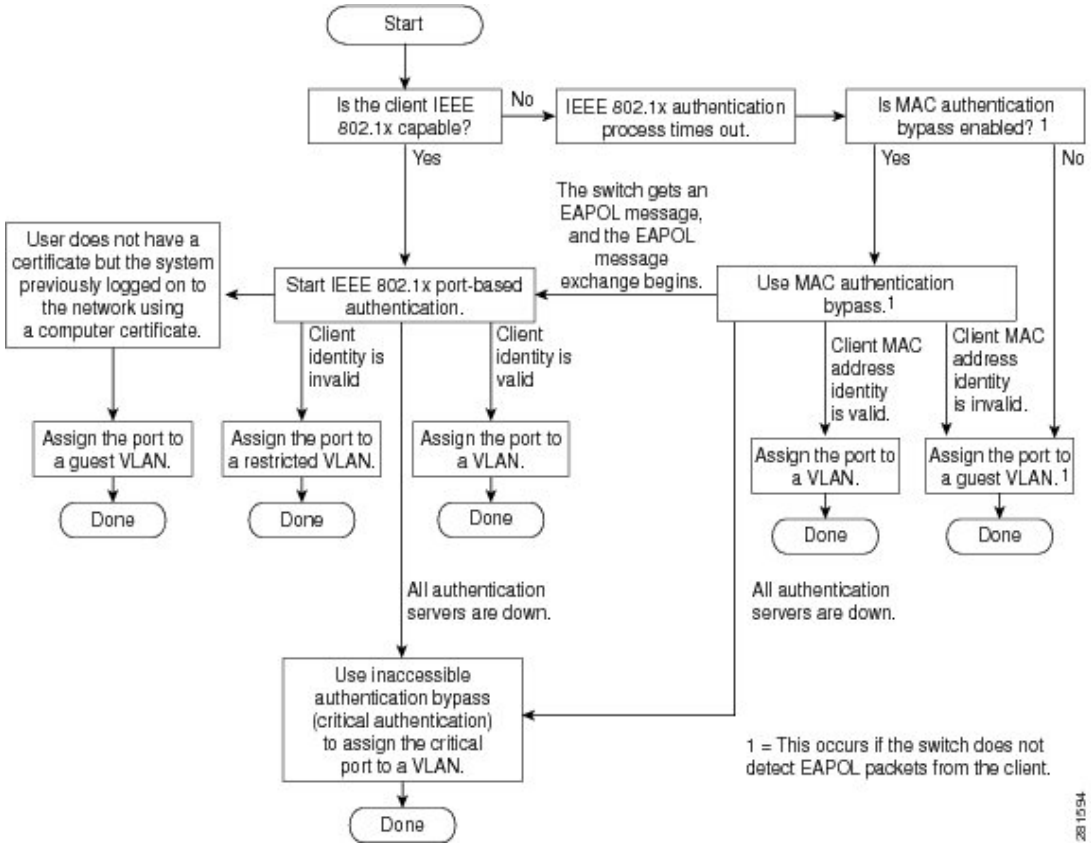
VLANで、ポートをクリティカル認証ステートにして、クライアントにネットワークへのアクセスを許可します。

(注) アクセスできない認証バイパスは、クリティカル認証、またはAAA失敗ポリシーとも呼ばれます。

ポートでMulti Domain Authentication（MDA）が有効になっている場合、音声許可に該当する例外をいくつか伴ったフローを使用できます。

図 *9*：認証フローチャート

次の図は認証プロセスを示します。



次の状況のいずれかが発生すると、デバイスはクライアントを再認証します。

- 定期的な再認証がイネーブルで、再認証タイマーの期限が切れている場合。

  デバイス固有の値を使用するか、RADIUSサーバからの値に基づいて再認証タイマーを設定できます。

  RADIUSサーバを使用した802.1x認証の後で、デバイスはSession-Timeout RADIUS属性（Attribute[27]）、およびTermination-Action RADIUS属性（Attribute[29]）に基づいてタイマーを使用します。

Session-Timeout RADIUS 属性（Attribute[27]）には再認証が行われるまでの時間を指定します。

Termination-Action RADIUS 属性（Attribute[29]）には、再認証中に行われるアクションを指定します。アクションは *Initialize* および *ReAuthenticate* に設定できます。アクションに *Initialize*（属性値は *DEFAULT*）を設定した場合、802.1x セッションは終了し、認証中、接続は失われます。アクションに *ReAuthenticate*（属性値は RADIUS-Request）を設定した場合、セッションは再認証による影響を受けません。

- クライアントを手動で再認証するには、**dot1x re-authenticate interface** *interface-id* 特権 EXEC コマンドを入力します。

ポートベース認証の開始およびメッセージ交換

802.1x 認証中に、デバイスまたはクライアントは認証を開始できます。**authentication port-control auto** インターフェイス コンフィギュレーション コマンドを使用してポート上で認証をイネーブルにすると、デバイスは、リンクステートがダウンからアップに移行したときに認証を開始し、ポートがアップしていて認証されていない場合は定期的に認証を開始します。デバイスはクライアントに EAP-Request/Identity フレームを送信し、識別情報を要求します。クライアントはフレームを受信すると、EAP-Response/Identity フレームで応答します。

ただし、クライアントが起動時にデバイスからの EAP-Request/Identity フレームを受信しなかった場合、クライアントは EAPOL-Start フレームを送信して認証を開始できます。このフレームはデバイスに対し、クライアントの識別情報を要求するように指示します。

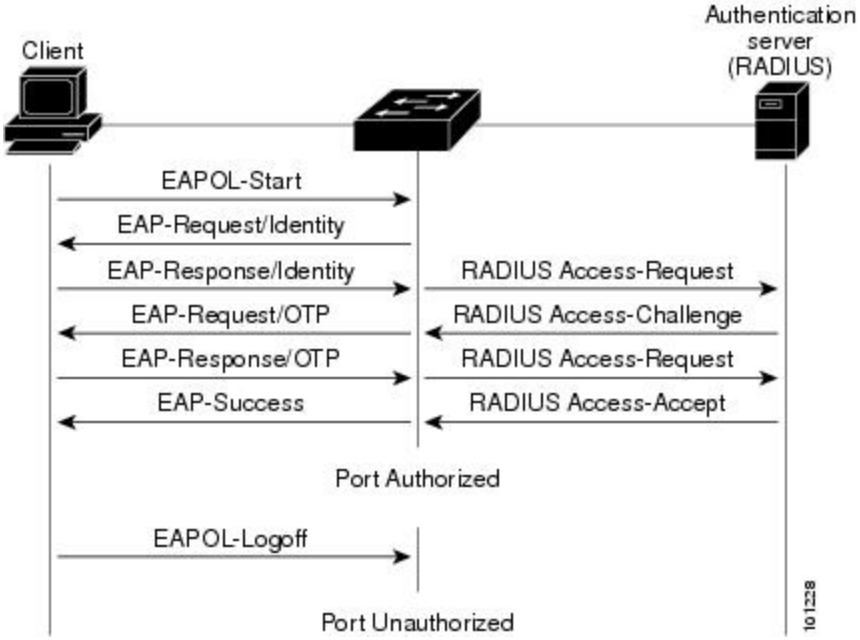
（注） ネットワーク アクセス デバイスで 802.1x 認証がイネーブルに設定されていない、またはサポートされていない場合には、クライアントからの EAPOL フレームはすべて廃棄されます。クライアントが認証の開始を 3 回試みても EAP-Request/Identity フレームを受信しなかった場合、クライアントはポートが許可ステートであるものとしてフレームを送信します。ポートが許可ステートであるということは、クライアントの認証が成功したことを実質的に意味します。

クライアントが自らの識別情報を提示すると、デバイスは仲介デバイスとしての役割を開始し、認証が成功または失敗するまで、クライアントと認証サーバの間で EAP フレームを送受信します。認証が成功すると、ポートは許可ステートになります。認証に失敗した場合、認証が再試行されるか、ポートが限定的なサービスを提供する VLAN に割り当てられるか、あるいはネットワーク アクセスが許可されないかのいずれかになります。

実際に行われる EAP フレーム交換は、使用する認証方式によって異なります。

図 *10* : メッセージ交換

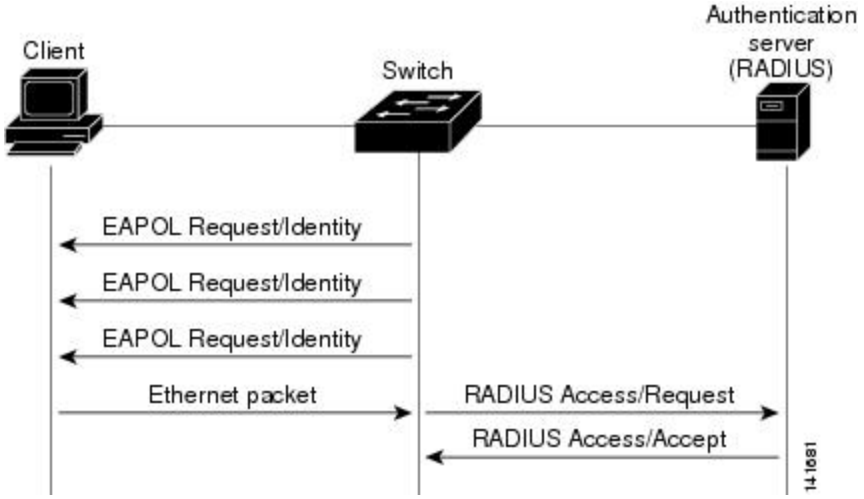
次の図に、クライアントが RADIUS サーバとの間で OTP（ワンタイム パスワード）認証方式を使用する際に行われるメッセージ交換を示します。



EAPOL メッセージ交換の待機中に 802.1x 認証がタイムアウトし、MAC 認証バイパスがイネーブルの場合、デバイスはクライアントからイーサネットパケットを検出するとそのクライアントを許可できます。デバイスは、クライアントの MAC アドレスを識別情報として使用し、RADIUS サーバに送信される RADIUS-Access/Request フレームにこの情報を保存します。サーバがデバイスに RADIUS-Access/Accept フレームを送信（許可が成功）すると、ポートが許可されます。許可に失敗してゲスト VLAN が指定されている場合、デバイスはポートをゲスト VLAN に割り当てます。イーサネットパケットの待機中にデバイスが EAPOL パケットを検出すると、デバイスは MAC 認証バイパスプロセスを停止して、802.1x 認証を開始します。

図 *11 : MAC* 認証バイパス中のメッセージ交換

次の図に、MAC 認証バイパス中のメッセージ交換を示します。



802.1x ユーザ ディストリビューション

802.1x ユーザ ディストリビューションを設定すると、複数の異なる VLAN で同じグループ名のユーザのロード バランシングを行うことができます。

VLAN は、RADIUS サーバにより提供されるか、VLAN グループ名でデバイス CLI を介して設定されます。

- RADIUS サーバを設定して、ユーザの複数の VLAN 名を送信します。複数の VLAN 名は、ユーザへの応答の一部として送信できます。802.1x ユーザ ディストリビューションは、特定の VLAN のすべてのユーザを追跡し、許可されたユーザをユーザ数が最も少ない VLAN に移動することでロード バランシングを行います。
- RADIUS サーバを設定してユーザの VLAN グループ名を送信します。VLAN グループ名は、ユーザへの応答の一部として送信できます。デバイス CLI を使用して設定した VLAN グループ名で、選択された VLAN グループ名を検索できます。VLAN グループ名が検出されると、この VLAN グループ名で対応する VLAN を検索して、ユーザ数が最も少ない VLAN が検出されます。ロード バランシングは、対応する許可済みユーザをその VLAN に移動することで行われます。

(注) RADIUS サーバは、VLAN ID、VLAN 名、または VLAN グループを任意に組み合わせて VLAN 情報を送信できます。

802.1x ユーザ ディストリビューションの設定時の注意事項

- 少なくとも1つのVLANがVLANグループにマッピングされることを確認してください。
- 複数の VLAN を VLAN グループにマッピングできます。
- VLAN を追加または削除することで、VLAN グループを変更できます。
- 既存の VLAN を VLAN グループ名からクリアする場合、VLAN の認証済みポートはクリアされませんが、既存の VLAN グループからマッピングが削除されます。
- 最後のVLANをVLANグループ名からクリアすると、VLANグループがクリアされます。
- アクティブ VLAN がグループにマッピングされても VLAN グループをクリアできます。VLAN グループをクリアすると、グループ内で任意の VLAN の認証ステートであるポートまたはユーザはクリアされませんが、VLAN の VLAN グループへのマッピングはクリアされます。

Network Edge Access Topology を使用した 802.1x サプリカントおよびオーセンティケータデバイス

Network Edge Access Topology（NEAT）機能は、ワイヤリング クローゼット（会議室など）外の領域まで識別を拡張します。これにより、任意のタイプのデバイスをポートで認証できます。

- 802.1xスイッチサプリカント：802.1xサプリカント機能を使用することで、別のデバイスのサプリカントとして機能するようにスイッチを設定できます。この設定は、たとえば、デバイスが配線用ボックス外にあり、トランクポートを介してアップストリームデバイスに接続される場合に役に立ちます。802.1xデバイスサプリカント機能を使用して設定されたデバイスは、セキュアな接続のためにアップストリームデバイスで認証します。サプリカントデバイスが認証に成功すると、オーセンティケータデバイスでポートモードがアクセスからトランクに変更されます。サプリカントデバイスでは、CISP をイネーブルにするときに手動でトランクを設定する必要があります。
- アクセス VLAN は、オーセンティケータデバイスで設定されている場合、認証が成功した後にトランクポートのネイティブ VLAN になります。

デフォルトでは、BPDUガードがイネーブルにされたオーセンティケータデバイスにサプリカントデバイスを接続する場合で、オーセンティケータのポートはサプリカントデバイスが認証する前にスパニングツリープロトコル（STP）のブリッジプロトコルデータユニット（BPDU）を受信した場合、err-disabled状態になる可能性があります。認証中にサプリカントのポートから送信されるトラフィックを制御できます。**dot1x supplicant controlled transient** グローバル コンフィギュレーション コマンドを入力すると、認証が完了する前にオーセンティケータのポートがシャットダウンすることがないように、認証中に一時的にサプリカントのポートがブロックされます。認証に失敗すると、サプリカントのポートが開きます。 **no dot1x supplicant controlled transient** グローバル コンフィギュレーション コマンドを入力すると、認証期間中にサプリカントのポートが開きます。これはデフォルトの動作です。

BPDU ガードが **spanning-tree bpduguard enable** インターフェイス コンフィギュレーション コマンドによりオーセンティケータ デバイス ポートでイネーブルになっている場合、サプリカントデバイスで **dot1x supplicant controlled transient** コマンドを使用することを強く推奨します。

（注） **spanning-tree portfast bpduguard default** グローバル コンフィギュレーション コマンドを使用して、グローバルにオーセンティケータデバイスで BPDU ガードをイネーブルにした場合、**dot1x supplicant controlled transient** コマンドを入力すると、BPDU の違反が避けられなくなります。

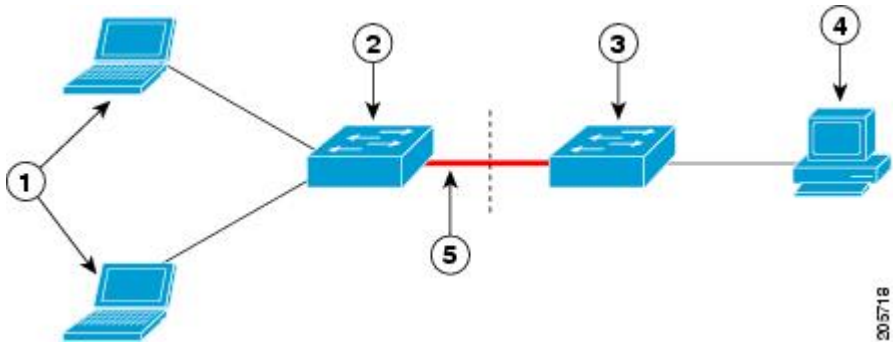
1 つ以上のサプリカントデバイスに接続するオーセンティケータ デバイス インターフェイスで MDA またはマルチ認証モードをイネーブルにできます。マルチホストモードはオーセンティケータ デバイス インターフェイスではサポートされていません。

インターフェイスでシングルホストモードがイネーブルになっている状態でオーセンティケータデバイスをリブートすると、認証の前にインターフェイスが err-disabled 状態に移行することがあります。err-disabled 状態から回復するには、オーセンティケータのポートをフラップしてインターフェイスを再度アクティブにし、認証を開始します。

すべてのホストモードで機能するように **dot1x supplicant force-multicast** グローバル コンフィギュレーション コマンドを Network Edge Access Topology（NEAT）のサプリカントデバイスで使用します。

- ホスト許可：許可済み（サプリカントでデバイスに接続する）ホストからのトラフィックだけがネットワークで許可されます。これらのデバイスは、Client Information Signalling Protocol（CISP）を使用して、サプリカントデバイスに接続する MAC アドレスをオーセンティケータデバイスに送信します。
- 自動イネーブル化：オーセンティケータデバイスでのトランク コンフィギュレーションを自動的にイネーブル化します。これにより、サプリカントデバイスから着信する複数の VLAN のユーザトラフィックが許可されます。ISE で cisco-av-pair を *device-traffic-class=switch* として設定します（この設定は *group* または *user* 設定で行うことができます）。

図 *12 : CISP* を使用したオーセンティケータおよびサプリカントデバイス



1	ワークステーション（クライアント）	2	サプリカントデバイス（配線用ボックス外）
3	オーセンティケータデバイス	4	Cisco ISE
5	トランク ポート		

（注） **switchport nonegotiate** コマンドは、NEAT を使用したサプリカントおよびオーセンティケータデバイスではサポートされません。このコマンドは、トポロジのサプリカント側で設定しないでください。オーセンティケータ サーバ側で設定した場合は、内部マクロによってポートからこのコマンドが自動的に削除されます。

ユーザ単位 ACL および Filter-ID

（注） **any** は、ACL の発信元としてだけ設定できます。

（注） マルチホストモードで設定された ACL では、ステートメントの発信元部分は *any* でなければなりません。（たとえば、 **permit icmp** *any* **host 10.10.1.1**）。

（注） filter-ID としてロールベース ACL を使用することは推奨されません。

定義された ACL の発信元ポートには **any** を指定する必要があります。指定しない場合、ACL は適用できず、認証は失敗します。シングルホストは唯一例外的に後方互換性をサポートします。

MDA 対応ポートおよびマルチ認証ポートでは、複数のホストを認証できます。ホストに適用される ACL ポリシーは、別のホストのトラフィックには影響を与えません。マルチ ホスト ポートで認証されるホストが 1 つだけで、他のホストが認証なしでネットワーク アクセスを取得する場合、発信元アドレスに any を指定することで、最初のホストの ACL ポリシーを他の接続ホストに適用できます。

802.1x/MAB/WebAuth ユーザによるユーザ単位での ACL 認証

ユーザ単位アクセス コントロール リスト（ACL）をイネーブルにして、異なるレベルのネットワーク アクセスおよびサービスを 802.1x 認証ユーザに提供できます。RADIUS サーバは、802.1x ポートに接続されるユーザを認証する場合、ユーザ ID に基づいて ACL 属性を受け取り、これらをデバイスに送信します。デバイスは、ユーザセッションの期間中、その属性を

802.1x ポートに適用します。セッションが終了すると、認証が失敗した場合、またはリンクダウン状態の発生時に、ユーザ単位 ACL 設定が削除されます。デバイスは、RADIUS 指定の ACL を実行コンフィギュレーションには保存しません。ポートが無許可の場合、デバイスはそのポートから ACL を削除します。

RADIUS は、ベンダー固有属性などのユーザ単位属性をサポートします。ベンダー固有属性（VSA）は、オクテットストリング形式で、認証プロセス中にデバイスに渡されます。ユーザ単位 ACL に使用される VSA は、入力方向では inacl#<*n*> で、出力方向では outacl#<*n*> です。MAC ACL は、入力方向に限りサポートされます。VSA は、入力方向に限りサポートされます。レイヤ 2 ポートの出力方向ではポート ACL をサポートしません。

拡張 ACL 構文形式だけを使用して、RADIUS サーバに保存するユーザ単位コンフィギュレーションを定義します。RADIUS サーバから定義が渡される場合、拡張命名規則を使用して作成されます。ただし、Filter-ID 属性を使用する場合、標準 ACL を示すことができます。

Filter-ID 属性を使用して、すでにデバイスに設定されているインバウンドまたはアウトバウンド ACL を指定できます。属性には、ACL 番号と、その後ろに入力フィルタリング、出力フィルタリングを示す *.in* または *.out* が含まれています。RADIUS サーバが *.in* または *.out* 構文を許可しない場合、アクセス リストはデフォルトで発信 ACL に適用されます。RADIUS サーバから送信された Filter-ID がデバイスで設定されていない場合、ユーザは未承認としてマークされます。デバイスでの Cisco IOS のアクセスリストに関するサポートが制限されているため、Filter-ID 属性は 1 ～ 199（IP 標準 ACL）および 1300 ～ 2699（IP 拡張 ACL）の範囲の IP ACL に対してだけサポートされます。

ユーザ単位 ACL の最大サイズは、4000 ASCII 文字ですが、RADIUS サーバのユーザ単位 ACL の最大サイズにより制限されます。

ユーザ単位の ACL を設定するには、次の前提条件を満たす必要があります。

- AAA 認証をイネーブルにします。
- **network** キーワードを使用して AAA 認証をイネーブルにし、RADIUS サーバからのインターフェイス設定を可能にします。
- 802.1x 認証をイネーブルにします。
- RADIUS サーバにユーザ プロファイルと VSA を設定します。

音声認識 802.1x セキュリティ

音声認識 802.1x セキュリティ機能を使用して、セキュリティ違反が発生した場合にデータまたは音声 VLAN に関係なく VLAN だけをディセーブルにするようにデバイスを設定します。この機能が導入される前は、セキュリティ違反の原因であるデータクライアントを認証しようとすると、ポート全体がシャットダウンし、接続が完全に切断されていました。

この機能は、PC が IP Phone に接続されている IP Phone 環境で使用します。データ VLAN でセキュリティ違反が検出されると、データ VLAN だけがシャットダウンされます。音声 VLAN のトラフィックは中断することなくデバイスで送受信されます。

802.1x ポートベース認証の設定方法

802.1x ポートベース認証の設定

802.1x ポートベース認証を設定するには、特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をイネーブルにします。
ステップ 4	**aaa authentication dot1x**{ **default** } *method1* 例： `Device(config)# aaa authentication dot1x default group radius`	802.1x 認証方式リストを作成します。 • **authentication** コマンドにリストが指定されていない場合に使用するデフォルトのリストを作成するには、**default** キーワードの後ろにデフォルト状況で使用される方式を指定します。デフォルトの方式リストは、自動的にすべてのポートに適用されます。 • *method1* には、**group radius** キーワードを入力して、認証用のすべての RADIUS サーバリストを使用できるようにします。

	コマンドまたはアクション	目的
		（注） コマンドラインのヘルプストリングには他のキーワードも表示されますが、サポートされるのは **group radius** キーワードのみです。
ステップ 5	**dot1x system-auth-control** 例： `Device(config)# dot1x system-auth-control`	デバイスで 802.1x 認証をグローバルにイネーブルにします。
ステップ 6	**aaa authorization network** {**default**} **group radius** 例： `Device(config)# aaa authorization network default group radius`	（任意）ユーザ単位 ACL や VLAN 割り当てなど、ネットワーク関連のすべてのサービス要求に対するユーザ RADIUS 許可をデバイスに設定します。
ステップ 7	**radius server** *server-name* 例： `Device(config)# radius server server1`	（任意）RADIUS サーバーの名前を指定し、RADIUS サーバー コンフィギュレーション モードを開始します。
ステップ 8	**address ipv4** *ip address* **auth-port port** *number* **acct-port port** *number* 例： `Device(config-radius-server)# address ipv4 10.1.10.1 auth-port 1645 acct-port 1682`	（任意）RADIUS サーバーを指定します。
ステップ 9	**key** *string* 例： `Device(config-radius-server)# key rad123`	（任意）デバイスと RADIUS サーバで動作する RADIUS デーモンの間で使用される認証と暗号キーを指定します。
ステップ 10	**exit** 例： `Device(config-radius-server)# exit`	RADIUS サーバー コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 11	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/2`	IEEE 802.1x 認証をイネーブルにするクライアントに接続しているポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 12	**switchport mode access** 例： `Device(config-if)# switchport mode access`	（任意）ステップ 6 および 7 で RADIUS サーバを設定した場合のみ、ポートをアクセス モードに設定します。
ステップ 13	**authentication port-control auto** 例： `Device(config-if)# authentication port-control auto`	ポートでの 802.1x 認証を有効にします。
ステップ 14	**dot1x pae authenticator** 例： `Device(config-if)# dot1x pae authenticator`	インターフェイスのポートアクセスエンティティを、オーセンティケータとしてのみ動作し、サプリカント用のメッセージは無視するように設定します。
ステップ 15	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

ポート上での 802.1x 認証のディセーブル化

802.1x 認証をポートでディセーブルにするには、**no dot1x pae** インターフェイス コンフィギュレーション コマンドを使用します。

ポートで 802.1x 認証をディセーブルにするには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	特権 EXEC モードを有効にします。

	コマンドまたはアクション	目的
	`Device> enable`	• パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**switchport mode access** 例： `Device(config-if)# switchport mode access`	（任意）RADIUS サーバを設定した場合に限り、ポートをアクセス モードに設定します。
ステップ 5	**no dot1x pae authenticator** 例： `Device(config-if)# no dot1x pae authenticator`	ポートでの 802.1x 認証をディセーブルにします。
ステップ 6	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

802.1x 認証設定のデフォルト値へのリセット

802.1x 認証設定をデフォルト値に戻すには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

手順

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

	コマンドまたはアクション	目的
	`Device> enable`	
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/1`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**dot1x default** 例： `Device(config-if)# dot1x default`	設定可能な 802.1x のパラメータをデフォルト値へ戻します。
ステップ 5	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

定期的な再認証の設定

802.1x クライアントの定期的な再認証をイネーブルにし、再認証の間隔を指定できます。再認証を行う間隔を指定しない場合、3600 秒おきに再認証が試みられます。

クライアントの定期的な再認証をイネーブルにし、再認証を行う間隔（秒）を設定するには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/1`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**authentication periodic** 例： `Device(config-if)# authentication periodic`	クライアントの定期的な再認証（デフォルトではディセーブル）をイネーブルにします。 （注） デフォルト値は3600秒です。再認証タイマーの値を変更するか、デバイスに RADIUS-provided セッション タイムアウトを使用させるには、**authentication timer reauthenticate** コマンドを入力します。
ステップ 5	**authentication timer** {{[**reauthenticate** | **restart** | **unauthorized**]} {*value*}} 例： `Device(config-if)# authentication timer reauthenticate 180`	再認証の試行の間隔（秒）を設定します。 **authentication timer** キーワードの意味は次のとおりです。 • **reauthenticate**：自動再認証試行が開始されるまでの時間（秒） • **restart** *value*：無許可ポートの認証の試行が行われるまでの間隔（秒） • **unauthorized** *value*：不正セッションが削除されるまでの間隔（秒） このコマンドがデバイスの動作に影響するのは、定期的な再認証をイネーブルに設定した場合だけです。
ステップ 6	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

再認証回数の設定

ポートが無許可ステートに変わる前に、デバイスが認証プロセスを再開する回数を変更することもできます。

（注） このコマンドのデフォルト値は、リンクの信頼性が低下した場合や、特定のクライアントおよび認証サーバの動作に問題がある場合など、異常な状況に対する調整を行う必要がある際に限って変更してください。

再認証回数を設定するには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**switchport mode access** 例： `Device(config-if)# switchport mode access`	RADIUS サーバを事前に設定した場合に限り、ポートをアクセス モードに設定します。
ステップ 5	**dot1x max-req** *count* 例： `Device(config-if)# dot1x max-req 4`	ポートが無許可ステートに変わる前に、デバイスが認証プロセスを再開する回数を設定します。指定できる範囲は 0 ～ 10 です。デフォルトは 2 です。

	コマンドまたはアクション	目的
ステップ 6	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

デバイスからクライアントへのフレーム再送信回数の設定

デバイスからクライアントへの再送信時間を変更できるだけでなく、（クライアントから応答が得られなかった場合に）デバイスが認証プロセスを再起動する前に、クライアントに EAP-Request/Identity フレームを送信する回数を変更できます。

（注） このコマンドのデフォルト値は、リンクの信頼性が低下した場合や、特定のクライアントおよび認証サーバの動作に問題がある場合など、異常な状況に対する調整を行う必要があるときに限って変更してください。

デバイスからクライアントへのフレーム再送信回数を設定するには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**dot1x max-reauth-req** *count* 例： `Device(config-if)# dot1x max-reauth-req`	デバイスが認証プロセスを再開するまでに、EAP-Request/Identity フレームをクライアントに送信する回数を設定しま

	コマンドまたはアクション	目的
	5	す。指定できる範囲は 1 ～ 10 です。デフォルトは 2 です。
ステップ 5	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

スイッチからクライアントへの再送信時間の変更

クライアントは、デバイスの EAP-Request/Identity フレームに対し、EAP-Response/Identity フレームで応答します。デバイスがこの応答を受信できなかった場合、所定の時間（再送信時間）だけ待機し、その後フレームを再送信します。

（注） このコマンドのデフォルト値は、リンクの信頼性が低下した場合や、特定のクライアントおよび認証サーバの動作に問題がある場合など、異常な状況に対する調整を行う必要があるときに限って変更してください。

デバイスがクライアントからの通知を待機する時間を変更するには、次の手順を実行します。この手順は任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-type interface-number* 例： 例： `Device(config)# interface gigabitethernet 1/0/1`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**authentication timer reauthenticate** *seconds* 例： `Device(config-if)# authentication timer reauthenticate 60`	スイッチが EAP-Request/Identity フレームに対するクライアントからの応答を待ち、要求を再送信するまでの秒数を設定します。 • 指定できる範囲は 1 ～ 65535 秒です。デフォルトは 5 秒です。
ステップ 5	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 6	**show authentication sessions interface** *type number* 例： 例： `Device# show authentication sessions gigabitethernet 1/0/1`	指定されたインターフェイスの現在の認証マネージャセッションに関する情報を表示します。

ホスト モードの設定

authentication port-control インターフェイス コンフィギュレーション コマンドが **auto** に設定されている IEEE 802.1x 許可ポート上で、複数のホスト（クライアント）を許可するには、特権 EXEC モードで次の手順を実行します。マルチドメイン認証（MDA）を設定してイネーブルにするには、**multi-domain** キーワードを使用します。これにより、ホストデバイス、および IP Phone（シスコ製または他社製）など音声デバイスの両方が同じスイッチポートで許可されます。この手順は任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**authentication host-mode**[**multi-auth** |**multi-domain** |**multi-host** |**single-host**] 例： `Device(config-if)# authentication host-mode multi-host`	単一の 802.1x 許可ポートで複数のホスト（クライアント）を許可することができます。 キーワードの意味は次のとおりです。 • **multi-auth**：音声 VLAN とデータ VLAN の両方で複数の認証クライアントを許可します。 （注） **multi-auth** キーワードは、**authentication host-mode** コマンドでのみ使用できます。 • **multi-host**：シングルホストの認証後に 802.1x 許可ポートで複数のホストの接続を許可します。 • **multi-domain**：ホストデバイスと IP Phone（シスコ製または他社製）などの音声デバイスの両方が、IEEE 802.1x 許可ポートで認証されるようにします。 （注） ホストモードが **multi-domain** に設定されている場合、IP Phone の音声 VLAN を設定する必要があります。 指定のインターフェイスに対し **authentication port-control** インターフェイス コンフィギュレーション コマンドが **auto** に設定されていることを確認してください。

	コマンドまたはアクション	目的
ステップ 5	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

MAC 移動のイネーブル化

MAC 移動を使用すると、認証されたホストをデバイスのポート間で移動できます。

デバイスで MAC 移動をグローバルに有効にするには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**authentication mac-move permit** 例： `Device(config)# authentication mac-move permit`	デバイスで MAC 移動を有効にします。デフォルトは deny です。 • セッション認識型ネットワークモードでは、デフォルト CLI は **access-session mac-move deny** です。セッション認識型ネットワークで MAC 移動をイネーブルにするには、**no access-session mac-move** 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。 • **mac-move** のデフォルト値は、レガシーモード（IBNS 1.0）の場合は **deny** で、C3PL モード（IBNS 2.0）の場合は **permit**です。

	コマンドまたはアクション	目的
ステップ 4	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

MAC 置換のイネーブル化

MAC 置換を使用すると、ホストはポート上の認証ホストを置換できます。

インターフェイス上で MAC 置換をイネーブルにするには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**authentication violation** {**protect** | **replace** | **restrict** | **shutdown**} 例： `Device(config-if)# authentication violation replace`	インターフェイス上で MAC 置換をイネーブルにするには、**replace** キーワードを使用します。ポートが現在のセッションを削除し、新しいホストを使用して認証を開始します。 他のキーワードは、次のような機能があります。 • **protect**：ポートは、システム メッセージを生成せずに、予期しない MAC を使用するパケットをドロップします。

	コマンドまたはアクション	目的
		• **restrict**：違反パケットが CPU によってドロップされ、システムメッセージが生成されます。 • **shutdown**：ポートは、予期しない MAC アドレスを受信すると error disabled になります。
ステップ 5	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

802.1x アカウンティングの設定

802.1xアカウンティングを使用して、AAAシステムアカウンティングをイネーブルにすると、ロギングのためにシステム リロード イベントをアカウンティング RADIUS サーバに送信できます。サーバは、アクティブな 802.1x セッションすべてが終了したものと判断します。

RADIUS は信頼性の低い UDP トランスポート プロトコルを使用するため、ネットワーク状態が良好でないと、アカウンティング メッセージが失われることがあります。設定可能な回数のアカウンティング要求の再送信後、デバイスがRADIUSサーバからアカウンティング応答メッセージを受信しない場合、次のメッセージが表示されます。

```
Accounting message %s for session %s failed to receive Accounting Response.
```

このストップ メッセージが正常に送信されない場合、次のメッセージが表示されます。

```
00:09:55: %RADIUS-4-RADIUS_DEAD: RADIUS server 172.20.246.201:1645,1646 is not responding.
```

（注） ロギングの開始、停止、仮のアップデート メッセージ、タイム スタンプなどのアカウンティング タスクを実行するように、RADIUS サーバを設定する必要があります。

手順

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

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/3`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**aaa accounting dot1x default start-stop group radius** 例： `Device(config-if)# aaa accounting dot1x default start-stop group radius`	すべての RADIUS サーバのリストを使用して 802.1x アカウンティングをイネーブルにします。
ステップ 5	**aaa accounting system default start-stop group radius** 例： `Device(config-if)# aaa accounting system default start-stop group radius`	（任意）システムアカウンティングをイネーブルにし（すべての RADIUS サーバのリストを使用）、デバイスがリロードするときにシステム アカウンティング リロード イベント メッセージを生成します。
ステップ 6	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

デバイスと RADIUS サーバの通信の設定

認証、許可、およびアカウンティング（AAA）をイネーブルにし、認証方式リストを指定する必要があります。方式リストは、ユーザ認証のためにクエリ送信を行う手順と認証方式を記述したものです。

すべての RADIUS サーバについて、タイムアウト、再送信回数、および暗号キー値をグローバルに設定するには、**radius server** 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。これらのオプションをサーバ単位で設定するには、**radius-server timeout**、**radius-server retransmit**、および **radius-server key** 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。

RADIUS サーバ上でも、いくつかの値を設定する必要があります。deviceの IP アドレス、およびサーバとdeviceの双方で共有するキーストリングなどの設定値です。詳細については、RADIUS サーバのマニュアルを参照してください。

デバイスでRADIUS サーバのパラメータを設定するには、次の手順を実行します。この手順は必須です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**radius server** *server-name* 例： `Device(config)# radius server server1`	（任意）RADIUS サーバーの名前を指定し、RADIUS サーバー コンフィギュレーション モードを開始します。
ステップ 4	**address ipv4** *ip address* **auth-port port** *number* **acct-port port** *number* 例： `Device(config-radius-server)# address ipv4 10.1.10.1 auth-port 1645 acct-port 1682`	（任意）RADIUS サーバーを指定します。
ステップ 5	**key** *string* 例： `Device(config-radius-server)# key rad123`	（任意）デバイスと RADIUS サーバで動作する RADIUS デーモンの間で使用される認証と暗号キーを指定します。
ステップ 6	**end** 例： `Device(config-radius-server)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

802.1X 認証の設定

ユーザ単位ACLまたはVLAN割り当てを可能にするには、AAA許可をイネーブルにしてネットワーク関連のすべてのサービス要求に対してデバイスを設定する必要があります。

次に、802.1x の AAA プロセスを示します。

始める前に

802.1xポートベース認証を設定するには、認証、許可、アカウンティング（AAA）をイネーブルにして認証方式リストを指定する必要があります。方式リストは、ユーザ認証のためにクエリー送信を行う手順と認証方式を記述したものです。

手順

	コマンドまたはアクション	目的
ステップ 1	ユーザがデバイスのポートに接続します。	
ステップ 2	認証が実行されます。	
ステップ 3	RADIUSサーバ設定に基づいて、VLAN割り当てが適宜イネーブルになります。	
ステップ 4	デバイスが開始メッセージをアカウンティングサーバに送信します。	
ステップ 5	必要に応じて、再認証が実行されます。	
ステップ 6	デバイスが仮のアカウンティング アップデートを、再認証結果に基づいたアカウンティングサーバに送信します。	
ステップ 7	ユーザがポートから切断します。	
ステップ 8	デバイスが停止メッセージをアカウンティングサーバに送信します。	

認証のリトライ回数の設定

ユーザに制限付きVLANを割り当てる前に、 **authentication event retry** *retry count* インターフェイス コンフィギュレーション コマンドを使用して、認証試行回数を最大に設定できます。指定できる試行回数は 1 ～ 3 です。デフォルトは 3 回に設定されています。

許可される認証の最大試行回数を設定するには、このオプションタスクを実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/1`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**authentication port-control auto** 例： `Device(config-if)# authentication port-control auto`	ポートでの 802.1X 認証をイネーブルにします。
ステップ 5	**authentication event fail action authorize vlan** *vlan-id* 例： `Device(config-if)# authentication event fail action authorize vlan 40`	アクティブ VLAN を 802.1X 認証失敗 VLAN として指定します。指定できる範囲は 1 ～ 4094 です。
ステップ 6	**authentication event failretry** *retry-count* 例： `Device(config-if)# authentication event fail retry 4`	ポートが認証失敗 VLAN に移行するまでの認証試行回数を指定します。範囲は 0 ～ 5 で、デフォルトでは、最初の失敗イベント後の 2 回の試行です。
ステップ 7	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

例

次の例では、ポートが認証失敗 VLAN に移行するまでに許容される認証試行回数を 2 に設定する方法を示します。

```
Device(config-if)# authentication event retry 2
```

柔軟な認証順序の設定

下の手順で使用される例は、MAB が IEEE 802.1x 認証（dot1x）の前に試行されるように柔軟な認証の順序設定の順序を変更します。MAB は最初の認証方式として設定されているため、MAB は他のすべての認証方式よりも優先されます。

（注） これらの認証方式のデフォルトの順序とプライオリティを変更する前に、これらの変更による潜在的な結果を理解する必要があります。詳細について、http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6638/application_note_c27-573287_ps6638_Products_White_Paper.html を参照してください。

特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/1`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**switchport mode access** 例： `Device(config-if)# switchport mode access`	RADIUS サーバを事前に設定した場合に限り、ポートをアクセス モードに設定します。
ステップ 5	**authentication order** [**dot1x** | **mab**] | {**webauth**} 例：	（任意）ポート上で使用される認証方式の順序を設定します。

	コマンドまたはアクション	目的
	`Device(config-if)# authentication order mab dot1x`	
ステップ 6	**authentication priority** [**dot1x** | **mab**] | {**webauth**} 例： `Device(config-if)# authentication priority mab dot1x`	（任意）認証方式をポート プライオリティ リストに追加します。
ステップ 7	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

ゲスト VLAN の設定

サーバが EAP Request/Identity フレームに対する応答を受信しない場合、ゲスト VLAN を設定すると、802.1x 対応でないクライアントはゲスト VLAN に配置されます。802.1x 対応であっても、認証に失敗したクライアントは、ネットワークへのアクセスが許可されません。デバイスは、シングルホストモードまたはマルチホストモードでゲスト VLAN をサポートします。

ゲスト VLAN を設定するには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/1`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**authentication event no-response action authorize vlan** *vlan-id* 例： `Device(config-if)# authentication event no-response action authorize vlan 2`	アクティブ VLAN を 802.1x ゲスト VLAN として指定します。指定できる範囲は 1 ～ 4094 です。 内部 VLAN（ルーテッド ポート）、RSPAN VLAN、音声 VLAN を除くあらゆるアクティブ VLAN を 802.1x ゲスト VLAN として設定できます。
ステップ 5	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

制限付き VLAN の設定

デバイスに制限付き VLAN を設定すると、認証サーバが有効なユーザ名とパスワードを受信しなかった場合、IEEE 802.1x 準拠のクライアントが制限付き VLAN に移動します。デバイスは、シングルホストモードでのみ制限付き VLAN をサポートします。

制限付き VLAN を設定するには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/1`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**authentication port-control auto** 例：	ポートでの 802.1x 認証をイネーブルにします。

	コマンドまたはアクション	目的
	`Device(config-if)# authentication port-control auto`	
ステップ 5	**authentication event fail action authorize vlan** *vlan-id* 例： `Device(config-if)# authentication event fail action authorize vlan 2`	アクティブ VLAN を 802.1x 制限付き VLAN として指定します。指定できる範囲は 1 ～ 4094 です。 • 内部VLAN（ルーテッドポート）、RSPAN VLAN または音声 VLAN を除き、任意のアクティブ VLAN を 802.1x 制限 VLAN として設定できます。
ステップ 6	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

802.1X 認証失敗 VLAN の設定

認証失敗 VLAN を設定するには、次の作業を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type slot/port* 例： `Device(config)# interface gigabitethernet 1/0/1`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**access-session port-control auto** 例： `Device(config-if)# access-session port-control auto`	ポートでの 802.1X 認証をイネーブルにします。
ステップ 5	**authentication event fail action authorize vlan** *vlan-id* 例： `Device(config-if)# authentication event fail action authorize vlan 40`	アクティブ VLAN を 802.1X 認証失敗 VLAN として指定します。指定できる範囲は 1 ～ 4094 です。
ステップ 6	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show access-session interface** *interface-id* 例： `Device# show access-session interface gigabitethernet 1/0/1`	（任意）設定を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

認証失敗 VLAN をディセーブルにして削除するには、**no authentication event fail** インターフェイス コンフィギュレーション コマンドを使用します。ポートはデフォルトステートに戻ります。

Open1x の設定

ポートの許可ステートの手動制御をイネーブルにするには、特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/1`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**switchport mode access** 例： `Device(config-if)# switchport mode access`	RADIUS サーバを設定した場合に限り、ポートをアクセス モードに設定します。
ステップ 5	**authentication control-direction** {**both** | **in**} 例： `Device(config-if)# authentication control-direction both`	（任意）ポート制御を単一方向モードまたは双方向モードに設定します。
ステップ 6	**authentication fallback** *name* 例： `Device(config-if)# authentication fallback profile1`	（任意）802.1x 認証をサポートしないクライアント用のフォールバック方法として Web 認証を使用するようポートを設定します。
ステップ 7	**authentication host-mode**[**multi-auth** |**multi-domain** |**multi-host** |**single-host**] 例： `Device(config-if)# authentication host-mode multi-auth`	（任意）ポート上で認証マネージャ モードを設定します。

	コマンドまたはアクション	目的
ステップ 8	**authentication open** 例： `Device(config-if)# authentication open`	（任意）ポート上でオープンアクセスをイネーブルまたはディセーブルにします。
ステップ 9	**authentication order** [**dot1x** | **mab**] | {**webauth**} 例： `Device(config-if)# authentication order dot1x webauth`	（任意）ポート上で使用される認証方式の順序を設定します。
ステップ 10	**authentication periodic** 例： `Device(config-if)# authentication periodic`	（任意）ポート上で再認証をイネーブルまたはディセーブルにします。
ステップ 11	**authentication port-control** {**auto** | **force-authorized** | **force-un authorized**} 例： `Device(config-if)# authentication port-control auto`	（任意）ポートの許可ステートの手動制御をイネーブルにします。
ステップ 12	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

ユーザのログイン制限の設定

手順

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

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa new-model** 例： `Device(config)# aaa new-model`	認証、許可、およびアカウンティング（AAA）アクセス コントロール モデルをイネーブルにします。
ステップ 4	**aaa authentication login default local** 例： `Device(config)# aaa authentication login default local`	デフォルトの認証方法を使用して、認証、許可、およびアカウンティング（AAA）認証を設定します。
ステップ 5	**aaa authentication rejected** *n* **in** *m* **ban** *x* 例： `Device(config)# aaa authentication rejected 3 in 20 ban 300`	ユーザによるログインが指定の時間および試行回数以内に成功しなかった場合にユーザをブロックする時間を設定します。 • *n*：ユーザがログインを試行できる回数を指定します。 • *m*：ユーザがログインを試行できる時間を秒数で指定します。 • *x*：ログインに成功しなかったユーザのアクセスを禁止する期間を指定します。
ステップ 6	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 7	**show aaa local user blocked** 例： `Device# show aaa local user blocked`	ブロックされたローカル ユーザのリストを表示します。
ステップ 8	**clear aaa local user blocked username** *username* 例： `Device# clear aaa local user blocked username user1`	ブロックされたローカル ユーザに関する情報を消去します。

例

次に、**show aaa local user blocked** コマンドの出力例を示します。

```
Device# show aaa local user blocked

       Local-user                  State

       user1                       Watched (till 11:34:42 IST Feb 5 2015)
```

クリティカル音声 VLAN を使用した 802.1x アクセス不能認証バイパスの設定

ポートにクリティカル音声 VLAN を設定し、アクセス不能認証バイパス機能をイネーブルにするには、特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をイネーブルにします。
ステップ 4	**radius-server dead-criteria** {**time** *seconds*} [**tries** *number*] 例： `Device(config)# radius-server dead-criteria time 20 tries 10`	RADIUS サーバが使用不可またはダウン（切断）と見なされる条件を設定します。 • **time**：1 ～ 120 秒。デバイスは、デフォルトの *seconds* 値を 10 ～ 60 の間で動的に決定します。 • **number**：1 ～ 100 の試行回数。デバイスは、デフォルトの **tries**

	コマンドまたはアクション	目的
		number を 10 ～ 100 の間で動的に決定します。
ステップ 5	**radius-server deadtime** 分 例： `Device(config)# radius-server deadtime 60`	（任意）RADIUS サーバに要求が送信されない分数を設定します。 • 指定できる範囲は 0 ～ 1440 分（24 時間）です。デフォルト値は 0 分です。
ステップ 6	**radius server** *server-name* 例： `Device(config)# radius server server1`	（任意）RADIUS サーバーの名前を指定し、RADIUS サーバー コンフィギュレーション モードを開始します。
ステップ 7	**address ipv4** *ip address* **auth-port port** *number* **acct-port port** *number* 例： `Device(config-radius-server)# address ipv4 10.1.10.1 auth-port 1645 acct-port 1682`	（任意）RADIUS サーバーを指定します。
ステップ 8	**key** *string* 例： `Device(config-radius-server)# key rad123`	（任意）デバイスと RADIUS サーバで動作する RADIUS デーモンの間で使用される認証と暗号キーを指定します。
ステップ 9	**dot1x critical eapol** 例： `Device(config)# dot1x critical eapol`	（任意）アクセス不能認証バイパスのパラメータを設定します。 **eapol**：デバイスがクリティカルポートを正常に認証すると、デバイスが EAPOL 成功メッセージを送信するように指定します。
ステップ 10	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/1`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 11	**authentication event server dead action** {**authorize** | **reinitialize**} **vlan** *vlan-id*] 例： `Device(config-if)# authentication event server dead action reinitialicze vlan 20`	これらのキーワードを使用して、RADIUS サーバが到達不能な場合にポートでホストを移動します。 • **authorize**：認証しようとする新しいホストをユーザ指定のクリティカル VLAN に移動します。 • **reinitialize**：ポートのすべての許可済みホストをユーザ指定のクリティカル VLAN に移動します。
ステップ 12	**switchport voice vlan** *vlan-id* 例： `Device(config-if)# switchport voice vlan`	ポートの音声 VLAN を指定します。音声 VLAN はステップ 6 で設定されたクリティカルデータ VLAN と同じにはできません。
ステップ 13	**authentication event server dead action authorize voice** 例： `Device(config-if)# authentication event server dead action authorize voice`	RADIUS サーバが到達不能な場合、ポートのデータ トラフィックを音声 VLAN に移動するために、クリティカル音声 VLAN を設定します。
ステップ 14	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 15	**show authentication interface** *type number* 例： `Device# show authentication interface gigabitethernet 1/0/1`	（任意）設定を確認します。

次のタスク

RADIUS サーバをデフォルトの設定に戻すには、**no radius-server dead-criteria**、**no radius-server deadtime**、および **no radius server** のグローバル コンフィギュレーション コマンドを使用します。アクセス不能認証バイパスをディセーブルにするには、**no authentication event server dead action** インターフェイス コンフィギュレーション コマンドを使用します。クリティカル音声

VLAN をディセーブルにするには、 **no authentication event server dead action authorize voice** インターフェイス コンフィギュレーション コマンドを使用します。

MAC 認証バイパスの設定

MAC 認証バイパスをイネーブルにするには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**authentication port-control auto** 例： `Device(config-if)# authentication port-control auto`	ポートでの 802.1x 認証をイネーブルにします。
ステップ 5	**mab** [**eap**] 例： `Device(config-if)# mab`	MAC 認証バイパスをイネーブルにします。 （任意）**eap** キーワードを使用して、許可用に EAP を使用できるようにデバイスを設定します。
ステップ 6	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

MAC 認証バイパスのユーザ名とパスワードの形式作成

オプションの **mab request format** コマンドを使用して認証サーバによって受け入れられる形式で MAB のユーザ名とパスワードをフ形式作成します。ユーザ名とパスワードは通常、クライアントの MAC アドレスです。認証サーバ設定の中には、ユーザ名と異なるパスワードを必要とするものがあります。

MAC 認証バイパス ユーザ名およびパスワードを形式作成するには、特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**mab request format attribute 1 groupsize**{**1** | **2** | **4** |**12**}[**separator** {**-** | **:** | **.**}{**lowercase** | **uppercase**}] 例： `Device(config)# mab request format attribute 1 groupsize 12`	MAB で生成された Access-Request パケットの User-Name 属性内の MAC アドレスの形式を指定します。 • **1**：MAC アドレスの 12 桁の 16 進数のユーザ名形式を設定します。 • **groupsize**：区切り文字の挿入の前に連結する 16 進ニブルの数。有効なグループ サイズは、1、2、4、12 のいずれかである必要があります。 • **separator**：グループサイズに従って 16 進ニブルを区切る文字。有効な区切り文字は、ハイフン、コロン、ピリオドのいずれかである必要があります。12 のグループ サイズでは、区切り文字は使用されません。 • {**lowercase** | **uppercase**}：数字以外の 16 進ニブルを小文字または大文字のどちらにするかを指定します。

	コマンドまたはアクション	目的
ステップ 4	**mab request format attribute2** {**0** | **7**} *text* 例： `Device(config)# mab request format attribute 2 7 A02f44E18B12`	• **2**：MAB で生成された Access-Request パケット内の User-Password 属性のカスタム（デフォルト以外の）値を指定します。 • **0**：追跡するクリアテキストパスワードを指定します。 • **7**：追跡する暗号化パスワードを指定します。 • *text*：User-Password 属性で使用するパスワードを指定します。 （注） 設定情報を電子メールで送信する場合、タイプ 7 のパスワード情報を削除してください。**show tech-support** コマンドは、デフォルトで出力からこの情報を削除します。
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

制限付き VLAN の認証試行回数の設定

ユーザーに制限付き VLAN を割り当てる前に、**authentication event fail retry** *retry count* インターフェイス コンフィギュレーション コマンドを使用して、認証試行回数を最大に設定できます。指定できる試行回数は 1 ～ 3 です。デフォルトは 3 回に設定されています。

認証試行回数を最大に設定するには、特権EXECモードで次の手順を実行します。この手順は任意です。

手順

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

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**authentication port-control auto** 例： `Device(config-if)# authentication port-control auto`	ポートでの 802.1x 認証をイネーブルにします。
ステップ 5	**authentication event fail action authorize vlan** *vlan-id* 例： `Device(config-if)# authentication event fail action authorize vlan 8`	アクティブ VLAN を 802.1x 制限付き VLAN として指定します。指定できる範囲は 1 ～ 4094 です。 • 内部VLAN（ルーテッドポート）、RSPAN VLAN または音声 VLAN を除き、任意のアクティブ VLAN を 802.1x 制限 VLAN として設定できます。
ステップ 6	**authentication event fail retry** *retry count* 例： `Device(config-if)# authentication event fail retry 2`	ポートが制限付き VLAN に移行するための認証試行回数を指定します。指定できる範囲は1～3秒です。デフォルトは 3 回に設定されています。
ステップ 7	**end** 例： `Device(config-if)# end`	インターフェイスコンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

VLAN ID ベース MAC 認証の設定

特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**mab request format attribute 32 vlan access-vlan** 例： `Device(config)# mab request format attribute 32 vlan access-vlan`	VLAN ID ベース MAC 認証をイネーブルにします。
ステップ 4	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

NEAT を使用したサプリカントデバイスの設定

デバイス VSA ではなく Auto Smartport ユーザ定義マクロを使用して、オーセンティケータデバイスを設定することもできます。

デバイスをサプリカントに設定するには、特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**cisp enable** 例： `Device(config)# cisp enable`	CISP をイネーブルにします。
ステップ 4	**dot1x credentials** *profile* 例： `Device(config)# dot1x credentials test`	802.1x クレデンシャル プロファイルを作成します。これは、サプリカントとして設定されるポートに接続する必要があります。
ステップ 5	**username** *suppswitch* 例： `Device(config)# username suppswitch`	ユーザ名を作成します。
ステップ 6	**password** *password* 例： `Device(config)# password myswitch`	新しいユーザ名のパスワードを作成します。
ステップ 7	**dot1x supplicant force-multicast** 例： `Device(config)# dot1x supplicant force-multicast`	ユニキャストまたはマルチキャストパケットのいずれかを受信した場合にデバイスに強制的にマルチキャスト EAPOL だけを送信させます。 これにより、NEAT がすべてのホストモードでのサプリカントデバイスで機能できるようにもなります。
ステップ 8	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 9	**switchport mode trunk** 例： `Device(config-if)# switchport mode trunk`	インターフェイスを VLAN トランク ポートとして設定します。

	コマンドまたはアクション	目的
ステップ 10	**dot1x pae supplicant** 例： `Device(config-if)# dot1x pae supplicant`	インターフェイスをポートアクセスエンティティ（PAE）サプリカントとして設定します。
ステップ 11	**dot1x credentials** *profile-name* 例： `Device(config-if)# dot1x credentials test`	802.1x クレデンシャル プロファイルをインターフェイスに対応付けます。
ステップ 12	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

NEAT を使用したオーセンティケータデバイスの設定

この機能を設定するには、配線用ボックス外の 1 つのデバイスがサプリカントとして設定され、オーセンティケータデバイスに接続されている必要があります。

（注）
- CISP または NEAT セッションがアクティブなときにラインカードを取り外してシャーシに挿入する場合は、オーセンティケータ デバイス インターフェイスの設定を明示的にフラッピングすることによって、アクセスモードに復元する必要があります。
- *cisco-av-pairs* は、Cisco ISE で *device-traffic-class=switch* として設定されている必要があります。これにより、サプリカントが正常に認証された後でトランクとしてインターフェイスが設定されます。

デバイスをオーセンティケータに設定するには、特権EXECモードで次の手順を実行します。

手順

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

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**cisp enable** 例： `Device(config)# cisp enable`	CISP をイネーブルにします。
ステップ 4	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 5	**switchport mode access** 例： `Device(config-if)# switchport mode access`	ポートモードを **access** に設定します。
ステップ 6	**authentication port-control auto** 例： `Device(config-if)# authentication port-control auto`	ポート認証モードを auto に設定します。
ステップ 7	**dot1x pae authenticator** 例： `Device(config-if)# dot1x pae authenticator`	インターフェイスをポート アクセス エンティティ（PAE）オーセンティケータとして設定します。
ステップ 8	**spanning-tree portfast** 例： `Device(config-if)# spanning-tree portfast trunk`	単一ワーク ステーションまたはサーバに接続されたアクセス ポート上で Port Fast をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 9	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 10	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。 （注） 変更をコンフィギュレーション ファイルに保存すると、オーセンティケータインターフェイスがリロード後も引き続きトランクモードになることを意味します。 オーセンティケータインターフェイスをアクセスポートとして維持する場合は、コンフィギュレーションファイルに変更を保存しないでください。

待機時間の変更

デバイスがクライアントを認証できない場合、デバイスは所定の時間アイドル状態になり、その後再試行します。**authentication timer restart** インターフェイス コンフィギュレーション コマンドは、アイドル状態の期間を制御します。認証が失敗する理由としては、クライアントが無効なパスワードを提示した場合などが考えられます。デフォルトよりも小さい値を入力することによって、ユーザへの応答時間を短縮できます。

待機時間を変更するには、特権EXECモードで次の手順を実行します。この手順は任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**interface** *interface-type interface-number* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**authentication timer restart** *seconds* 例： `Device(config-if)# authentication timer restart 30`	クライアントとの認証のやり取りに失敗した場合に、スイッチが待機状態のままでいる秒数を設定します。 • 指定できる範囲は 1 ～ 65535 秒です。デフォルトは 60 秒です。
ステップ 5	**end** 例： `Device(config-if)# end`	インターフェイスコンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 6	**show authentication sessions interface** *interface-type interface-number* 例： 例： `Device# show authentication sessions interface gigabitethernet 1/0/2`	現在の認証マネージャセッションに関する情報を表示します。

802.1x 違反モードの設定

次に示す状況で、シャットダウン、Syslog エラーを生成、または新しいデバイスからのパケットを廃棄するように 802.1x ポートを設定できます。

- デバイスが 802.1x 対応のポートに接続した
- ポートで認証されるデバイスの最大数に達した

デバイス上にセキュリティ違反アクションを設定するには、特権 EXEC モードで次の手順を実行します。

手順

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

	コマンドまたはアクション	目的
	`Device> enable`	
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をイネーブルにします。
ステップ 4	**aaa authentication dot1x**{ **default** } *method1* 例： `Device(config)# aaa authentication dot1x default group radius`	802.1x 認証方式リストを作成します。 • **authentication** コマンドにリストが指定されていない場合に使用するデフォルトのリストを作成するには、**default** キーワードの後ろにデフォルト状況で使用される方式を指定します。デフォルトの方式リストは、自動的にすべてのポートに適用されます。 • *method1* には、**group radius** キーワードを入力して、認証用のすべての RADIUS サーバリストを使用できるようにします。
ステップ 5	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/2`	IEEE 802.1x 認証をイネーブルにするクライアントに接続しているポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 6	**switchport mode access** 例： `Device(config-if)# switchport mode access`	ポートをアクセスモードに設定します。
ステップ 7	**authentication violation** {**shutdown** | **restrict** | **protect** | **replace**} 例： `Device(config-if)# authentication`	違反モードを設定します。キーワードの意味は次のとおりです。 • **shutdown**：エラーによってポートがディセーブルになります。

	コマンドまたはアクション	目的
	violation restrict	• **restrict**：Syslog エラーを生成します。 • **protect**：トラフィックをポートに送信するすべての新しいデバイスからパケットをドロップします。 • **replace**：現在のセッションを削除し、新しいホストで認証します。
ステップ 8	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

音声認識 802.1x セキュリティの設定

音声認識 802.1x セキュリティ機能をデバイスで使用して、セキュリティ違反が発生した場合にデータまたは音声 VLAN に関係なく VLAN だけをディセーブルにします。この機能は、PC が IP Phone に接続されている IP Phone 環境で使用できます。データ VLAN でセキュリティ違反が検出されると、データ VLAN だけがシャットダウンされます。音声 VLAN のトラフィックは中断することなくデバイスで送受信されます。

デバイスで音声認識 802.1x 音声セキュリティを設定する場合、次の注意事項に従ってください。

- 音声認識 802.1x セキュリティをイネーブルにするには、**errdisable detect cause security-violation shutdown vlan** グローバル コンフィギュレーション コマンドを入力します。音声認識 802.1x セキュリティをディセーブルにするには、このコマンドの **no** バージョンを入力します。このコマンドは、デバイスの 802.1x 設定ポートのすべてに適用されます。

  注 **shutdown vlan** キーワードを指定しない場合、error-disabled ステートになった際にポート全体がシャットダウンされます。

- **errdisable recovery cause security-violation** グローバル コンフィギュレーション コマンドを使用して、error-disabled リカバリを設定すると、ポートは自動的に再びイネーブルにされます。error-disabled リカバリがポートで設定されていない場合、**shutdown** および **no shutdown** インターフェイス コンフィギュレーション コマンドを使用してポートを再びイネーブルにします。

- 個々の VLAN を再びイネーブルにするには、**clear errdisable interface** *interface-id* **vlan** [*vlan-list*] 特権 EXEC コマンドを使用します。範囲を指定しない場合、ポートのすべての VLAN がイネーブルにされます。

音声認識 802.1x セキュリティをイネーブルにするには、特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**errdisable detect cause security-violation shutdown vlan** 例： `Device(config)# errdisable detect cause security-violation shutdown vlan`	セキュリティ違反エラーが発生したすべての VLAN をシャットダウンします。 （注） **shutdown vlan** キーワードを指定しない場合、すべてのポートが error-disabled ステートになり、シャットダウンされます。
ステップ 4	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 5	**clear errdisable interface***interface-type interface-number* **vlan** *[vlan-list]* 例： `Device(config)# clear errdisable interface gigabitethernet 1/0/2 vlan`	（任意）errdisable になっている個々の VLAN を再びイネーブルにします。 • *interface-type interface-number* 引数の場合、個々の VLAN を再びイネーブルにするポートを指定します。 • （任意）*[vlan-list]* 引数の場合、再びイネーブルにする VLAN のリストを指定します。VLANのリストを指定しない場合は、すべてのVLANが再びイネーブルになります。

	コマンドまたはアクション	目的
ステップ 6	**show errdisable detect** 例： `Device# show errdisable detect`	errdisable 検出ステータスを表示します。

IEEE 802.1x ポートベースの認証の設定例

例：アクセス不能認証バイパスの設定

次に、アクセス不能認証バイパス機能を設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# radius-server dead-criteria time 30 tries 20
Device(config)# radius-server deadtime 60
Device(config)# radius server server1
Device(config-radius-server)# address ipv4 10.1.10.1 auth-port 1645 acct-port 1682
Device(config-radius-server)# key rad123
Device(config-radius-server)# exit
Device(config)# dot1x critical eapol
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# authentication event server dead action reinitialicze vlan 20
Device(config-if)# switchport voice vlan
Device(config-if)# authentication event server dead action authorize voice
Device(config-if)# end
```

例：802.1x/MAB/WebAuth ユーザによるユーザ単位での ACL 認証

次に、ダウンロード可能なポリシーのデバイスを設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authorization network default local group radius
Device(config)# ip device tracking
Device(config)# ip access-list extended default_acl
Device(config-ext-nacl)# permit ip any any
Device(config-ext-nacl)# exit
Device(config)# radius-server vsa send authentication
Device(config)# interface fastEthernet 2/13
Device(config-if)# ip access-group default_acl in
Device(config-if)# end
```

その他の参考資料

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*

標準および RFC

標準/RFC	タイトル
RFC 3580	『*IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines*』

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

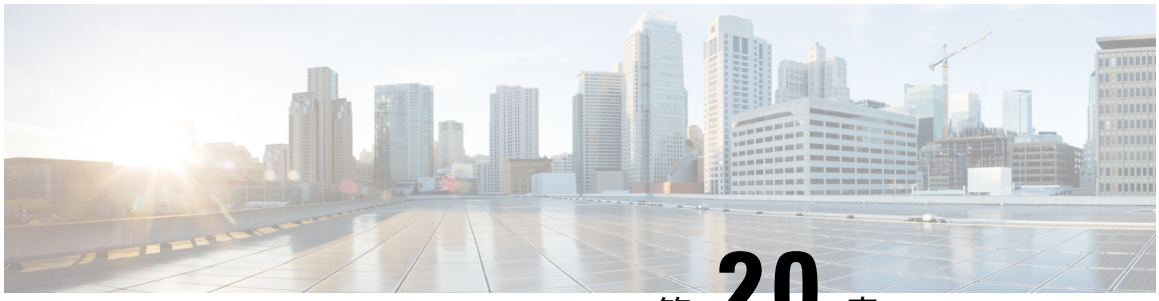
IEEE 802.1x ポートベースの認証の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	IEEE 802.1x ポートベースの認証	IEEE 802.1x 認証は、不正なデバイス（クライアント）によるネットワーク アクセスを防止します。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 20 章

ポート セキュリティの設定

ポート セキュリティの前提条件

最大値をインターフェイス上ですでに設定されているセキュアアドレスの数より小さい値に設定しようとすると、コマンドが拒否されます。

ポート セキュリティの制約事項

スイッチに設定できるセキュア MAC アドレスの最大数は、システムで許可されている MAC アドレスの最大数によって決まります。この値は、使用可能な MAC アドレス（その他のレイヤ 2 機能やインターフェイスに設定されたその他のセキュア MAC アドレスで使用される MAC アドレスを含む）の総数を表します。

ポート セキュリティの概要

ポート セキュリティ

ポート セキュリティ機能を使用すると、ポートへのアクセスを許可するステーションの MAC アドレスを制限および識別して、インターフェイスへの入力を制限できます。セキュア ポートにセキュア MAC アドレスを割り当てると、ポートは定義されたアドレス グループ以外の送信元アドレスを持つパケットを転送しません。セキュア MAC アドレス数を 1 つに制限し、単一

のセキュア MAC アドレスを割り当てると、そのポートに接続されたワークステーションに、ポートの帯域幅全体が保証されます。

セキュア ポートとしてポートを設定し、セキュア MAC アドレスが最大数に達した場合、ポートにアクセスを試みるステーションの MAC アドレスが識別されたセキュア MAC アドレスのいずれとも一致しないので、セキュリティ違反が発生します。また、あるセキュア ポート上でセキュア MAC アドレスが設定または学習されているステーションが、別のセキュア ポートにアクセスしようとしたときにも、違反のフラグが立てられます。

セキュア MAC アドレスのタイプ

デバイスは、次のセキュア MAC アドレスのタイプをサポートします。

- スタティックセキュア MAC アドレス：**switchport port-security mac-address** *mac-address* インターフェイス コンフィギュレーション コマンドを使用して手動で設定され、アドレステーブルに保存された後、スイッチの実行コンフィギュレーションに追加されます。
- ダイナミックセキュア MAC アドレス：動的に設定されてアドレステーブルにのみ保存され、スイッチの再起動時に削除されます。
- スティッキーセキュア MAC アドレス：動的に学習することも、手動で設定することもできます。アドレステーブルに保存され、実行コンフィギュレーションに追加されます。このアドレスがコンフィギュレーションファイルに保存されていると、スイッチの再起動時にインターフェイスはこれらを動的に再設定する必要がありません。

スティッキ セキュア MAC アドレス

スティッキー ラーニングをイネーブルにすると、ダイナミック MAC アドレスをスティッキー セキュア MAC アドレスに変換して実行コンフィギュレーションに追加するようにインターフェイスを設定できます。インターフェイスはスティッキ ラーニングがイネーブルになる前に学習したものを含め、すべてのダイナミック セキュア MAC アドレスをスティッキー セキュア MAC アドレスに変換します。すべてのスティッキー セキュア MAC アドレスは実行コンフィギュレーションに追加されます。

スティッキー セキュア MAC アドレスは、コンフィギュレーション ファイル（スイッチが再起動されるたびに使用されるスタートアップ コンフィギュレーション）に、自動的には反映されません。スティッキー セキュア MAC アドレスをコンフィギュレーション ファイルに保存すると、スイッチの再起動時にインターフェイスはこれらを再び学習する必要がありません。スティッキ セキュア アドレスを保存しない場合、アドレスは失われます。

スティッキ ラーニングがディセーブルの場合、スティッキ セキュア MAC アドレスはダイナミック セキュア アドレスに変換され、実行コンフィギュレーションから削除されます。

セキュリティ違反

次のいずれかの状況が発生すると、セキュリティ違反になります。

- 最大数のセキュア MAC アドレスがアドレス テーブルに追加されている状態で、アドレス テーブルに未登録の MAC アドレスを持つステーションがインターフェイスにアクセスしようとした場合。
- あるセキュア インターフェイスで学習または設定されたアドレスが、同一 VLAN 内の別のセキュア インターフェイスで使用された場合。
- ポートセキュリティが有効な状態で診断テストを実行しています。

違反が発生した場合の対処に基づいて、次の 3 種類の違反モードのいずれかにインターフェイスを設定できます。

- Protect（保護）：セキュア MAC アドレスの数がポートで許可されている最大限度に達すると、最大値を下回るまで十分な数のセキュア MAC アドレスを削除するか、許可アドレス数を増やさないかぎり、未知の送信元アドレスを持つパケットはドロップされます。セキュリティ違反が起こっても、ユーザには通知されません。

  (注) トランク ポートに protect 違反モードを設定することは推奨しません。保護モードでは、ポートが最大数に達していなくても VLAN が保護モードの最大数に達すると、ラーニングがディセーブルになります。

- Restrict（制限）：セキュア MAC アドレスの数がポートで許可されている最大限度に達すると、最大値を下回るまで十分な数のセキュア MAC アドレスを削除するか、許可アドレス数を増やさないかぎり、未知の送信元アドレスを持つパケットはドロップされます。このモードでは、セキュリティ違反が発生したことが通知されます。SNMP トラップが送信されます。Syslog メッセージがロギングされ、違反カウンタが増加します。
- Shutdown（シャットダウン）：ポートセキュリティ違反により、インターフェイスが error-disabled になり、ただちにシャットダウンされます。その後、ポートの LED が消灯します。セキュアポートが error-disabled 状態の場合は、**errdisable recovery cause** *psecure-violation* グローバル コンフィギュレーション コマンドを入力してこの状態を解消するか、**shutdown** および **no shut down** インターフェイス コンフィギュレーション コマンドを入力して手動で再度有効にできます。これは、デフォルトのモードです。
- Shutdown VLAN（VLAN シャットダウン）：VLAN 単位でセキュリティ違反モードを設定するために使用します。このモードで違反が発生すると、ポート全体ではなく、VLAN が errdisable になります。

次の表に、ポートセキュリティをインターフェイスに設定した場合の違反モードおよび対処について示します。

表 *23*：セキュリティ違反モードの処置

違反モード	トラフィックの転送 [2]	SNMP トラップの送信	Syslog メッセージの送信	エラー メッセージの表示 [3]	違反カウンタの増加	ポートのシャットダウン
protect	なし	なし	なし	なし	なし	なし
restrict	なし	対応	対応	なし	対応	非対応
shutdown	なし	なし	なし	なし	対応	対応
shutdown vlan	なし	なし	対応	なし	対応	非対応 [4]

[2] 十分な数のセキュアMACアドレスを削除するまで未知の送信元アドレスを持つパケットがドロップされます。
[3] セキュリティ違反を引き起こすアドレスを手動で設定した場合、スイッチがエラー メッセージを返します。
[4] 違反が発生した VLAN のみシャットダウンします。

ポート セキュリティ エージング

ポート上のすべてのセキュア アドレスにエージング タイムを設定するには、ポート セキュリティ エージングを使用します。ポートごとに2つのタイプのエージングがサポートされています。

- absolute：指定されたエージングタイムの経過後に、ポート上のセキュアアドレスが削除されます。
- inactivity：指定されたエージングタイムの間、セキュアアドレスが非アクティブであった場合に限り、ポート上のセキュアアドレスが削除されます。

デフォルトのポート セキュリティ設定

表 *24*：デフォルトのポート セキュリティ設定

\	
機能	**デフォルト設定**
ポート セキュリティ	ポート上でディセーブル
スティッキー アドレス ラーニング	ディセーブル

\	
機能	**デフォルト設定**
ポートあたりのセキュア MAC アドレスの最大数	1
違反モード	shutdown。セキュア MAC アドレスが最大数を上回ると、ポートがシャットダウンします。
ポート セキュリティ エージング	ディセーブルエージング タイムは 0 スタティック エージングはディセーブル タイプは absolute

ポート セキュリティの設定時の注意事項

- ポート セキュリティを設定できるのは、スタティック アクセス ポートまたはトランク ポートに限られます。セキュア ポートをダイナミック アクセス ポートにすることはできません。
- セキュア ポートをスイッチド ポート アナライザ（SPAN）の宛先ポートにすることはできません。
- 音声 VLAN はアクセス ポートでのみサポートされており、設定可能であってもトランク ポートではサポートされていません。
- 音声 VLAN が設定されたインターフェイス上でポート セキュリティをイネーブルにする場合は、ポートの最大セキュア アドレス許容数を 2 に設定します。ポートを Cisco IP Phone に接続する場合は、IP Phone に MAC アドレスが 1 つ必要です。Cisco IP Phone のアドレスは音声 VLAN 上で学習されますが、アクセス VLAN 上では学習されません。1 台の PC を Cisco IP Phone に接続する場合、MAC アドレスの追加は必要ありません。複数の PC を Cisco IP Phone に接続する場合、各 PC と IP Phone に 1 つずつ使用できるように、十分な数のセキュア アドレスを設定する必要があります。
- トランクポートがポートセキュリティで設定され、データトラフィック用のアクセスVLANと音声トラフィック用の音声 VLAN に割り当てられている場合、**switchport voice** およびインターフェイス コンフィギュレーション コマンドを入力して **switchport priority extend** も効果はありません。

  接続装置が同じ MAC アドレスを使用してアクセス VLAN の IP アドレス、音声 VLAN の IP アドレスの順に要求すると、アクセス VLAN だけが IP アドレスに割り当てられます。
- インターフェイスの最大セキュアアドレス値を入力したときに、新しい値がそれまでの値より大きいと、それまで設定されていた値が新しい値によって上書きされます。新しい値が前回の値より小さく、インターフェイスで設定されているセキュアアドレス数が新しい値より大きい場合、コマンドは拒否されます。

- デバイスはスティッキーセキュア MAC アドレスのポート セキュリティ エージングをサポートしていません。

次の表に、他のポートベース機能と互換性のあるポート セキュリティについてまとめます。

表 *25*：他の機能とポートセキュリティとの互換性

ポート タイプまたはポートの機能	ポート セキュリティとの互換性
DTP [5] ポート [6]	なし
トランク ポート	あり
ダイナミックアクセスポート[7]	なし
ルーテッド ポート	なし
SPAN 送信元ポート	あり
SPAN 宛先ポート	なし
EtherChannel	対応
トンネリング ポート	あり
保護ポート	あり
IEEE 802.1x ポート	あり
音声 VLAN ポート[8]	あり
IP ソース ガード	あり
ダイナミック アドレス解決プロトコル（ARP）インスペクション	あり

[5] DTP = Dynamic Trunking Protocol
[6] **switchport mode dynamic** インターフェイス コンフィギュレーション コマンドで設定されたポート A。
[7] **switchport access vlan dynamic** インターフェイス コンフィギュレーション コマンドで設定される VLAN Query Protocol（VQP）ポート。
[8] ポートに最大限可能なセキュアなアドレスを設定します（アクセス VLAN で可能なセキュアなアドレスの最大数に 2 を加えた数）。

ポート セキュリティの設定方法

ポート セキュリティのイネーブル化および設定

始める前に

このタスクは、ポートにアクセスできるステーションの MAC アドレスを制限および識別して、インターフェイスへの入力を制約します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**switchport port-security mac-address forbidden** *mac address* 例： `Device(config-if)# switchport port-security mac-address forbidden 2.2.2`	すべてのインターフェイスのポート セキュリティで禁止する MAC アドレスを指定します。
ステップ 4	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 5	**switchport mode** {**access** | **trunk**} 例： `Device(config-if)# switchport mode access`	インターフェイス スイッチポート モードを access または trunk に設定します。デフォルト モード（dynamic auto）のインターフェイスは、セキュア ポートとして設定できません。

	コマンドまたはアクション	目的
ステップ 6	**switchport voice vlan** *vlan-id* 例： `Device(config-if)# switchport voice vlan 22`	ポート上で音声 VLAN をイネーブルにします。 • *vlan-id*：音声トラフィックに使用する VLAN を指定します。
ステップ 7	**switchport port-security** 例： `Device(config-if)# switchport port-security`	インターフェイス上でポートセキュリティをイネーブルにします。
ステップ 8	**switchport port-security [maximum** *value* [**vlan** {*vlan-list* | {**access** | **voice**}}]] 例： `Device(config-if)# switchport port-security maximum 20`	（任意）インターフェイスの最大セキュア MAC アドレス数を設定します。スイッチに設定できるセキュア MAC アドレスの最大数は、システムで許可されている MAC アドレスの最大数によって決まります。この値は、使用可能な MAC アドレス（その他のレイヤ 2 機能やインターフェイスに設定されたその他のセキュア MAC アドレスで使用される MAC アドレスを含む）の総数を表します。 （任意）**vlan**：VLAN あたりの最大値を設定します **vlan** キーワードを入力後、次のいずれかのオプションを入力します。 • *vlan-list*：トランクポート上で、ハイフンで区切った範囲の VLAN、またはカンマで区切った一連の VLAN における、VLAN 単位の最大値を設定できます。VLAN を指定しない場合、VLAN ごとの最大値が使用されます。 • **access**：アクセスポートで、VLAN をアクセス VLAN として指定します。 • **voice**：アクセスポートで、VLAN を音声 VLAN として指定します。

	コマンドまたはアクション	目的
		（注） **voice** キーワードは、音声 VLAN がポートに設定されていて、さらにそのポートがアクセス VLAN でない場合のみ有効です。インターフェイスに音声 VLAN が設定されている場合、セキュア MAC アドレスの最大数を 2 に設定します。
ステップ 9	**switchport port-security violation** {**protect** | **restrict** | **shutdown** | **shutdown vlan**} 例： `Device(config-if)# switchport port-security violation restrict`	（任意）違反モードを設定します。セキュリティ違反が発生した場合に、次のいずれかのアクションを実行します。 • **protect**：ポート セキュア MAC アドレスの数がポートで許可されている最大限度に達すると、最大値を下回るまで十分な数のセキュア MAC アドレスを削除するか、許可アドレス数を増やさない限り、未知の送信元アドレスを持つパケットはドロップされます。セキュリティ違反が起こっても、ユーザには通知されません。 （注） トランク ポート上に保護モードを設定することは推奨できません。保護モードでは、ポートが最大数に達していなくても VLAN が保護モードの最大数に達すると、ラーニングがディセーブルになります。 • **restrict**：セキュア MAC アドレス数がポートで許可されている最大限度に達すると、十分な数のセキュア MAC アドレスを削除するか、許可アドレス数を増やさない限り、未知の送信元アドレスを持つパケットはドロップされます。SNMP トラップが送信されます。

	コマンドまたはアクション	目的
		Syslog メッセージがロギングされ、違反カウンタが増加します。 • **shutdown**：違反が発生すると、インターフェイスが error-disabled になり、ポートの LED が消灯します。SNMP トラップが送信されます。Syslog メッセージがロギングされ、違反カウンタが増加します。 • **shutdown vlan**：VLAN 単位でセキュリティ違反モードを設定するために使用します。このモードで違反が発生すると、ポート全体ではなく、VLAN が errdisable になります。 （注） セキュア ポートが error-disabled ステートの場合は、**errdisable recovery cause psecure-violation** グローバル コンフィギュレーション コマンドを入力して、このステートから回復させることができます。手動で再びイネーブルにするには、**shutdown** および **no shutdown** インターフェイス コンフィギュレーション コマンドを入力するか、**clear errdisable interface vlan** 特権 EXEC コマンドを入力します。
ステップ 10	**switchport port-security [mac-address** *mac-address* [**vlan** {*vlan-id* | {**access** | **voice**}}] 例： `Device(config-if)# switchport port-security mac-address 00:A0:C7:12:C9:25 vlan 3 voice`	（任意）インターフェイスのセキュア MAC アドレスを入力します。このコマンドを使用すると、最大数のセキュア MAC アドレスを入力できます。設定したセキュア MAC アドレスが最大数より少ない場合、残りの MAC アドレスは動的に学習されます。

	コマンドまたはアクション	目的
		（注） このコマンドの入力後にスティッキー ラーニングをイネーブルにすると、動的に学習されたセキュアアドレスがスティッキー セキュア MAC アドレスに変換されて実行コンフィギュレーションに追加されます。 （任意）**vlan**：VLAN あたりの最大値を設定します。 **vlan** キーワードを入力後、次のいずれかのオプションを入力します。 • *vlan-id*：トランクポートで、VLAN ID および MAC アドレスを指定できます。VLAN ID を指定しない場合、ネイティブ VLAN が使用されます。 • **access**：アクセスポートで、VLAN をアクセス VLAN として指定します。 • **voice**：アクセスポートで、VLAN を音声 VLAN として指定します。 （注） **voice** キーワードは、音声 VLAN がポートに設定されていて、さらにそのポートがアクセス VLAN でない場合のみ有効です。インターフェイスに音声 VLAN が設定されている場合、セキュア MAC アドレスの最大数を 2 に設定します。
ステップ 11	**switchport port-security mac-address sticky** 例： `Device(config-if)# switchport port-security mac-address sticky`	（任意）インターフェイス上でスティッキ ラーニングをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 12	**switchport port-security mac-address sticky** [*mac-address* | **vlan** {*vlan-id* | {**access** | **voice**}}] 例： `Device(config-if)# switchport port-security mac-address sticky 00:A0:C7:12:C9:25 vlan voice`	（任意）スティッキー セキュア MAC アドレスを入力し、必要な回数だけコマンドを繰り返します。設定したセキュア MAC アドレスの数が最大数より少ない場合、残りの MAC アドレスは動的に学習されてスティッキー セキュア MAC アドレスに変換され、実行コンフィギュレーションに追加されます。 （注） このコマンドの入力前にスティッキー ラーニングをイネーブルにしないと、エラーメッセージが表示されてスティッキー セキュア MAC アドレスを入力できません。 （任意）**vlan**：VLAN あたりの最大値を設定します。 **vlan** キーワードを入力後、次のいずれかのオプションを入力します。 • *vlan-id*：トランクポートで、VLAN ID および MAC アドレスを指定できます。VLAN ID を指定しない場合、ネイティブ VLAN が使用されます。 • **access**：アクセスポートで、VLAN をアクセス VLAN として指定します。 • **voice**：アクセスポートで、VLAN を音声 VLAN として指定します。 （注） **voice** キーワードは、音声 VLAN がポートに設定されていて、さらにそのポートがアクセス VLAN でない場合のみ有効です。
ステップ 13	**switchport port-security mac-address forbidden** *mac address* 例：	特定のインターフェイスのポート セキュリティで禁止する MAC アドレスを指定します。

	コマンドまたはアクション	目的
	`Device(config-if)# switchport port-security mac-address forbidden 2.2.2`	
ステップ 14	**end** 例： `Device(config-f)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 15	**show port-security** 例： `Device# show port-security`	ポートセキュリティ設定に関する情報を表示します。

ポート セキュリティ エージングのイネーブル化および設定

この機能を使用すると、既存のセキュア MAC アドレスを手動で削除しなくても、セキュアポート上のデバイスを削除および追加し、なおかつポート上のセキュア アドレス数を制限できます。セキュア アドレスのエージングは、ポート単位でイネーブルまたはディセーブルにできます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**switchport port-security aging** {**static** | **time** *time* | **type absolute** 例： `Device(config-if)# switchport port-security aging time 120`	セキュア ポートのスタティック エージングをイネーブルまたはディセーブルにします。またはエージング タイムやタイプを設定します。 （注） デバイスはスティッキセキュアアドレスのポート セキュリティ エージングをサポートしていません。 • このポートに、スタティックに設定されたセキュアアドレスのエージングをイネーブルにする場合は、**static** キーワードを入力します。 • *time* 引数は、このポートのエージングタイムを指定します。有効値の範囲は 0 ～ 1440 分です。 • **type absolute**：エージングタイプを絶対エージングとして設定します。このポートのセキュア アドレスはすべて、指定した時間（分単位）が経過すると期限切れになり、セキュア アドレス リストから削除されます。
ステップ 5	**end** 例： `Device(config-f)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 6	**show port-security** 例： `Device# show port-security`	ポートセキュリティ設定に関する情報を表示します。

ポート セキュリティの監視

次の表に、ポート セキュリティ情報を表示します。

表 *26 :* ポート セキュリティのステータスおよび設定を表示するコマンド

コマンド	目的
show port-security [**interface** *interface-id*]	スイッチまたは指定されたインターフェイスのポートセキュリティ設定を、各インターフェイスで許容されるセキュアMACアドレスの最大数、インターフェイスのセキュアMACアドレスの数、発生したセキュリティ違反の数、違反モードを含めて表示します。
show port-security [**interface** *interface-id*] **address**	すべてのスイッチ インターフェイスまたは指定されたインターフェイスに設定されたすべてのセキュアMACアドレス、および各アドレスのエージング情報を表示します。
show port-security interface *interface-id* **vlan**	指定されたインターフェイスに VLAN 単位で設定されているセキュアMACアドレスの数を表示します。

ポート セキュリティの設定例

例：ポートセキュリティのイネーブル化および設定

次に、ポート上でポート セキュリティをイネーブルにし、セキュア アドレスの最大数を 50 に設定する例を示します。違反モードはデフォルトです。スタティック セキュアMAC アドレスは設定せず、スティッキー ラーニングはイネーブルです。

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# switchport mode access
Device(config-if)# switchport port-security
Device(config-if)# switchport port-security maximum 50
Device(config-if)# switchport port-security mac-address sticky
Device(config-if)# end
```

次に、ポートの VLAN 3 上にスタティック セキュア MAC アドレスを設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# switchport mode trunk
Device(config-if)# switchport port-security
```

```
Device(config-if)# switchport port-security mac-address 0000.0200.0004 vlan 3
Device(config-if)# end
```

次に、ポートのスティッキー ポート セキュリティをイネーブルにする例を示します。データ VLAN および音声 VLAN の MAC アドレスを手動で設定し、セキュア アドレスの総数を 20 に設定します（データ VLAN に 10、音声 VLAN に 10 を割り当てます）。

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# switchport access vlan 21
Device(config-if)# switchport mode access
Device(config-if)# switchport voice vlan 22
Device(config-if)# switchport port-security
Device(config-if)# switchport port-security maximum 20
Device(config-if)# switchport port-security violation restrict
Device(config-if)# switchport port-security mac-address sticky
Device(config-if)# switchport port-security mac-address sticky 0000.0000.0002
Device(config-if)# switchport port-security mac-address 0000.0000.0003
Device(config-if)# switchport port-security mac-address sticky 0000.0000.0001 vlan voice
Device(config-if)# switchport port-security mac-address 0000.0000.0004 vlan voice
Device(config-if)# switchport port-security maximum 10 vlan access
Device(config-if)# switchport port-security maximum 10 vlan voice
Device(config-if)# end
```

例：ポート セキュリティ エージングのイネーブル化および設定

次の例では、ポートセキュリティエージングのイネーブル化方法と設定方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# switchport port-security aging time 120
evice(config-f)# end
```

その他の参考資料

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

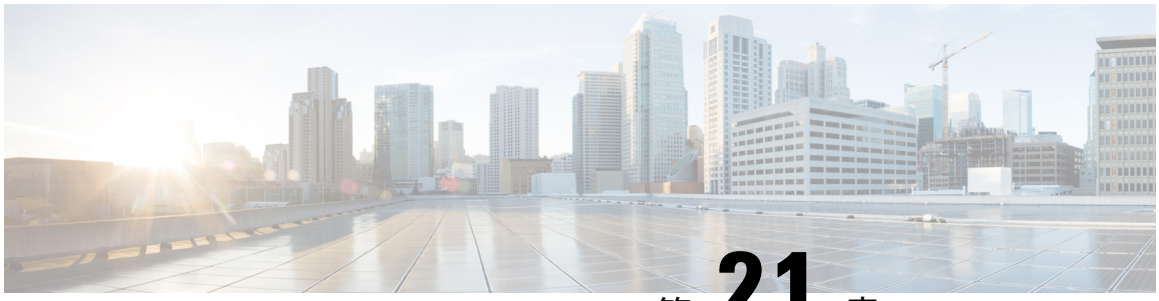
ポート セキュリティの機能の履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	ポート セキュリティ	ポートセキュリティ機能で、ポートへのアクセスを許可するステーションの MAC アドレスを制限および識別して、インターフェイスへの入力を制限します。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 21 章

ポート ブロッキングの設定

ポート ブロッキングに関する情報

デフォルトでは、スイッチは未知の宛先 MAC アドレスが指定されたパケットをすべてのポートからフラッディングします。未知のユニキャストおよびマルチキャスト トラフィックが保護ポートに転送されると、セキュリティ上、問題になる可能性があります。未知のユニキャストおよびマルチキャスト トラフィックがあるポートから別のポートに転送されないようにするために、（保護または非保護）ポートをブロックし、未知のユニキャストまたはマルチキャスト パケットが他のポートにフラッディングされないようにします。

インターフェイスでのフラッディング トラフィックのブロッキング

インターフェイスでフラッディング トラフィックをブロックするには、次の手順を実行します。

始める前に

インターフェイスは物理インターフェイスまたは EtherChannel グループのいずれも可能です。ポート チャネルのマルチキャストまたはユニキャスト トラフィックをブロックすると、ポート チャネル グループのすべてのポートでブロックされます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**switchport block multicast** 例： `Device(config-if)# switchport block multicast`	ポートからの未知のマルチキャストの転送をブロックします。 （注） ヘッダーに IPv6 情報を含むマルチキャスト パケットだけでなく、純粋なレイヤ 2 マルチキャスト トラフィックもブロックされます。
ステップ 5	**switchport block unicast** 例： `Device(config-if)# switchport block unicast`	ポートからの未知のユニキャストの転送をブロックします。
ステップ 6	**end** 例： `Device(config-line)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show interfaces** *interface-id* **switchport** 例： `Device# show interfaces gigabitethernet 1/0/2 switchport`	入力を確認します。
ステップ 8	**show running-config** 例： `Device# show running-config`	入力を確認します。

	コマンドまたはアクション	目的
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ポート ブロッキングの監視

表 *27*：ポート ブロッキングの設定を表示するコマンド

コマンド	目的
show interfaces [*interface-id*] **switchport**	すべてのスイッチング（非ルーティング）ポートまたは指定されたポートの管理ステータスまたは動作ステータスを、ポート ブロッキングおよびポート保護の設定を含めて表示します。

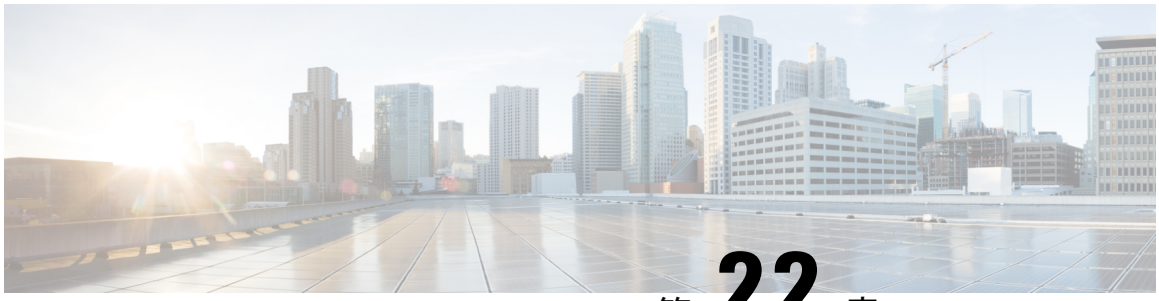
ポートブロッキングの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	ポート ブロッキング	未知のユニキャストおよびマルチキャスト トラフィックがあるポートから別のポートに転送されないようにするために、（保護または非保護）ポートをブロックし、未知のユニキャストまたはマルチキャスト パケットが他のポートにフラッディングされないようにします。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 22 章

保護ポートの設定

保護ポートに関する情報

次の各項では、保護ポートについて説明します。

保護ポート

アプリケーションによっては、あるネイバーが生成したトラフィックが別のネイバーにわからないように、同一スイッチ上のポート間でレイヤ2トラフィックが転送されないように設定する必要があります。このような環境では、保護ポートを使用すると、スイッチ上のポート間でユニキャスト、ブロードキャスト、またはマルチキャストトラフィックの交換が確実になくなります。

保護ポートには、次の機能があります。

- 保護ポートは、同様に保護ポートになっている他のポートに対して、ユニキャスト、マルチキャスト、またはブロードキャスト トラフィックを転送しません。データ トラフィックはレイヤ 2 の保護ポート間で転送されません。PIM パケットなどは CPU で処理されてソフトウェアで転送されるため、このような制御トラフィックだけが転送されます。保護ポート間を通過するすべてのデータ トラフィックは、レイヤ3デバイスを介して転送されなければなりません。
- 保護ポートと非保護ポート間の転送動作は、通常どおりに進みます。

保護ポートのデフォルト設定

デフォルトでは、保護ポートは定義されません。

保護ポートのガイドライン

保護ポートは、物理インターフェイス（GigabitEthernet ポート 1 など）または EtherChannel グループ（port-channel 5 など）に設定できます。ポート チャネルで保護ポートをイネーブルにした場合は、そのポート チャネル グループ内のすべてのポートでイネーブルになります。

保護ポートの設定方法

次の項では、保護ポートの設定について説明します。

保護ポートの設定

保護ポートを設定するには、次の手順を実行します。

始める前に

保護ポートは事前定義されていません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**switchport protected** 例： `Device(config-if)# switchport protected`	インターフェイスを保護ポートとして設定します。
ステップ 5	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 6	**show interfaces** *interface-id* **switchport** 例： `Device(config)# show interfaces gigabitethernet 1/0/2 switchport`	入力を確認します。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

保護ポートの監視

表 *28 :* 保護ポートの設定を表示するコマンド

コマンド	目的
show interfaces [*interface-id*] **switchport**	すべてのスイッチング（非ルーティング）ポートまたは指定されたポートの管理ステータスまたは動作ステータスを、ポート ブロッキングおよびポート保護の設定を含めて表示します。

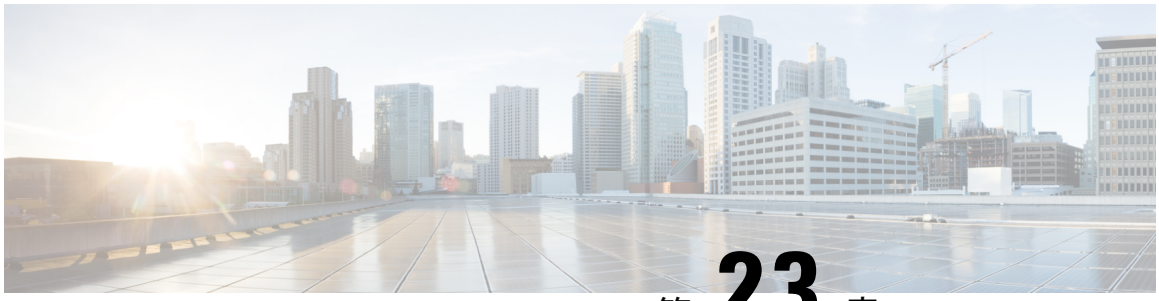
保護ポートの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	保護ポート	保護ポートを使用すると、スイッチ上のポート間でユニキャスト、ブロードキャスト、またはマルチキャストトラフィックの交換が確実になくなります。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 23 章

プロトコル ストーム プロテクションの設定

プロトコル ストーム プロテクションの設定の制約事項

仮想ポートの errdisable は、EtherChannel インターフェイスではサポートされません。

プロトコル ストーム プロテクションに関する情報

スイッチがアドレス解決プロトコル（ARP）または制御パケットでフラッドされると、CPU の高い使用率により CPU のオーバーロードが発生する可能性があります。これらの問題は、次のように発生します。

- プロトコル制御パケットが受信されず、ネイバーの隣接がドロップされるため、ルーティング プロトコルがフラップする場合があります。
- スパニングツリー プロトコル（STP）ブリッジ プロトコル データ ユニット（BPDU）が送受信されないため、STP が再収束します。
- CLI が遅くなるか応答しなくなります。

プロトコル ストーム プロテクションを使用すると、パケットのフロー レートの上限しきい値を指定して、制御パケットが送信されるレートを制御できます。サポートされるプロトコルは、ARP、ARP スヌーピング、Dynamic Host Configuration Protocol（DHCP）v4、DHCP スヌーピング、インターネット グループ管理プロトコル（IGMP）、および IGMP スヌーピングです。

パケットのレートが定義されたしきい値を超えると、スイッチは指定されたポートに着信したすべてのトラフィックを 30 秒間ドロップします。パケット レートが再度計測され、必要な場合はプロトコル ストーム プロテクションが再度適用されます。

より強力な保護が必要な場合は、仮想ポートを手動で errdisable にし、その仮想ポートのすべての着信トラフィックをブロックできます。また、手動で仮想ポートをイネーブルにしたり、仮想ポートの自動再イネーブル化の時間間隔を設定したりすることもできます。

（注） 超過したパケットは、2 つ以下の仮想ポートにおいてドロップされます。

プロトコル ストーム プロテクションはデフォルトでディセーブルです。これがイネーブルになると、仮想ポートの自動リカバリがデフォルトでディセーブルになります。

プロトコルストームプロテクションのイネーブル化方法

プロトコル ストーム プロテクションをイネーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**psp** {**arp** | **dhcp** | **igmp**} pps *value* 例： `Device(config)# psp dhcp pps 35`	ARP、IGMP、または DHCP に対してプロトコル ストーム プロテクションを設定します。 *value*：1 秒あたりのパケット数のしきい値を指定します。トラフィックがこの値を超えると、プロトコル ストーム プロテクションが適用されます。範囲は毎秒 5 ～ 50 パケットです。
ステップ 4	**errdisable detect cause psp** 例： `Device(config)# errdisable detect cause psp`	（任意）プロトコル ストーム プロテクションの errdisable 検出をイネーブルにします。この機能がイネーブルになると、仮想ポートが errdisable になります。この機能がディセーブルになると、そのポートは、ポートを errdisable にせ

	コマンドまたはアクション	目的
		ずに超過したパケットをドロップします。
ステップ 5	**errdisable recovery interval** *time* 例： `Device(config)# errdisable recovery interval 100`	（任意）error-disabled の仮想ポートの自動リカバリ時間を秒単位で設定します。仮想ポートが error-disabled の場合、この時間を過ぎるとスイッチは自動的にリカバリします。指定できる範囲は 30 ～ 86400 秒です。
ステップ 6	**end** 例： `Device(config-line)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show psp config** {**arp** | **dhcp** | **igmp**} 例： `Device# show psp config dhcp`	入力を確認します。

プロトコル ストーム プロテクションのモニタリング

表 *29 :* エントリを検証するためのコマンド

コマンド	目的
show psp config {**arp** | **dhcp** | **igmp**}	入力内容を確認します。

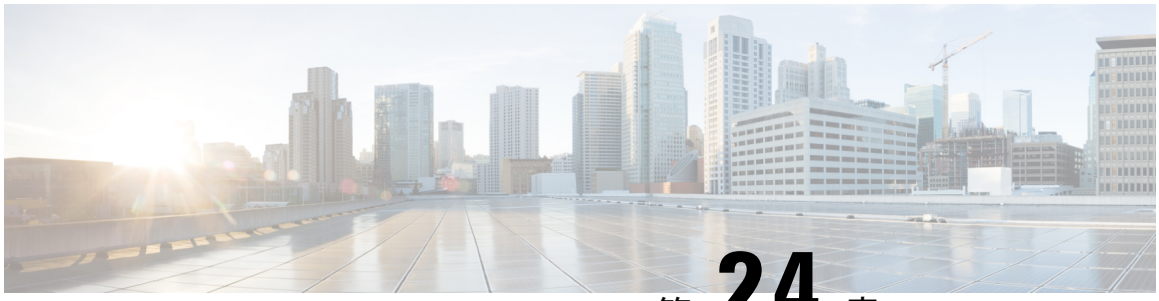
プロトコル ストーム プロテクションの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	プロトコル ストーム プロテクション	プロトコル ストーム プロテクションを使用すると、パケットのフロー レートの上限しきい値を指定して、制御パケットが送信されるレートを制御できます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn [英語] からアクセスします。



第 24 章

ストーム制御の設定

ストーム制御に関する情報

ストーム制御

ストーム制御は、物理インターフェイスの 1 つで発生したブロードキャスト、マルチキャスト、またはユニキャスト ストームによって LAN 上のトラフィックが混乱することを防ぎます。LAN ストームは、LAN にパケットがフラッディングした場合に発生します。その結果、トラフィックが極端に増えてネットワーク パフォーマンスが低下します。プロトコルスタックの実装エラー、ネットワーク構成の間違い、またはユーザによって引き起こされる DoS 攻撃もストームの原因になります。

ストーム コントロール（またはトラフィック抑制）は、インターフェイスからスイッチング バスを通過するパケットをモニタし、パケットがユニキャスト、マルチキャスト、またはブロードキャストのいずれであるかを判別します。デバイスは、1 秒間に受け取った特定のタイプのパケット数をカウントして、事前に定義された抑制レベルのしきい値とその測定結果を比較します。

トラフィック アクティビティの測定方法

ストーム コントロールは、次のうちのいずれかをトラフィック アクティビティの測定方法に使用します。

- 帯域幅（ブロードキャスト、マルチキャスト、またはユニキャスト トラフィックが使用できるポートの総帯域幅の割合）。

- 秒単位で受信するパケット（ブロードキャスト、マルチキャスト、またはユニキャスト）のトラフィック レート
- 秒単位で受信するビット（ブロードキャスト、マルチキャスト、またはユニキャスト）のトラフィック レート

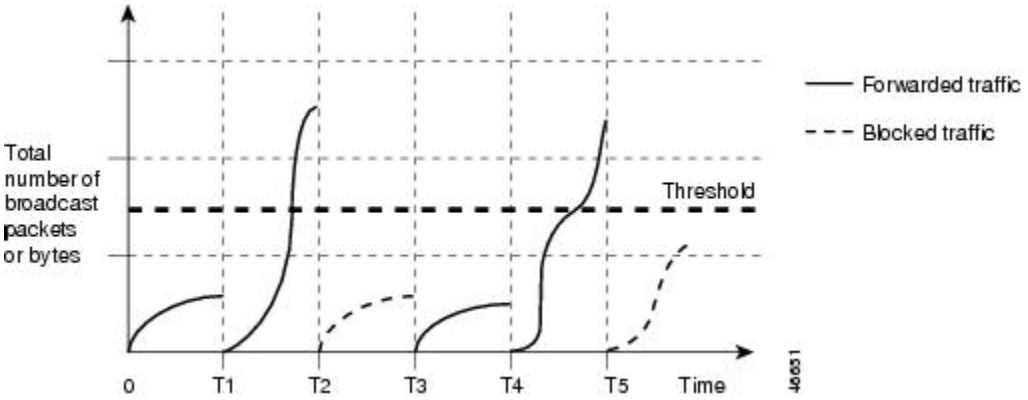
上記の方法のいずれを使用しても、しきい値に到達すると、ポートはトラフィックをブロックします。トラフィック レートが下限しきい値（指定されている場合）を下回らない限り、ポートはブロックされたままになり、その後、通常の転送が再開されます。下限抑制レベルが指定されていない場合、トラフィックレートが上限抑制レベルを下回るまで、デバイスはすべてのトラフィックをブロックします。一般に、そのレベルが高ければ高いほど、ブロードキャスト ストームに対する保護効果は薄くなります。

（注） マルチキャストトラフィックのストーム制御しきい値に達した場合、ブリッジプロトコルデータ ユニット（BPDU）および Cisco Discovery Protocol フレームなどの制御トラフィック以外のマルチキャストトラフィックはすべてブロックされます。ただし、デバイスでは Open Shortest Path First（OSPF）などのルーティングアップデートと、正規のマルチキャスト データ トラフィックは区別されないため、両方のトラフィックタイプがブロックされます。

トラフィック パターン

図 *13 :* ブロードキャスト ストーム制御の例

次の例は、一定時間におけるインターフェイス上のブロードキャスト トラフィック パターンを示しています。



T1 から T2、T4 から T5 のタイム インターバルで、転送するブロードキャスト トラフィックが設定されたしきい値を上回っています。指定のトラフィック量がしきい値を上回ると、次のインターバルで、そのタイプのトラフィックがすべてドロップされます。したがって、T2 と T5 の後のインターバルの間、ブロードキャスト トラフィックがブロックされます。その次のインターバル（たとえば、T3）では、しきい値を上回らない限り、ブロードキャスト トラフィックが再び転送されます。

ストーム制御抑制レベルと1秒間のインターバルを組み合わせて、ストーム制御アルゴリズムの動作を制御します。しきい値が高いほど、通過できるパケット数が多くなります。しきい値が 100% であれば、トラフィックに対する制限はありません。値を 0.0 にすると、そのポート上ではすべてのブロードキャスト、マルチキャスト、またはユニキャスト トラフィックがブロックされます。

（注） パケットは一定の間隔で届くわけではないので、トラフィック アクティビティを測定する1秒間のインターバルがストーム制御の動作を左右する可能性があります。

各トラフィックタイプのしきい値を設定するには、**storm-control** インターフェイス コンフィギュレーション コマンドを使用します。

ストーム制御の設定方法

ストーム制御およびしきい値レベルの設定

ポートにストーム制御を設定し、特定のトラフィック タイプで使用するしきい値レベルを入力します。

ただし、ハードウェアの制約とともに、さまざまなサイズのパケットをどのように数えるかという問題があるので、しきい値の割合はあくまでも近似値です。着信トラフィックを形成するパケットのサイズによって、実際に適用されるしきい値は設定されたレベルに対して、数 % の差異が生じる可能性があります。

（注） ストーム制御は、物理インターフェイスでサポートされています。また、EtherChannel でもストーム制御を設定できます。ストーム制御を EtherChannel で設定する場合、ストーム制御設定は EtherChannel 物理インターフェイスに伝播します。

ストーム制御としきい値レベルを設定するには、次の手順を実行します。

始める前に

ストーム制御は、物理インターフェイスでサポートされています。また、EtherChannel でもストーム制御を設定できます。ストーム制御を EtherChannel で設定する場合、ストーム制御設定は EtherChannel 物理インターフェイスに伝播します。

手順

	コマンドまたはアクション	目的
ステップ1	**enable** 例：	特権 EXEC モードを有効にします。

	コマンドまたはアクション	目的
	`Device> enable`	• パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/2`	設定するインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**storm-control action** {**shutdown** | **trap**} 例： `Device(config-if)# storm-control action trap`	ストーム検出時に実行するアクションを指定します。デフォルトではトラフィックにフィルタリングを実行し、トラップは送信しない設定です。 • ストーム中、ポートを errdisable の状態にするには、**shutdown** キーワードを選択します。 • ストームが検出された場合、SNMP トラップを生成するには、**trap** キーワードを選択します。
ステップ 5	**storm-control** {**broadcast** | **multicast** | **unicast**} **level** {*level* [*level-low*] | **bps** *bps* [*bps-low*] | **pps** *pps* [*pps-low*]} 例： `Device(config-if)# storm-control unicast level 87 65`	ブロードキャスト、マルチキャスト、またはユニキャスト ストーム制御を設定します。デフォルトでは、ストーム制御はディセーブルに設定されています。 キーワードの意味は次のとおりです。 • *level* には、ブロードキャスト、マルチキャスト、またはユニキャスト トラフィックの上限しきい値レベルを帯域幅のパーセンテージで指定します（小数点第2位まで）。上限しきい値に到達すると、ポートはトラフィックをブロックします。指定できる範囲は 0.00 ～ 100.00 です。 • （任意）*level-low* には、下限しきい値レベルを帯域幅のパーセンテージで指定します（小数点第 2 位まで）。この値は上限抑制値より小さ

	コマンドまたはアクション	目的
		いか、または等しくなければなりません。トラフィックがこのレベルを下回っていれば、ポートはトラフィックを転送します。下限抑制レベルを設定しない場合、上限抑制レベルの値に設定されます。指定できる範囲は 0.00 ～ 100.00 です。 しきい値に最大値（100%）を指定した場合、トラフィックの制限はなくなります。しきい値に 0.0 を設定すると、そのポート上のすべてのブロードキャスト、マルチキャスト、またはユニキャスト トラフィックがブロックされます。 • **bps** *bps* には、ブロードキャスト、マルチキャスト、またはユニキャストトラフィックの上限しきい値レベルをビット/秒で指定します（小数点第1位まで）。上限しきい値に到達すると、ポートはトラフィックをブロックします。指定できる範囲は 0.0 ～ 10000000000.0 です。 • （任意）*bps-low* には、下限しきい値レベルをビット/秒で指定します（小数点第1位まで）。この値は上限しきい値レベル以下の値である必要があります。トラフィックがこのレベルを下回っていれば、ポートはトラフィックを転送します。指定できる範囲は 0.0 ～ 10000000000.0 です。 • **pps** *pps* には、ブロードキャスト、マルチキャスト、またはユニキャストトラフィックの上限しきい値レベルをパケット/秒で指定します（小数点第1位まで）。上限しきい値に到達すると、ポートはトラフィックをブロックします。指定できる範囲は 0.0 ～ 10000000000.0 です。 • （任意）*pps-low* には、下限しきい値レベルをパケット/秒で指定しま

	コマンドまたはアクション	目的
		す（小数点第1位まで）。この値は上限しきい値レベル以下の値である必要があります。トラフィックがこのレベルを下回っていれば、ポートはトラフィックを転送します。指定できる範囲は 0.0 ～ 10000000000.0 です。 BPS および PPS の設定には、しきい値の数値を大きく設定できるように、サフィックスに測定記号（k、m、gなど）を使用できます。
ステップ 6	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show storm-control** [*interface-id*] [**broadcast** | **multicast** | **unicast**] 例： `Device# show storm-control gigabitethernet 1/0/2 unicast`	指定したトラフィック タイプについて、インターフェイスで設定したストーム制御抑制レベルを確認します。トラフィック タイプを入力しない場合は、すべてのトラフィック タイプ（ブロードキャスト、マルチキャスト、ユニキャスト）の詳細が表示されます。

ストーム制御の設定例

例：ストーム制御およびしきい値レベルの設定

次に、ストーム制御としきい値レベルの設定例を示します。

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# storm-control action trap
Device(config-if)# storm-control unicast level 87 65
Device(config-if)# end
Device# show storm-control gigabitethernet 1/0/1 unicast
```

ストーム制御に関する追加情報

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst Micro Switches)*

シスコのテクニカル サポート

説明	リンク
シスコのサポートWebサイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

ストーム制御の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS Release 15.2(7)E3k	ストーム制御	ストーム制御は、物理インターフェイスの1つで発生したブロードキャスト、マルチキャスト、またはユニキャストストームによってLAN上のトラフィックが混乱することを防ぎます。

Cisco Feature Navigatorを使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigatorには、http://www.cisco.com/go/cfn [英語] からアクセスします。