Cisco Catalyst IE9300 高耐久性シリーズ スイッチ セキュリティ コンフィギュレーション ガイド

最終更新：2025 年 8 月 6 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/

Full Cisco Trademarks with Software License

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

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

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

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

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

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

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

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

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

通信、サービス、およびその他の情報

通信、サービス、およびその他の情報

- シスコからタイムリーな関連情報を受け取るには、Cisco Profile Manager でサインアップしてください。
- 重要な技術によりビジネスに必要な影響を与えるには、Cisco Services にアクセスしてください。
- サービス リクエストを送信するには、Cisco Support にアクセスしてください。
- 安全で検証済みのエンタープライズクラスのアプリケーション、製品、ソリューション、およびサービスを探して参照するには、Cisco DevNet [英語] にアクセスしてください。
- 一般的なネットワーキング、トレーニング、認定関連の出版物を入手するには、Cisco Press にアクセスしてください。
- 特定の製品または製品ファミリの保証情報を探すには、Cisco Warranty Finder にアクセスしてください。

Cisco バグ検索ツール

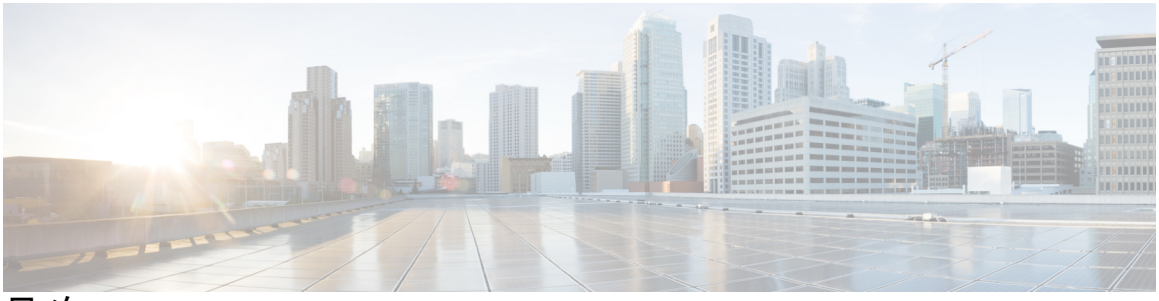
シスコバグ検索ツール（BST）は、シスコ製品とソフトウェアの障害と脆弱性の包括的なリストを管理するシスコバグ追跡システムへのゲートウェイです。BST は、製品とソフトウェアに関する詳細な障害情報を提供します。

マニュアルに関するフィードバック

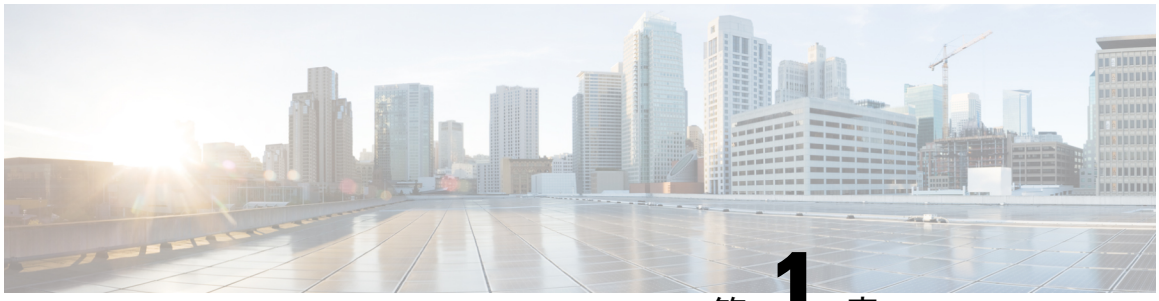
シスコのテクニカルドキュメントに関するフィードバックを提供するには、それぞれのオンラインドキュメントの右側のペインにあるフィードバックフォームを使用してください。

偏向のない言語

この製品のマニュアルセットは、偏向のない言語を使用するように配慮されています。このドキュメントセットでの偏向のない言語とは、年齢、障害、性別、人種的アイデンティティ、民族的アイデンティティ、性的指向、社会経済的地位、およびインターセクショナリティに基づく差別を意味しない言語として定義されています。製品ソフトウェアのユーザインターフェイスにハードコードされている言語、基準ドキュメントに基づいて使用されている言語、または参照されているサードパーティ製品で使用されている言語によりドキュメントに例外が存在する場合があります。



目 次



第 1 章

RADIUS の設定

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）を有効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有効にできます。
- **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 を有効にします。
- **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有効にします。
- 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。
- 通常、RADIUS ホストは、シスコ（Cisco ISE）、Livingston、Merit、Microsoft、または他のソフトウェアプロバイダーの RADIUS サーバー ソフトウェアを実行するマルチユーザー 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 の動作：

- ユーザーは 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（有効になっている場合）。
- RADIUS over IPv6 構成の場合、ユーザーは **ipv6 unicast-routing** コマンドを有効にして、IPv6 ユニキャストルーティングを有効にする必要があります。

RADIUS の設定に関する制約事項

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 および AAA サーバーは、標準のデフォルトポートでのみ実行するように設定できます。
  - 1812 および 1813
  - 1645 および 1646

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 パケットの DSCP マーキングサポート：

- 認証とアカウンティングの DSCP マーキングは、プライベートサーバー、完全修飾ドメイン名（FQDN）サーバー、および radsec サーバーではサポートされていません。
- 有線 IEEE 802.1x 認証の場合、送信元ポート拡張が有効になっていないと、デフォルトポートが使用されます。DSCP マーキングはデフォルトポートに設定され、すべての要求は同じ DSCP 値でマーキングされます。

- 送信元ポート拡張がデフォルトで有効になっている無線 IEEE 802.1x 認証の場合、DSCP マーキングはサポートされません。

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 を有効にし、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。

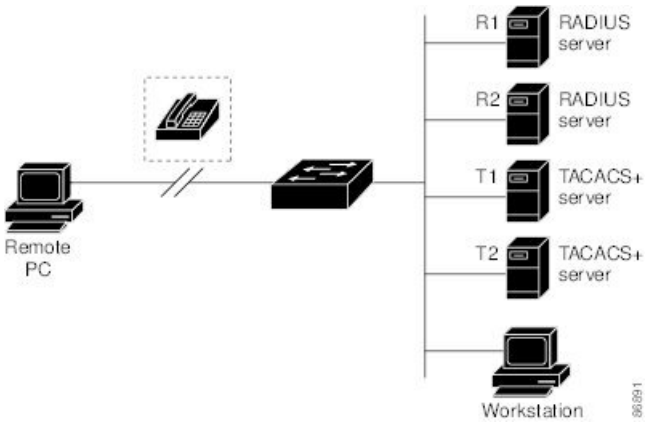
RADIUS の概要

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。

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。
- 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。

図 *1 : RADIUS* サービスから *TACACS+* サービスへの移行



- 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 ポートベースの認証の設定」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。

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：ユーザーが認証されたことを表します。
   - REJECT：ユーザーの認証が失敗し、ユーザー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：ユーザーに追加データを要求します。
   - CHALLENGE PASSWORD：ユーザー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 許可の変更

RADIUS 許可の変更（CoA）は、認証、認可、およびアカウンティング（AAA）セッションの属性を認証された後に変更するためのメカニズムを提供します。AAA でユーザー、またはユーザー グループのポリシーが変更された場合、管理者は、AAA サーバーから Cisco Secure Access Control Server（ACS）などの RADIUS CoA パケットを送信し、認証を再初期化して新しいポリシーを適用することができます。このセクションでは、使用可能なプリミティブおよびそれらの CoA での使用方法を含む、RADIUS インターフェイスの概要について説明します。

- Change-of-Authorization 要求
- CoA 要求応答コード
- CoA 要求コマンド
- セッション再認証
- セッション強制終了のスタック構成ガイドライン

標準 RADIUS インターフェイスは通常、ネットワークに接続しているデバイスから要求が送信され、クエリーが送信されたサーバーが応答するプル モデルで使用されます。シスコ デバイスは、RFC 5176 で規定された（通常はプッシュモデルで使用される）RADIUS CoA 拡張機能をサポートし、外部の AAA またはポリシーサーバーからのセッションを動的に再設定できるようにします。

シスコ デバイスは、次のセッション単位の CoA 要求をサポートしています。

- セッション再認証
- セッションの終了
- ポート シャットダウンでのセッション終了
- ポート バウンスでのセッション終了

この機能は、Cisco Secure Access Control Server（ACS）5.1 に統合されています。

シスコデバイスで、RADIUS インターフェイスはデフォルトで有効に設定されています。ただし、次の属性については、一部の基本的な設定が必要になります。

- セキュリティおよびパスワード：このガイドの「スイッチへの不正アクセスの防止」を参照してください。

- アカウンティング：このガイドの「スイッチベース認証の設定」の章の「RADIUS アカウンティングの起動」の項を参照してください。

Cisco IOS XE ソフトウェアは、RFC 5176 で定義されている RADIUS CoA の拡張をサポートします。この拡張は、一般に、外部 AAA またはポリシーサーバーからのセッションの動的な再構成を可能にするプッシュモデルで使用されます。セッションの特定、セッションの終了、ホストの再認証、ポートのシャットダウン、およびポートバウンスでは、セッションごとの CoA 要求がサポートされます。このモデルは、次のように、1 つの要求（CoA-Request）と 2 つの考えられる応答コードで構成されます。

- CoA acknowledgement（ACK）[CoA-ACK]
- CoA nonacknowledgement（NAK）[CoA-NAK]

要求は CoA クライアント（通常は AAA またはポリシー サーバー）から開始されて、リスナーとして動作するデバイスに転送されます。

次の表は、Identity-Based Networking Services でサポートされている RADIUS CoA コマンドとベンダー固有属性（VSA）を示します。すべての CoA コマンドには、デバイスと CoA クライアント間のセッション ID が含まれている必要があります。

表 *1 : Identity-Based Networking Services* でサポートされている *RADIUS CoA* コマンド

CoA コマンド	**シスコの VSA**
Activate service	Cisco:Avpair=“subscriber:command=activate-service” Cisco:Avpair=“subscriber:service-name=<*service-name*>” Cisco:Avpair=“subscriber:precedence=<*precedence-number*>” Cisco:Avpair=“subscriber:activation-mode=replace-all”
Deactivate service	Cisco:Avpair=“subscriber:command=deactivate-service” Cisco:Avpair=“subscriber:service-name=<*service-name*>”
Bounce host port	Cisco:Avpair="subscriber:command=bounce-host-port"
Disable host port	Cisco:Avpair="subscriber:command=disable-host-port"
Session query	Cisco:Avpair=“subscriber:command=session-query”
Session reauthenticate	Cisco:Avpair="subscriber:command=reauthenticate" Cisco:Avpair=“subscriber:reauthenticate-type=last” または Cisco:Avpair=“subscriber:reauthenticate-type=rerun”
Session terminate	これは、VSA を必要としない、標準の接続解除要求です。
Interface template	Cisco:AVpair="interface-template-name=<*interfacetemplate*>"

Change-of-Authorization 要求

Change of Authorization（CoA）要求は、RFC 5176 に記載されているように、プッシュ モデルで使用することによって、セッション識別、ホスト再認証、およびセッション終了を行うことができます。このモデルは、1 つの要求（CoA-Request）と 2 つの可能な応答コードで構成されています。

- CoA acknowledgment（ACK）[CoA-ACK]
- CoA non-acknowledgment（NAK）[CoA-NAK]

要求は CoA クライアント（通常は RADIUS またはポリシー サーバー）から発信されて、リスナーとして動作するスイッチに送信されます。

RFC 5176 規定

Disconnect Request メッセージは Packet of Disconnect（POD）とも呼ばれますが、セッション終了に対してスイッチでサポートされています。

次の表に、この機能でサポートされている IETF 属性を示します。

表 *2 :* サポートされている *IETF* 属性

属性番号	属性名
24	State
31	Calling-Station-ID
44	Acct-Session-ID
80	Message-Authenticator
101	Error-Cause

次の表に、Error-Cause 属性で取ることができる値を示します。

表 *3 : Error-Cause* の値

値	説明
201	削除された残留セッション コンテキスト
202	無効な EAP パケット（無視）
401	サポートされていない属性
402	見つからない属性
403	NAS 識別情報のミスマッチ
404	無効な要求

値	説明
405	サポートされていないサービス
406	サポートされていない拡張機能
407	無効な属性値
501	管理上の禁止
502	ルート不可能な要求（プロキシ）
503	セッション コンテキストが検出されない
504	セッション コンテキストが削除できない
505	その他のプロキシ処理エラー
506	リソースが使用不可能
507	要求が発信された
508	マルチ セッションの選択がサポートされてない

CoA 要求応答コード

CoA 要求応答コードを使用すると、スイッチにコマンドを伝達できます。

RFC 5176 で定義されている CoA 要求応答コードのパケットの形式は、コード、ID、長さ、オーセンティケータ、およびタイプ、長さ、値（TLV）形式の属性から構成されます。属性フィールドは、シスコのベンダー固有属性（VSA）を送信するために使用します。

セッションの識別

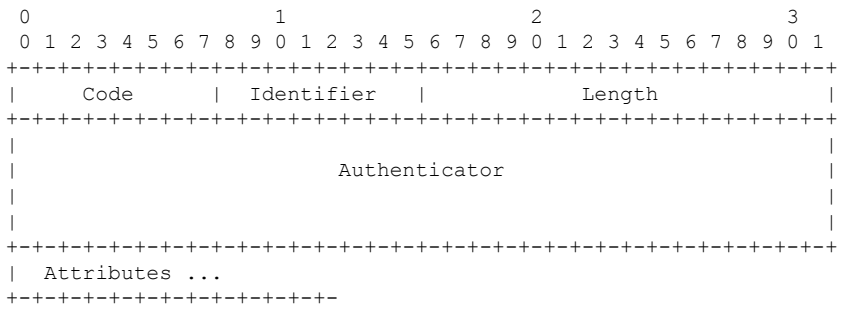
特定のセッションに向けられた切断と CoA 要求については、スイッチは 1 つ以上の次の属性に基づいて、セッションを検索します。

- Acct-Session-Id（IETF 属性 #44）
- Audit-Session-Id VSA（シスコの VSA）
- Calling-Station-Id（ホスト MAC アドレスを含む IETF 属性 #31）
- 次のいずれかの IPv6 属性。
  - Framed-IPv6-Prefix（IETF 属性 #97）および Framed-Interface-Id（IETF 属性 #96）。ともに RFC 3162 に従った完全な IPv6 アドレスを作成する
  - Framed-IPv6-Address
- プレーン IP アドレス（IETF 属性 #8）

CoA メッセージに含まれるすべてのセッション ID 属性がそのセッションと一致しない限り、スイッチは「Invalid Attribute Value」エラー コード属性を含む Disconnect-NAK または CoA-NAK を返します。

複数のセッション ID 属性がメッセージ含まれる場合は、すべての属性がセッションと一致しなければなりません。そうでない場合は、スイッチが Disconnect - negative acknowledgement（NAK）または CoA -NAK と、「Invalid Attribute Value」エラー コードを返します。

RFC 5176 で定義されている CoA 要求コードのパケットの形式は、コード、ID、長さ、オーセンティケータ、およびタイプ、長さ、値（TLV）形式の属性から構成されます。



属性フィールドは、シスコのベンダー固有属性（VSA）を送信するために使用します。

特定の適用ポリシーを対象とする CoA 要求の場合、上記のセッション ID 属性のいずれかがメッセージに含まれていると、デバイスはエラー コードが「Invalid Attribute Value」の CoA-NAK を返します。

CoA ACK 応答コード

許可ステートの変更に成功した場合は、肯定確認応答（ACK）が送信されます。CoA ACK 内で返される属性は CoA 要求によって異なり、個々の CoA コマンドで検討されます。

CoA NAK 応答コード

否定応答（NAK）は許可ステートの変更に失敗したことを示し、エラーの理由を示す属性を含めることができます。CoA が成功したかを確認するには、**show** コマンドを使用します。

CoA 要求コマンド

表 *4*：サポートされる *CoA* コマンド

コマンド [1]	シスコの VSA
Reauthenticate host	Cisco:Avpair="subscriber:command=reauthenticate"
Terminate session	これは、VSA を要求しない、標準の接続解除要求です。
Bounce host port	Cisco:Avpair="subscriber:command=bounce-host-port"

コマンド [1]	シスコの VSA
Disable host port	Cisco:Avpair="subscriber:command=disable-host-port"

[1] すべての CoA コマンドには、デバイスと CoA クライアント間のセッション ID が含まれている必要があります。

セッション再認証

不明な ID またはポスチャを持つホストがネットワークに加入して、制限されたアクセス許可プロファイル（たとえば、ゲスト VLAN）に関連付けられると、AAA サーバーは通常、セッション再認証要求を生成します。再認証要求は、クレデンシャルが不明である場合にホストが適切な認証グループに配置されることを許可します。

セッション認証を開始するために、AAA サーバーは Cisco:Avpair="subscriber:command=reauthenticate" の形式で Cisco VSA と 1 つ以上のセッション ID 属性を含む標準 CoA 要求メッセージを送信します。

現在のセッション ステートは、メッセージに対するスイッチの応答を決定します。セッションが現在、IEEE 802.1x によって認証されている場合、スイッチは EAPOL（LAN 経由の拡張認証プロトコル）RequestId メッセージをサーバーに送信することで応答します。

現在、セッションが MAC 認証バイパス（MAB）で認証されている場合は、スイッチはサーバーにアクセス要求を送信し、初期正常認証で使用されるものと同じ ID 属性を渡します。

スイッチがコマンドを受信した際にセッション認証が実行中である場合は、スイッチはプロセスを終了し、認証シーケンスを再開し、最初に試行されるように設定された方式で開始します。

セッションがまだ認証されてない、あるいはゲスト VLAN、クリティカル VLAN、または同様のポリシーで認証されている場合は、再認証メッセージがアクセス コントロール方式を再開し、最初に試行されるように設定された方式で開始します。セッションの現在の許可は、再認証によって異なる認証結果になるまで維持されます。

セッションの終了

セッションを終了させる 3 種類の CoA 要求があります。CoA 接続解除要求は、ホストポートを無効にせずにセッションを終了します。このコマンドを使用すると、指定されたホストのオーセンティケータ ステート マシンが再初期化されますが、そのホストのネットワークへのアクセスは制限されません。

ホストのネットワークへのアクセスを制限するには、Cisco:Avpair="subscriber:command=disable-host-port" VSA の設定で CoA 要求を使用します。このコマンドは、ネットワーク上で障害を引き起こしたと認識されているホストがある場合に便利であり、そのホストに対してネットワーク アクセスをただちにブロックする必要があります。ポートへのネットワークアクセスを復旧する場合は、非 RADIUS メカニズムを使用して再び有効にします。

プリンタなどのサプリカントを持たないデバイスが新しい IP アドレスを取得する必要がある場合（たとえば、VLAN変更後）は、ポートバウンスでホストポート上のセッションを終了します（ポートを一時的に無効した後、再び有効にする）。

CoA 接続解除要求

このコマンドは標準の接続解除要求です。セッションが見つからない場合、デバイスは「Session Context Not Found」エラー コード属性を使用して Disconnect-NAK メッセージを返します。セッションが見つかった場合、デバイスはセッションを終了します。セッションが完全に削除されると、デバイスは Disconnect-ACK を返します。

デバイスがクライアントに接続解除 ACK を返す前にスタンバイデバイスにフェールオーバーする場合は、クライアントから要求が再送信される際に、新しいアクティブデバイス上でそのプロセスが繰り返されます。再送信後もセッションが見つからない場合は、Disconnect-ACK と「Session Context Not Found」エラー コード属性が送信されます。

CoA 要求：ホストポートの無効化

RADIUS サーバーの CoA disable port コマンドを実行すると、セッションをホストしている認証ポートが管理的にシャットダウンされます。その結果、セッションは終了します。このコマンドは、ホストがネットワーク上で問題を起こしていることを把握し、ホストのネットワークアクセスを即座にブロックする必要がある場合に便利です。ポートのネットワークアクセスを復元するには、非RADIUSメカニズムを使用して再び有効にします。このコマンドは、次の新しいベンダー固有属性（VSA）が含まれている標準 CoA 要求メッセージで伝達されます。

Cisco:Avpair="subscriber:command=disable-host-port"

このコマンドはセッション指向であるため、「セッション ID」セクションに示されている 1 つ以上のセッション ID 属性とともに使用する必要があります。セッションが見つからない場合、デバイスは「Session Context Not Found」エラー コード属性を使用して CoA-NAK メッセージを返します。このセッションがある場合は、デバイスはホストポートを無効にし、CoA-ACK メッセージを返します。

デバイスが CoA-ACK をクライアントに返す前にデバイスに障害が発生した場合、クライアントから要求が再送信されると、新しいアクティブ デバイス上でそのプロセスが繰り返されます。デバイスが CoA-ACK メッセージをクライアントに返した後で障害が発生したが、操作が完了していない場合、その操作は新しいアクティブデバイスで再開されます。

（注） 再送信コマンドの後に接続解除要求が失敗すると、（接続解除ACKが送信されてない場合に）チェンジオーバー前にセッションが正常終了するか、または元のコマンドが実行されてスタンバイデバイスがアクティブになるまでの間に発生した他の方法（たとえば、リンク障害）によりセッションが終了することがあります。

CoA 要求：バウンス ポート

RADIUS サーバーの CoA bounce port が RADIUS サーバーから送信されると、認証ポートでリンクのフラップが発生します。その結果、このポートに接続している1つまたは複数のホストから、DHCP の再ネゴシエーションが開始されます。この状況は、VLAN の変更があり、この

認証ポートに関する変化を検出するメカニズムがないデバイス（プリンタなど）がエンドポイントの場合に発生する可能性があります。CoA bounce port は、次の新しい VSA を含む標準の CoA-Request メッセージで伝達されます。

Cisco:Avpair="subscriber:command=bounce-host-port"

このコマンドはセッション指向であるため、1 つ以上のセッション ID 属性とともに使用する必要があります。セッションが見つからない場合、デバイスは「Session Context Not Found」エラーコード属性を使用して CoA-NAK メッセージを返します。このセッションがある場合は、デバイスはホストポートを 10 秒間無効にし、再び有効にし（ポートバウンス）、CoA-ACK を返します。

デバイスが CoA-ACK をクライアントに返す前にデバイスに障害が発生した場合、クライアントから要求が再送信されると、新しいアクティブ デバイス上でそのプロセスが繰り返されます。デバイスが CoA-ACK メッセージをクライアントに返した後で障害が発生したが、操作が完了していない場合、その操作は新しいアクティブデバイスで再開されます。

RADIUS のデフォルト設定

RADIUS および AAA は、デフォルトでは無効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 を有効にすると、CLI 経由でデバイスにアクセスするユーザーを認証できます。

RADIUS サーバ ホスト

デバイスと RADIUS サーバー間の通信には、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
- 鍵文字列
- 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 サーバー上の異なる 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。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、デバイスは「%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 番目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 サーバーとデバイスは、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 サーバーデーモンが稼働するホストと、そのホストがデバイスと共有する秘密テキスト（鍵）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、および暗号鍵の値は、すべての RADIUS サーバーに対してグローバルに設定することもできますし、サーバー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ー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。

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有効にすると、デバイスはユーザーアクティビティ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ーに報告します。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文字列です。

```
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 認証中（PPP の IPCP アドレス割り当て中）には、シスコの「multiple named IP address pools」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“
```

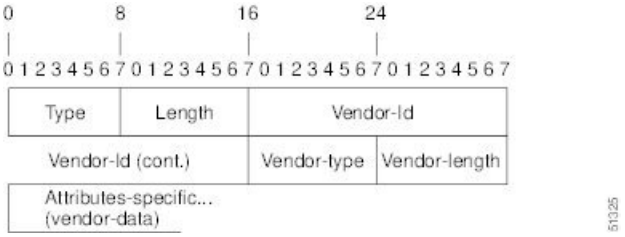
他のベンダーにも、それぞれ独自のベンダー ID、オプション、および対応する VSA があります。ベンダー ID および VSA の詳細については、RFC 2138『Remote Authentication Dial-In User Service (RADIUS)』を参照してください。

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
- 文字列（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 *2*：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 *5*：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。

フィールド	説明
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 文字列名。
説明	属性の説明。

表 *6 :* 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。

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-drop-out-of-order	正しくない順序で受信したデータパケットを破棄して、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。
26	9	1	l2tp-hidden-avp	有効にすると、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。

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-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。
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。

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	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 が有効にされているかどうかを示します。true は DSN が有効にされていることを示します。false は DSN が有効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）が有効にされているかどうかを示します。true は MDN が有効にされていることを示します。false は MDN が有効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。

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	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。
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。

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	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。
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。

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	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				

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	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。
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。

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	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）、鍵、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。

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、デバイスと RADIUS サーバー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 XE 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。

前述のように、（ベンダー固有かIETF ドラフト準拠かに関係なく）RADIUS を設定するには、RADIUS サーバーデーモンを実行するホストと、デバイスと共有する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。

RADIUS パケットの DSCP マーキング

差別化サービス（DiffServ）は、他のトラフィッククラスよりも優先的に処理するためにトラフィックを分類および管理する Quality of Service（QoS）モデルです。DiffServ は、IP パケットの 6 ビット DiffServ コードポイント（DSCP）設定を使用して、トラフィッククラスに相対的な優先順位をマークします。Cisco IOS XE ソフトウェアは、RADIUS パケットの DSCP マーキングをサポートして、RADIUS パケットの認証とアカウンティングを高速化します。

RADIUS サーバー、RADIUS サーバーグループ、およびグローバル コンフィギュレーション モードで DSCP マーキングを設定できます。DSCP マーキングが RADIUS サーバー、サーバーグループ、およびグローバル コンフィギュレーション モードに設定されると、RADIUS サーバーに入力された DSCP マーキング値が優先されます。

- RADIUS サーバーに DSCP マーキング設定がない場合、サーバーグループに設定された DSCP マーキング値が RADIUS パケットに適用されます。
- RADIUS サーバーまたは RADIUS サーバーグループに DSCP マーキング設定がない場合、グローバル コンフィギュレーション モードで設定された DSCP マーキング値が RADIUS パケットに適用されます。

RADIUS の設定

RADIUS サーバ ホストの識別

デバイスと通信するすべての RADIUS サーバー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 **key** *string* 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。

既存のサーバー ホストを認証用にグループ化するため、AAA サーバー グループを使用するようにデバイスを設定できます。

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。デバイスの IP アドレス、およびサーバーとdeviceの双方で共有する鍵文字列などの設定値です。

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

デバイス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。

（注） RADIUS および AAA サーバーは、標準のデフォルトポートでのみ実行するように設定できます。

- 1812 および 1813
- 1645 および 1646

手順の概要

1. **enable**

2. **configure terminal**
3. **radius server** *server name*
4. **address** {**ipv4 | ipv6**}*ip address*{ **auth-port** *port number* | **acct-port** *port number*}
5. **key** *string*
6. **retransmit** *value*
7. **timeout** *seconds*
8. **end**

手順の詳細

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 rsim`	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*{ **auth-port** *port number*	**acct-port** *port number*} 例： `Device(config-radius-server)# address ipv4 124.2.2.12 auth-port 1612`	（任意）RADIUS サーバー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、アカウンティング要求の 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** *string* 例： `Device(config-radius-server)# key rad123`	（任意）**key** *string* には、デバイスと RADIUS サーバーで動作する RADIUS デーモンの間で使用される認証と暗号鍵を指定します。 （注） 鍵は、RADIUS サーバーで使用する暗号鍵に一致するテキスト文字列でなければなりません。必ず **radius server** コマンドの最終項目として鍵を設定してください。先頭のスペースは無視されますが、鍵

	コマンドまたはアクション	目的
		の中間および末尾のスペースは使用されます。鍵にスペースを使用する場合は、引用符が鍵の一部分である場合を除き、引用符で鍵を囲まないでください。
ステップ 6	**retransmit** *value* 例： `Device(config-radius-server)# retransmit 10`	（任意）サーバが応答しない、または応答が遅い場合に、RADIUS 要求をリセットする回数を指定します。指定できる範囲は 1 ～ 100 です。この設定は、**radius-server retransmit** グローバル コンフィギュレーション コマンドによる設定を上書きします。
ステップ 7	**timeout** *seconds* 例： `Device(config-radius-server)# timeout 60`	（任意）deviceが要求を再送信する前にRADIUSサーバからの応答を待機する時間間隔を指定します。指定できる範囲は 1 ～ 1000 です。この設定は、**radius-server timeout** グローバル コンフィギュレーション コマンドによる設定を上書きします。
ステップ 8	**end** 例： `Device(config-radius-server)# end`	RADIUS サーバー コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。

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 コマンドを設定する必要があります。デフォルトでは、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。

手順の概要

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login** {**default** | *list-name*} *method1* [*method2...*]
5. **line** [**console** | **tty** | **vty**] *line-number* [*ending-line-number*]
6. **login authentication** {**default** | *list-name*}
7. **end**

手順の詳細

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 を有効にします。
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

	コマンドまたはアクション	目的
		RADIUS サーバー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*：ローカルユーザー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ー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、ユーザー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-line)# 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-line)# end`	ライン コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。

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。

手順の概要

1. **enable**
2. **configure terminal**
3. **radius server** *name*
4. **address** {**ipv4** | **ipv6**} {*ip-address* | *hostname*} **auth-port** *port-number* **acct-port** *port-number*
5. **key** *string*
6. **end**

手順の詳細

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。

	コマンドまたはアクション	目的
ステップ 5	**key** *string* 例： `Device(config-radius-server)# key cisco123`	デバイスとRADIUSサーバーとの間におけるすべての RADIUS 通信用の認証および暗号鍵を指定します。
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。

ユーザー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。

手順の概要

1. **enable**
2. **configure terminal**
3. **aaa authorization network** *authorization-list***radius**
4. **aaa authorization exec** *authorization-list***radius**
5. **end**

手順の詳細

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	**aaa authorization network** *authorization-list***radius** 例： `Device(config)# aaa authorization network list1 radius`	ネットワーク関連のすべてのサービス要求に対して、ユーザーが RADIUS 許可を受けるように device を設定します。
ステップ 4	**aaa authorization exec** *authorization-list***radius** 例： `Device(config)# aaa authorization exec list1 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、ユーザープロファイル情報（**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`	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

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。

手順の概要

1. **enable**
2. **configure terminal**
3. **aaa accounting network** *accounting-list***start-stop radius**
4. **aaa accounting exec** *accounting-list***start-stop radius**
5. **end**

手順の詳細

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** *accounting-list***start-stop radius** 例： `Device(config)# aaa accounting network accounting-list 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有効にします。
ステップ 4	**aaa accounting exec** *accounting-list***start-stop radius** 例： `Device(config)# aaa accounting exec acc-list start-stop radius`	RADIUSアカウンティングを有効にして、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`	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。

すべての RADIUS サーバの設定

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

手順の概要

1. **enable**
2. **configure terminal**
3. **radius server** *server name*
4. **key** *string*
5. **retransmit** *retries*
6. **timeout** *seconds*
7. **end**

手順の詳細

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 rsim`	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	**key** *string* 例： `Device(config-radius-server)# key your_server_key`	スイッチとすべての RADIUS サーバ間で共有される秘密テキスト文字列を指定します。 （注） 鍵は、RADIUS サーバーで使用する暗号鍵に一致するテキスト文字列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、鍵の中間および末尾のスペースは使用されます。鍵にスペースを使用する場合は、引用符が鍵の一部分である場合を除き、引用符で鍵を囲まないでください。
ステップ 5	**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 6	**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 7	**end** 例： `Device(config-radius-server)# end`	RADIUS サーバー コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。

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 属性を設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **radius-server vsa send** [**accounting** | **authentication**]
4. **end**

手順の詳細

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 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`	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。

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 サーバー通信を設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **radius server** *server name*
4. **address** { **ipv4 | ipv6** } *ip address*
5. **non-standard**
6. **key** *string*
7. **end**

手順の詳細

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 rsim`	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 172.24.25.10`	（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。

	コマンドまたはアクション	目的
ステップ 6	**key** *string* 例： `Device(config-radius-server)# key rad123`	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テキスト文字列を指定します。デバイスと RADIUS サーバーはこのテキスト文字列を使用してパスワードを暗号化し、応答を交換します。
ステップ 7	**end** 例： `Device(config-radius-server)# end`	RADIUS サーバーモードを終了し、特権 EXEC モードを開始します。

RADIUS サーバーでの DSCP マーキングの設定

RADIUS サーバーでの認証とアカウンティング用の DSCP マーキングを設定するには、次の手順に従います。

手順の概要

1. **enable**
2. **configure terminal**
3. **radius server** *server_name*
4. **address** { **ipv4 | ipv6** } *ip address* [**auth-port** *auth_port_number* **acct-port** *acct_port_number*]
5. **dscp** {**acct** *dscp_acct_value* | **auth** *dscp_auth_value*}
6. **key** *string*
7. **end**

手順の詳細

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。
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	**radius server** *server_name* 例： `Device(config)# radius server rsim`	RADIUS サーバー設定の名前を Protected Access Credential（PAC）のプロビジョニング用に指定し、RADIUS サーバー設定モードを開始します。
ステップ 4	**address** { **ipv4** | **ipv6** } *ip address* [**auth-port** *auth_port_number* **acct-port** *acct_port_number*] 例： `Device(config-radius-server)# address ipv4 10.1.1.1 auth-port 1645 acct-port 1646`	（任意）RADIUS サーバーの IP アドレスを指定します。 • **auth-port** は、RADIUS 認証サーバーのポート値を設定します。デフォルト値は 1812 です。 • **acct-port** は、RADIUS アカウンティングサーバーのポート値を設定します。デフォルト値は 1813 です。
ステップ 5	**dscp** {**acct** *dscp_acct_value* | **auth** *dscp_auth_value*} 例： `Device(config-radius-server)# dscp auth 10 acct 20`	RADIUS サーバーでの認証とアカウンティング用の DSCP マーキングを設定します。 • **acct** はアカウンティングの RADIUS DSCP マーキング値を設定します。有効な範囲は 1 ～ 63 です。デフォルト値は 0 です • **auth** は認証の RADIUS DSCP マーキング値を設定します。有効な範囲は 1 ～ 63 です。デフォルト値は 0 です
ステップ 6	**key** *string* 例： `Device(config-radius-server)# key rad123`	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テキスト文字列を指定します。デバイスと RADIUS サーバーはこのテキスト文字列を使用してパスワードを暗号化し、応答を交換します。
ステップ 7	**end** 例： `Device(config-radius-server)# end`	RADIUS サーバーモードを終了し、特権 EXEC モードを開始します。

RADIUS サーバーグループでの送信元インターフェイスと DSCP マーキングの設定

次の手順に従って、RADIUS サーバーグループでの認証とアカウンティング用の送信元インターフェイスと DSCP マーキングを設定します。

手順の概要

1. **enable**
2. **configure terminal**
3. **aaa group server radius** *group_name*
4. **server name** *name*
5. {**ip** | **ipv6**} **radius source-interface** *type number*
6. **dscp** {**acct** *dscp_acct_value* | **auth** *dscp_auth_value*}
7. **end**

手順の詳細

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。
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 group server radius** *group_name* 例： `Device(config)# aaa group server radius abc`	RADIUS サーバー グループ コンフィギュレーションを定義し、RADIUS サーバー グループ コンフィギュレーション モードを開始します。
ステップ 4	**server name** *name* 例： `Device(config-sg-radius)# server name serv1`	RADIUS サーバーをサーバーグループに関連付けます。
ステップ 5	{**ip**	**ipv6**} **radius source-interface** *type number* 例： `Device(config-sg-radius)# ipv6 radius source-interface ethernet 0/0`	RADIUS サーバーの送信元アドレスに使用するインターフェイスを指定します。
ステップ 6	**dscp** {**acct** *dscp_acct_value*	**auth** *dscp_auth_value*} 例：	RADIUS サーバーグループでの認証とアカウンティング用の DSCP マーキングを設定します。

	コマンドまたはアクション	目的
	`Device(config-sg-radius)# dscp auth 10 acct 20`	• **acct** はアカウンティングの RADIUS DSCP マーキング値を設定します。有効な範囲は 1 ～ 63 です。デフォルト値は 0 です • **auth** は認証の RADIUS DSCP マーキング値を設定します。有効な範囲は 1 ～ 63 です。デフォルト値は 0 です
ステップ 7	**end** 例： `Device(config-radius-server)# end`	RADIUS サーバーモードを終了し、特権 EXEC モードを開始します。

デバイス上での CoA の設定

CoA をdeviceで設定するには、次の手順を実行します。この手順は必須です。

手順の概要

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa server radius dynamic-author**
5. **client** {*ip-address* | *name*} [**vrf** *vrfname*] [**server-key** *string*]
6. **server-key** [**0** | **7**] *string*
7. **port** *port-number*
8. **auth-type** {**any** | **all** | **session-key**}
9. **ignore server-key**
10. **exit**
11. **authentication command bounce-port ignore**
12. **authentication command disable-port ignore**
13. **end**

手順の詳細

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`	AAA を有効にします。
ステップ 4	**aaa server radius dynamic-author** 例： `Device(config)# aaa server radius dynamic-author`	デバイスを認証、許可、アカウンティング（AAA）サーバーとして設定して外部ポリシーサーバーとの通信を容易にし、ダイナミック許可ローカル 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 5	**client** {*ip-address* | *name*} [**vrf** *vrfname*] [**server-key** *string*] 例： `Device(config-locsvr-da-radius)# client client1 vrf vrf1`	デバイスが CoA を受け取り、要求を取り外す RADIUS クライアントを指定します。
ステップ 6	**server-key** [**0** | **7**] *string* 例： `Device(config-locsvr-da-radius)# server-key your_server_key`	RADIUS 鍵をデバイスと RADIUS クライアントとの間で共有されるように設定します。
ステップ 7	**port** *port-number* 例： `Device(config-locsvr-da-radius)# port 25`	設定された RADIUS クライアントから RADIUS 要求をデバイスが受信するポートを指定します。
ステップ 8	**auth-type** {**any** | **all** | **session-key**} 例： `Device(config-locsvr-da-radius)# auth-type any`	deviceが RADIUS クライアントに使用する許可のタイプを指定します。 クライアントは、許可用に設定されたすべての属性と一致していなければなりません。
ステップ 9	**ignore server-key** 例： `Device(config-locsvr-da-radius)# ignore`	（任意）server-key を無視するように device を設定します。

	コマンドまたはアクション	目的
	server-key	
ステップ 10	**exit** 例： `Device(config-locsvr-da-radius)# exit`	ダイナミック認可ローカル 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。
ステップ 11	**authentication command bounce-port ignore** 例： `Device(config)# authentication command bounce-port ignore`	（任意）CoA 要求を無視して、セッションをホスティングするポートを一時的に無効にするようにdeviceを設定します。ポートを一時的に無効にする目的は、VLAN の変更が発生しても、その変更を検出するサプリカントがエンドポイント上にない場合に、ホストから DHCP 再ネゴシエーションを行わせることです。
ステップ 12	**authentication command disable-port ignore** 例： `Device(config)# authentication command disable-port ignore`	（任意）セッションをホスティングしているポートを管理上のシャットダウン状態にするよう要求する非標準コマンドを無視するようにdeviceを設定します。ポートをシャットダウンすると、セッションが終了します。 ポートを再び有効にするには、標準の CLI または SNMP コマンドを使用します。
ステップ 13	**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。

CoA 機能のモニタリング

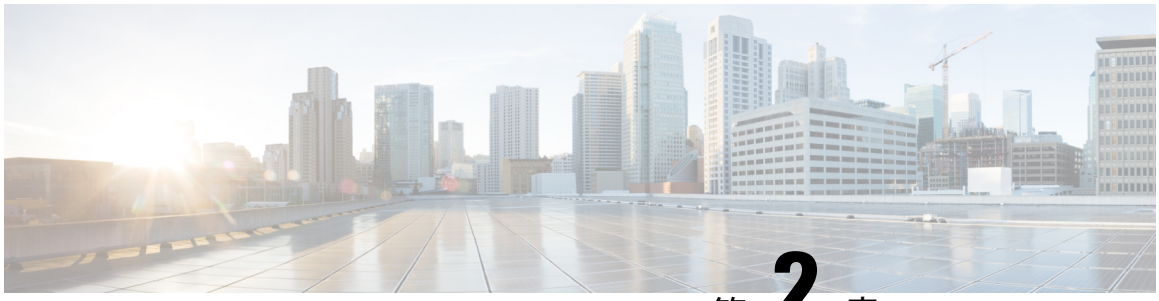
表 *7:* 特権 *EXEC* 表示コマンド

コマンド	目的
show aaa attributes protocol radius	RADIUS コマンドの AAA 属性を表示します。

表 *8:* グローバル トラブルシューティング コマンド

コマンド	目的
debug radius	RADIUS のトラブルシューティングを行うための情報を表示します。

コマンド	目的
debug aaa coa	CoA 処理のトラブルシューティングを行うための情報を表示します。
debug aaa pod	POD パケットのトラブルシューティングを行うための情報を表示します。
debug aaa subsys	POD パケットのトラブルシューティングを行うための情報を表示します。



第 2 章

MACsec の暗号化

MACsec の暗号化

MACsec は 2 台の MACsec 対応デバイス間のパケットの認証と暗号化の IEEE 802.1AE 規格です。Catalyst スイッチは、スイッチとホストデバイス間の暗号化に、スイッチからホストへのリンクでの MACsec Key Agreement（MKA）による 802.1AE 暗号化をサポートします。また、スイッチは、MKA ベースの鍵交換プロトコルを使用して、スイッチ間（ネットワーク間デバイス）セキュリティの MACsec 暗号化をサポートします。

（注） スイッチ間 MACSec が有効な場合、EAP-over-LAN（EAPOL）パケットを除くすべてのトラフィックが暗号化されます。

リンク層セキュリティはスイッチ間のパケット認証とスイッチ間のMACsec暗号化の両方を含みます（暗号化は任意です）。

表 *9*：スイッチ ポートの *MACsec* サポート

接続	MACsec のサポート
スイッチからホストへ	MACsec MKA の暗号化

スイッチからスイッチへ	MACsec MKA の暗号化

Cisco TrustSec はスイッチ間のリンクにのみ使用され、PC や IP フォンなどのエンドホストに接続されたスイッチポートではサポートされません。MKA は、スイッチからホストへのリンクとスイッチ間リンクでサポートされます。ホスト側のリンクは、IEEE 802.1x の有無にかかわらず異種デバイスを扱うために、一般に柔軟な認証順序を使用し、オプションで MKA ベースの MACsec 暗号化を使用できます。ネットワーク エッジ アクセス トポロジ（NEAT）はコンパクトなスイッチがワイヤリングクローゼットの外側にセキュリティを拡張するために使用されます。

MACsec Key Agreement

802.1AE で定義された MACsec では、暗号化キー入力のためにアウトオブバンド方式を使用することによって、有線ネットワーク上で MAC レイヤの暗号化を実現します。MACsec Key Agreement（MKA）プロトコルでは、必要なセッション キーを提供し、必要な暗号化キーを管理します。MKA と MACsec は、証明書ベース MACsec または事前共有キー（PSK）フレームワークを使用した認証に成功した後に実装されます。

MACsec を使用するスイッチでは、MKA ピアに関連付けられたポリシーに応じて、MACsec フレームまたは非 MACsec フレームを許可します。MACsec フレームは暗号化され、整合性チェック値（ICV）で保護されます。スイッチは MKA ピアからフレームを受信すると、MKA によって提供されたセッション キーを使用してこれらのフレームを暗号化し、正しい ICV を計算します。スイッチはこの ICV をフレーム内の ICV と比較します。一致しない場合は、フレームが破棄されます。また、スイッチは現在のセッション キーを使用して、ICV を暗号化し、セキュアなポート（セキュアな MAC サービスを MKA ピアに提供するために使用されるアクセス ポイント）を介して送信されたフレームに追加します。

MKA プロトコルは、基礎となる MACsec プロトコルで使用される暗号キーを管理します。MKA の基本的な要件は 802.1x-REV で定義されています。MKA プロトコルでは 802.1x を拡張し、相互認証の確認によってピアを検出し、MACsec 秘密キーを共有してピアで交換されるデータを保護できます。

EAP フレームワークでは、新しく定義された EAP-over-LAN（EAPOL）パケットとして MKA を実装します。EAP 認証では、データ交換で両方のパートナーで共有されるマスターセッション キー（MSK）を生成します。EAP セッション ID を入力すると、セキュアな接続アソシエーション キー名（CKN）が生成されます。スイッチは、アップリンクおよびダウンリンクの両方のオーセンティケータとして機能します。また、ダウンリンクのキー サーバーとして機能します。これによってランダムなセキュア アソシエーション キー（SAK）が生成され、クライアント パートナーに送信されます。クライアントはキー サーバーではなく、単一の MKA エンティティであるキー サーバーとだけ対話できます。キーの派生と生成の後で、スイッチは定期的にトランスポートをパートナーに送信します。デフォルトの間隔は 2 秒間です。

EAPOL プロトコル データ ユニット（PDU）のパケット本体は、MACsec Key Agreement PDU（MKPDU）と呼ばれます。MKA セッションと参加者は、MKA ライフタイム（6 秒間）が経過しても参加者から MKPDU を受信していない場合に削除されます。たとえば、MKA ピアが

接続を解除した場合、スイッチ上の参加者は MKA ピアから最後の MKPDU を受信した後、6 秒間が経過するまで MKA の動作を継続します。

（注） MKPDU の整合性チェック値（ICV）インジケータはオプションです。トラフィックが暗号化されている場合、ICV はオプションではありません。

EAPoL 通知は、キー関連情報のタイプの使用を示します。通知は、サプリカントとオーセンティケータの機能を通知するために使用できます。各側の機能に基づいて、キー関連情報の最大公分母を使用できます。

MKA ポリシー

インターフェイスで MKA を有効にするには、定義された MKA ポリシーをインターフェイスに適用する必要があります。次のオプションを設定可能です。

- 16 ASCII 文字未満のポリシー名。
- 物理インターフェイスごとの 0 バイト、30 バイト、または 50 バイトの機密保持（暗号化）オフセット。

ポリシーマップアクションの定義

ここでは、ポリシーマップアクションとその定義について説明します。

- Activate：サービステンプレートをセッションに適用します。
- Authenticate：セッションの認証を開始します。
- Authorize：セッションを明示的に許可します。
- Set-domain：クライアントのドメインを明示的に設定します。
- Terminate：実行中のメソッドを終了し、セッションに関連付けられているすべてのメソッドの詳細を削除します。
- Deactivate：セッションに適用されたサービステンプレートを削除します。適用されない場合、アクションは実行されません。
- Set-timer：タイマーを開始し、セッションに関連付けます。タイマーが期限切れになると、開始する必要があるアクションを処理できます。
- Authentication-restart：認証を再開します。
- Clear-session：セッションを削除します。
- Pause：認証を一時停止します。

残りのアクションについては説明の必要はなく、認証に関連したものです。

仮想ポート

仮想ポートは、1 つの物理ポート上の複数のセキュアな接続アソシエーションに使用します。各接続アソシエーション（ペア）は仮想ポートを表します。アップリンクでは、物理ポートごとに1つの仮想ポートのみを指定できます。同じポートで同じVLAN内のセキュアなセッションとセキュアでないセッションを同時にホストすることはできません。この制限のため、802.1x マルチ認証モードはサポートされません。

この制限の例外は、マルチホスト モードで最初の MACsec サプリカントが正常に認証され、スイッチに接続されたハブに接続される場合です。ハブに接続された非MACsecホストでは、マルチホストモードであるため、認証なしでトラフィックを送信できます。最初にクライアントが成功した後、他のクライアントでは認証が必要ないため、マルチホストモードの使用は推奨しません。

仮想ポートは、接続アソシエーションの任意のIDを表し、MKAプロトコル外では意味を持ちません。仮想ポートは個々の論理ポート ID に対応します。仮想ポートの有効なポート ID は 0x0002 ～ 0xFFFF です。各仮想ポートは、16 ビットのポート ID に連結された物理インターフェイスの MAC アドレスに基づいて、一意のセキュア チャネル ID（SCI）を受け取ります。

MKA 統計情報

一部の MKA カウンタはグローバルに集約され、その他のカウンタはグローバルとセッション単位の両方で更新されます。また、MKA セッションのステータスに関する情報も取得できます。詳細については、MKA 統計情報の表示を参照してください。

キー ライフタイムおよびヒットレス キー ロールオーバー

MACsec キー チェーンには、キー ID とオプションのライフタイムが設定された複数の事前共有キー（PSK）を含めることができます。キーのライフタイムには、キーが期限切れになる時刻が指定されます。ライフタイム設定が存在しない場合は、無期限のデフォルトライフタイムが使用されます。ライフタイムが設定されている場合、ライフタイムの期限が切れた後に、MKA はキー チェーン内の次に設定された事前共有キーにロールオーバーします。キーのタイムゾーンは、ローカルまたはUTCを指定できます。デフォルトのタイムゾーンはUTCです。

キーチェーン内に2番目のキーを設定し、最初のキーのライフタイムを設定することで、同じキーチェーン内の次のキーにロールオーバーできます。最初のキーのライフタイムが期限切れになると、リスト内の次のキーに自動的にロールオーバーします。同一のキーがリンクの両側で同時に設定されている場合、キーのロールオーバーはヒットレスになります。つまり、キーはトラフィックを中断せずにロールオーバーされます。

すべての参加デバイスで、MACsec キーチェーンを Network Time Protocol（NTP）を使用して同期し、同じタイムゾーンを使用する必要があります。参加しているすべてのデバイスが同期されていない場合、接続アソシエーションキー（CAK）のキー再生成はすべてのデバイスで同時に開始されません。

（注） キーのライフタイムは、ヒットレス キー ロールオーバーを実現するためにオーバーラップする必要があります。

リプレイ保護ウィンドウ サイズ

リプレイ保護は、リプレイ攻撃に対抗するためにMACsecにより提供される機能です。暗号化された各パケットには一意のシーケンス番号が割り当てられ、シーケンスはリモートエンドで確認されます。メトロ イーサネット サービス プロバイダー ネットワークを介して送信されるフレームは、順序が変更されることが多くあります。これは、ネットワーク内で使用されている優先順位付けとロードバランシング のメカニズムによるものです。

フレームの順序が変更されるプロバイダーネットワーク上でMACsecの使用をサポートするには、リプレイウィンドウが必要です。ウィンドウ内のフレームは順不同で受信できますが、リプレイ保護されません。デフォルトのウィンドウサイズは0で、厳密な受信順序が適用されます。リプレイウィンドウのサイズは、0 ー 2^{32}- 1 の範囲で設定できます。

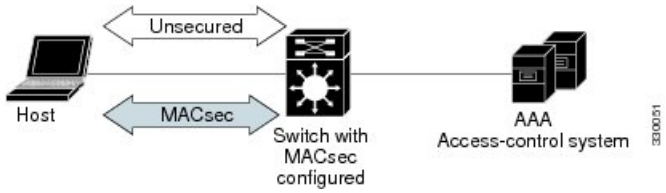
MACsec、MKA、および 802.1x ホスト モード

MACsec と MKA プロトコルは、802.1x シングルホスト モード、マルチホスト モード、またはマルチドメイン認証（MDA）モードで使用できます。マルチ認証モードはサポートされません。

シングルホスト モード

次の図に、MKA を使用して、MACsec で 1 つの EAP 認証済みセッションをセキュアにする方法を示します。

図 *3:* セキュアなデータ セッションでのシングルホスト モードの *MACsec*

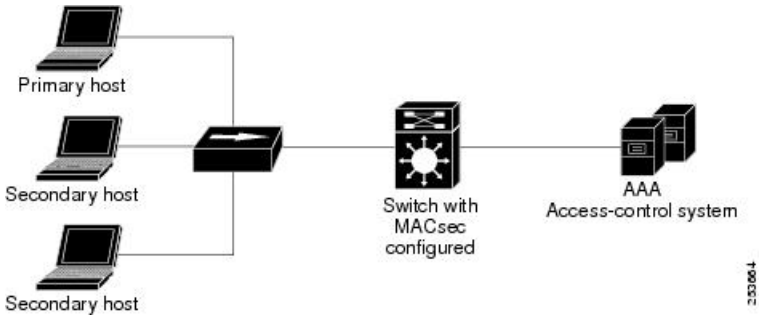


マルチホスト モード

標準の（802.1x REV ではない）802.1x マルチホスト モードでは、1 つの認証に基づいてポートが開いているか、閉じられています。1 人のユーザー（プライマリ セキュア クライアント サービスのクライアント ホスト）が認証される場合は、同じポートに接続されているホストに同じレベルのネットワーク アクセスが提供されます。セカンダリ ホストが MACsec サプリカントの場合、認証できず、トラフィック フローは発生しません。非 MACsec ホストであるセカンダリ ホストは、マルチホスト モードであるため、認証なしでネットワークにトラフィックを

送信できます。次の図に、標準のマルチホスト非セキュア モードにおける MACsec を示します。

図 *4:* マルチホスト モードの *MACsec*：非セキュア



（注） マルチホストモードは推奨されていません。これは最初にクライアントが成功した後、他のクライアントでは認証が必要ないことから、安全性が低いためです。

標準の（802.1x REV ではない）802.1x マルチドメイン モードでは、1 つの認証に基づいてポートが開いているか、閉じられています。プライマリ ユーザー（データ ドメインの PC）が認証されると、同じレベルのネットワーク アクセスが同じポートに接続されているドメインに提供されます。セカンダリ ユーザーが MACsec サプリカントの場合、認証できず、トラフィック フローは発生しません。非 MACsec ホストであるセカンダリ ユーザー（音声ドメインの IP フォン）は、マルチドメイン モードであるため、認証なしでネットワークにトラフィックを送信できます。

マルチドメインモード

標準の（802.1x REV ではない）802.1x マルチドメイン モードでは、1 つの認証に基づいてポートが開いているか、閉じられています。プライマリ ユーザー（データ ドメインの PC）が認証されると、同じレベルのネットワーク アクセスが同じポートに接続されているドメインに提供されます。セカンダリ ユーザーが MACsec サプリカントの場合、認証できず、トラフィック フローは発生しません。非 MACsec ホストであるセカンダリ ユーザー（音声ドメインの IP フォン）は、マルチドメイン モードであるため、認証なしでネットワークにトラフィックを送信できます。

証明書ベースの MACsec を使用した MACsec MKA

MACsec MKA はスイッチ間リンクでサポートされます。証明書ベースの MACsec を使用して、デバイスのアップリンクポート間で MACsec MKA を設定できます。証明書ベースの MACsec は相互認証を許可し、MSK（マスターセッションキー）を取得します。そのキーから、MKA 操作用の接続アソシエーションキー（CAK）が取得されます。デバイスの証明書は、AAA サーバーへの認証用に、証明書ベースの MACsec を使用して伝送されます。

（注） 証明書ベースの MACsec は、Cisco IOS XE 17.13.1 リリース以降の Cisco Catalyst ESS9300 エンベデッド シリーズ スイッチでサポートされています。

証明書ベースの MACsec を使用する MACsec MKA の前提条件

- 認証局（CA）サーバーがネットワークに設定されていることを確認します。
- CA 証明書を生成します。
- Cisco Identity Services Engine（ISE）リリース 2.0 が設定されていることを確認します。
- 両方の参加デバイス（CA サーバーと Cisco Identity Services Engine（ISE））が Network Time Protocol（NTP）を使用して同期されていることを確認します。時間がすべてのデバイスで同期されていないと、証明書は検証されません。
- 802.1x 認証と AAA がデバイスに設定されていることを確認します。

スイッチ間 MKA MACsec マストセキュアポリシー

入力と出力の両方で must-secure のサポートが有効になります。MKA では、must-secure がサポートされています。must-secure を有効にすると、EAPoL トラフィックのみが暗号化されません。他のトラフィックは暗号化されます。暗号化されないパケットはドロップされます。

（注） デフォルトでは、Must-secure モードが有効になっています。

ポートチャネルの MKA/MACsec

MKA/MACsec は、ポートチャネルのポートメンバで設定できます。ポートチャネルのポートメンバ間でMKAセッションが確立されるため、MKA/MACsecはポートチャネルに依存しません。

（注） ポートチャネルは PSK ベースの MACsec ではサポートされますが、証明書ベースの MACsec ではサポートされません。

（注） ポートチャネルの一部として形成される EtherChannel リンクは、同様であってもなくても構いません。つまり、リンクは MACsec セキュアまたは非 MACsec セキュアのいずれかになれます。ポートチャネルの一方のポートメンバが MACsec に設定されていない場合でも、ポートメンバ間の MKA セッションが確立されます。

ポートチャネルのセキュリティを強化するために、すべてのメンバーポートで MKA/MACsec を有効にすることをお勧めします。

MACsec 暗号アナウンスメント

暗号アナウンスメントを使用すると、サプリカントとオーセンティケータは、それぞれの MACsec 暗号スイート機能を相互にアナウンスできます。サプリカントとオーセンティケータの両方が、サポートされる最大の共通 MACsec 暗号スイートを計算し、MKA セッションの鍵情報と同じものを使用します。

（注） MKA ポリシーで設定されている MACsec 暗号スイート機能だけが、オーセンティケータからサプリカントにアナウンスされます。

EAPoL アナウンスメントには 2 つのタイプがあります。

- 非セキュアアナウンスメント（EAPoL PDU）：非セキュアアナウンスメントは、MACsec 暗号スイート機能を非セキュアな方法で伝送する EAPoL アナウンスメントです。これらのアナウンスメントは、認証の前に MKA セッションに使用する鍵の幅を決定するために使用されます。
- セキュアアナウンスメント（MKPDU）：セキュアアナウンスメントは、以前は非セキュアアナウンスメントで共有されていた MACsec 暗号スイート機能を再検証します。

セッションが認証されると、EAPoL アナウンスメントを介して受信されたピア機能がセキュアアナウンスメントで再検証されます。機能に不一致がある場合、MKA セッションは切断されます。

MACsec 暗号アナウンスメントに関する制約事項

- MACsec 暗号アナウンスメントは、スイッチからホストへのリンクでのみサポートされます。
- サプリカントとオーセンティケータ間の MKA セッションは、両方に設定された MACsec 暗号スイート機能が共通の暗号スイートにならない場合でも切断されません。
- ホストとスイッチ間の MKA MACsec 256 ビット暗号は、Network Advantage と Network Essentials の両方でサポートされていません。

MACsec 暗号化の設定方法

MACsec 暗号化の前提条件

MACsec 暗号化の前提条件

- シスコ以外および IOS XE 以外のデバイスとの相互運用性を可能にするために、デバイスで MACsec 暗号化を設定するときに **ssci-based-on-sci** コマンドを有効にします。
- 802.1x 認証と AAA がデバイスに設定されていることを確認します。

証明書ベース MACsec の前提条件

- 認証局（CA）サーバーがネットワークに設定されていることを確認します。
- CA 証明書を生成します。
- Cisco Identity Services Engine（ISE）リリース 2.0 が設定されていることを確認します。
- 両方の参加デバイス（CA サーバーと Cisco Identity Services Engine（ISE））が Network Time Protocol（NTP）を使用して同期されていることを確認します。時間がすべてのデバイスで同期されていないと、証明書は検証されません。

MACsec 暗号化の制約事項

- MACsec Key Agreement（MKA）は、ステートフルまたはステートレス両方のハイアベイラビリティではサポートされません。
- MKA を使用した MACsec は、ポイントツーポイントリンクでのみサポートされます。
- MACsec 設定は、EtherChannel ポートではサポートされません。代わりに、EtherChannel の個々のメンバポートに MACsec 設定を適用できます。MACsec 設定を削除するには、最初に EtherChannel からメンバポートをバンドル解除してから、個々のメンバポートから削除する必要があります。
- Cisco Catalyst IE9300 高耐久性シリーズ スイッチは、Network Essentials ライセンスで 128 ビット MACsec 暗号化、Network Advantage ライセンスで 256 ビット MACsec 暗号化をサポートしています。
- 証明書ベースの MACsec は、アクセスセッションがクローズドとして、またはマルチホストモードで設定されている場合にのみサポートされます。他のコンフィギュレーションモードはサポートされません。
- パケット番号枯渇キー再生成はサポートされません。

- **dot1q tag vlan native** コマンドがグローバルレベルで設定されている場合、トランクポートでの dot1x 再認証は失敗します。
- Precision Time Protocol（PTP）を備えた MACsec はサポートされません。
- **should-secure** アクセスモードは、PSK 認証を使用するスイッチ間ポートでのみサポートされます。
- PSK フォールバックキーチェーンは、ポイントツーマルチポイントではサポートされません。
- PSK フォールバックキーチェーンは、高可用性設定ではサポートされません。
- PSK フォールバックキーチェーンは、1 つのキーのみで無期限にサポートします。
- フォールバックキーチェーンで使用される接続アソシエーションキー名（CKN）の ID は、プライマリキーチェーンで使用される CKN ID のいずれとも一致しないようにしてください。
- 次の制限は、証明書ベースの MACsec にのみ適用されます。
  - ポートは、アクセスモードまたはトランクモードである必要があります。
  - MKA は、ポートチャネルではサポートされません。
  - no switch port の設定されたポートはサポートされません。

MACsec 暗号化の推奨事項

ここでは、MACsec 暗号化の設定に関する推奨事項を示します。

- スイッチとホスト間の接続では、機密性（暗号化）オフセットを 0 として使用します。
- アクティブセッションの MKA ポリシーまたは MACsec 設定を変更した後、ポートで **shutdown** コマンドを実行し、**no shutdown** コマンドを実行して、変更がアクティブセッションに適用されるようにします。
- 接続アソシエーションキー（CAK）キー再生成オーバーラップタイマーを 30 秒以上に設定します。

MKA および MACsec の設定

デフォルトでは、MACsec は無効です。MKA ポリシーは設定されていません。

MKA ポリシーの設定

MKA プロトコルポリシーを作成するには、特権 EXEC モードで次の手順を実行します。MKA では 802.1x をイネーブルにすることも必要であることに注意してください。

手順の概要

1. **enable**
2. **configure terminal**
3. **mka policy** *policy-name*
4. **key-server** *priority*
5. **include-icv-indicator**
6. **macsec-cipher-suite** {*gcm-aes-128* | *gcm-aes-256*}
7. **confidentiality-offset** *offset-value*
8. **ssci-based-on-sci**
9. **end**
10. **show mka policy**

手順の詳細

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	**mka policy** *policy-name* 例： `Device(config)# mka policy mka_policy`	MKA ポリシーを指定して、MKA ポリシー コンフィギュレーション モードを開始します。ポリシー名の長さは最大で 16 文字です。 （注） MKA ポリシー内のデフォルトの MACsec 暗号スイートは常に「GCM-AES-128」です。デバイスが「GCM-AES-128」および「GCM-AES-256」の両方の暗号方式をサポートしている場合は、ユーザー定義の MKA ポリシーを定義して使用し、必要に応じて、128 および 256 ビット両方の暗号を含めるか、または 256 ビットのみの暗号を含めることを強くお勧めします。
ステップ 4	**key-server** *priority* 例： `Device(config-mka-policy)# key-server priority 200`	MKA キーサーバオプションを設定し、優先順位を設定します（0 ～ 255 の値）。 （注） 鍵サーバープライオリティの値を 255 に設定した場合、ピアは鍵サーバーになることはできません。

	コマンドまたはアクション	目的
		鍵サーバーの優先順位の値は MKA PSK に対してのみ有効です。MKA EAPTLS に対しては有効ではありません。
ステップ 5	**include-icv-indicator** 例： `Device(config-mka-policy)# include-icv-indicator`	MKPDU の ICV インジケータを有効にします。ICV インジケータを無効にするには、このコマンドの **no** 形式を使用します。
ステップ 6	**macsec-cipher-suite** {*gcm-aes-128* | *gcm-aes-256*} 例： `Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128`	128 ビットまたは 256 ビット暗号化により SAK を取得するための暗号スイートを設定します。
ステップ 7	**confidentiality-offset** *offset-value* 例： `Device(config-mka-policy)# confidentiality-offset 0`	各物理インターフェイスに機密性（暗号化）オフセットを設定します。 （注） オフセット値は、0、30、または 50 を指定できます。クライアントで Anyconnect を使用している場合は、オフセット 0 を使用することをお勧めします。
ステップ 8	**ssci-based-on-sci** 例： `Device(config-mka-policy)# ssci-based-on-sci`	（任意）Secure Channel Identifier（SCI）値に基づいて Short Secure Channel Identifier（SSCI）値を計算します。SCI 値が高いほど、SSCI 値は低くなります。
ステップ 9	**end** 例： `Device(config-mka-policy)# end`	MKA ポリシー 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。
ステップ 10	**show mka policy** 例： `Device# show mka policy`	MKA ポリシー設定情報を表示します。

スイッチからホストへの MACsec の暗号化設定

音声用に 1 つの MACsec セッションとデータ用に 1 つの MACsec セッションが存在するインターフェイスで MACsec を設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configureterminal**
3. **interface** *type number*

4. **switchport access vlan***vlan-id*
5. **switchport mode access**
6. **macsec**
7. **authentication event linksec fail action authorize vlan** *vlan-id*
8. **authentication host-mode multi-domain**
9. **authentication linksec policy must-secure**
10. **authentication port-control auto**
11. **authentication periodic**
12. **authentication timer reauthenticate**
13. **authentication violation protect**
14. **mka policy** *policy-name*
15. **dot1x pae authenticator**
16. **spanning-tree portfast**
17. **end**
18. **show authentication session interface** *interface-id*
19. **show mka sessions**

手順の詳細

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	**configureterminal** 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`	MACsec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。インターフェイスは物理インターフェイスでなければなりません。
ステップ 4	**switchport access vlan***vlan-id* 例： `Device(config-if)# switchport access vlan 1`	このポートのアクセス VLAN を設定します。
ステップ 5	**switchport mode access** 例： `Device(config-if)# switchport mode access`	インターフェイスをアクセス 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 6	**macsec** 例： `Device(config-if)# macsec`	インターフェイス上で 802.1ae MACsec を有効にします。macsec コマンドを使用すると、スイッチからホストへのリンクでのみ MKA MACsec が有効になります。
ステップ 7	**authentication event linksec fail action authorize vlan** *vlan-id* 例： `Device(config-if)# authentication event linksec fail action authorize vlan 1`	（任意）認証の試行に失敗した後で、ポート上の制限付き VLAN を許可することによって、ユーザー証明書が認識されない認証リンク セキュリティの問題をスイッチが処理することを指定します。
ステップ 8	**authentication host-mode multi-domain** 例： `Device(config-if)# authentication host-mode multi-domain`	ホストと音声デバイスの両方が、802.1x で許可されたポート上で認証されるように、ポート上の認証マネージャ モードを設定します。設定されていない場合、デフォルトのホストモードはシングルです。
ステップ 9	**authentication linksec policy must-secure** 例： `Device(config-if)# authentication linksec policy must-secure`	LinkSec セキュリティ ポリシーを設定して、ピアを利用できる場合に、MACsec でセッションをセキュアにします。設定されていない場合、デフォルト値は *should secure* です。
ステップ 10	**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。スイッチとクライアント間の認証交換に基づいてポートが許可ステートまたは無許可ステートに変わります。
ステップ 11	**authentication periodic** 例： `Device(config-if)# authentication periodic`	（任意）このポートの再認証を有効または無効にします。
ステップ 12	**authentication timer reauthenticate** 例： `Device(config-if)# authentication timer reauthenticate`	（任意）1 から 65535 までの値（秒）を入力します。サーバーから再認証タイムアウト値を取得します。デフォルトの再認証時間は 3600 秒です。
ステップ 13	**authentication violation protect** 例： `Device(config-if)# configure terminal`	新しいデバイスがポートに接続された場合、または最大数のデバイスがポートに接続された後に新しいデバイスがそのポートに接続された場合に、予期しない着信 MAC アドレスを破棄するようポートを設定します。設定されていない場合、デフォルトではポートをシャット ダウンします。
ステップ 14	**mka policy** *policy-name* 例： `Device(config-if)# mka policy mka_policy`	既存の MKA プロトコル ポリシーをインターフェイスに適用し、インターフェイス上で MKA を有効にします。MKA ポリシーを設定しなかった場合

	コマンドまたはアクション	目的
		（**mka policy** 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）。
ステップ 15	**dot1x pae authenticator** 例： `Device(config-if)# dot1x pae authenticator`	ポートを 802.1x 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 16	**spanning-tree portfast** 例： `Device(config-if)# spanning-tree portfast`	関連するすべてのVLAN内のインターフェイスで、スパニングツリー Port Fast を有効にします。Port Fast機能が有効の場合、インターフェイスはブロッキングステートからフォワーディングステートに直接移行します。その際に、中間のスパニングツリーステートは変わりません
ステップ 17	**end** 例： `Device(config)# 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 18	**show authentication session interface** *interface-id* 例： `Device# show authentication session interface GigabitEthernet 1/0/1`	許可されたセッションのセキュリティ ステータスを確認します。
ステップ 19	**show mka sessions** 例： `Device# show mka sessions`	確立された MKA セッションを確認します。

PSK を使用する MACsec MKA の設定

事前共有キー（PSK）を使用して、MACsec MKA ポリシーを設定するには、特権 EXEC モードで次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **key chain** *key-chain-name* **macsec**
4. **key** *hex-string*
5. **cryptographic-algorithm** *{aes-128-cmac | aes-256-cmac}*
6. **key-string** *{ [0|6|7] pwd-string | pwd-string}*
7. **lifetime local** [*start timestamp {hh::mm::ss | day | month | year}*] [**duration** *seconds* | *end timestamp {hh::mm::ss | day | month | year}*]
8. **end**

手順の詳細

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 chain** *key-chain-name* **macsec** 例： `Device(config)# key chain keychain1 macsec`	鍵チェーンを設定して、鍵 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	**key** *hex-string* 例： `Device(config-key-chain)# key 1000`	鍵チェーン内の各キーの固有識別子を設定し、鍵チェーンのキー コンフィギュレーション モードを開始します。 （注） 128 ビット暗号化の場合は、1 ～ 32 文字の 16 進数キー文字列を使用します。256 ビット暗号の場合は、64 文字の 16 進数キー文字列を使用します。
ステップ 5	**cryptographic-algorithm** *{aes-128-cmac | aes-256-cmac}* 例： `Device(config-key-chain)# cryptographic-algorithm aes-128-cmac`	128 ビットまたは 256 ビット暗号による暗号化認証アルゴリズムを設定します。
ステップ 6	**key-string** *{ [0|6|7] pwd-string | pwd-string}* 例： `Device(config-key-chain)# key-string 12345678901234567890123456789012`	鍵文字列のパスワードを設定します。16進数の文字のみを入力する必要があります。
ステップ 7	**lifetime local** [*start timestamp {hh::mm::ss | day | month | year}*] [**duration** *seconds* | *end timestamp {hh::mm::ss | day | month | year}*] 例： `Device(config-key-chain)# lifetime local 12:12:00 July 28 2016 12:19:00 July 28 2016`	事前共有鍵の有効期間を設定します。
ステップ 8	**end** 例：	キー チェーン コンフィギュレーション モードを終了して、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-key-chain)# end`	

PSK を使用する MACsec MKA のインターフェイスへの設定

事前共有キー（PSK）を使用する MACsec MKA ポリシーをインターフェイスに設定するには、特権 EXEC モードで次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **macsec network-link**
5. **mka policy** *policy-name*
6. **mka pre-shared-key key-chain** *key-chain name* [**fallback key-chain** *key-chain name*]
7. **macsec replay-protection window-size** *frame number*
8. **end**

手順の詳細

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-if)# interface GigabitEthernet 0/0/0`	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	**macsec network-link** 例： `Device(config-if)# macsec network-link`	インターフェイス上で MACsec を有効にします。
ステップ 5	**mka policy** *policy-name* 例： `Device(config-if)# mka policy mka_policy`	MKA ポリシーを設定します。

	コマンドまたはアクション	目的
ステップ 6	**mka pre-shared-key key-chain** *key-chain name* [**fallback key-chain** *key-chain name*] 例： `Device(config-if)# mka pre-shared-key key-chain key-chain-name`	MKA 事前共有鍵の鍵チェーン名を設定します。
ステップ 7	**macsec replay-protection window-size** *frame number* 例： `Device(config-if)# macsec replay-protection window-size 10`	リプレイ保護の MACsec ウィンドウ サイズを設定します。
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。

次のタスク

セッションの実行中に MKA PSK が設定されたインターフェイスで MKA ポリシーを変更することは推奨されません。ただし、変更が必要な場合は、次のようにポリシーを再設定する必要があります。

1. **no macsec network-link** コマンドを使用して、各参加ノードの macsec network-link 設定を削除し、既存のセッションを無効にします。
2. **mka policy policy-name** コマンドを使用して、各参加ノードのインターフェイスで MKA ポリシーを設定します。
3. **macsec network-link** コマンドを使用して、各参加ノードで新しいセッションを有効にします。

証明書ベースの MACsec を使用する MACsec MKA の設定

ポイントツーポイント リンクで MKA による MACsec を設定するには、次のタスクを実行します。

- 証明書登録の設定
  - キー ペアの生成
  - SCEP 登録の設定
  - 証明書の手動設定
- 認証ポリシーの設定
- 証明書ベース MACsec 暗号化プロファイルと IEEE 802.1x ログイン情報の設定

• インターフェイスで証明書ベース MACsec を使用する MKA MACsec の設定

キー ペアの生成

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa label** *label-name* **general-keys modulus** *size*
4. **end**
5. **show authentication session interface** *interface-id*

手順の詳細

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 label** *label-name* **general-keys modulus** *size* 例： `Device(config)# crypto key generate rsa label general-keys  modulus 2048`	署名および暗号化用に RSA 鍵ペアを作成します。 label キーワードを使用すると、各鍵ペアにラベルを割り当てることもできます。このラベルは、鍵ペアを使用するトラストポイントによって参照されます。ラベルを割り当てなかった場合、鍵ペアには <Default-RSA-Key> というラベルが自動的に付けられます。 追加のキーワードを使用しない場合、このコマンドは汎用 RSA 鍵ペアを 1 つ生成します。法（modulus）が指定されていない場合は、デフォルトの鍵の法である 1024 が使用されます。その他の法サイズを指定するには、modulus キーワードを使用します。
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。

	コマンドまたはアクション	目的
ステップ 5	**show authentication session interface** *interface-id* 例： `Device# show authentication session interface gigabitethernet 0/1/1`	許可されたセッションのセキュリティ ステータスを確認します。

SCEP を使用した登録の設定

Simple Certificate Enrollment Protocol（SCEP）は、HTTP を使用して認証局（CA）または登録局（RA）と通信する、シスコが開発した登録プロトコルです。SCEP は、要求および証明書の送受信用に最も一般的に使用される方式です。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *server name*
4. **enrollment url** *url name pem*
5. **rsakeypair** *label*
6. **serial-number none**
7. **ip-address none**
8. **revocation-check crl**
9. **auto-enroll** *percent* **regenerate**
10. **exit**
11. **crypto pki authenticate** *name*
12. **end**
13. **show crypto pki certificate** *trustpoint name*

手順の詳細

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	**crypto pki trustpoint** *server name* 例： `Device(config)# crypto pki trustpoint ka`	トラストポイントおよび設定された名前を宣言して、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 4	**enrollment url** *url name pem* 例： `Device(ca-trustpoint)# enrollment url http://url:80`	デバイスが証明書要求を送信する CA の URL を指定します。 URL 内の IPv6 アドレスは括弧で囲む必要があります。たとえば、http:// [2001:DB8:1:1::1]:80 です。 pem キーワードは、証明書要求に Privacy Enhanced Mail（PEM）の境界を追加します。
ステップ 5	**rsakeypair** *label* 例： `Device(ca-trustpoint)# rsakeypair exampleCAkeys`	証明書に関連付けるキー ペアを指定します。 （注） **rsakeypair** 名は、信頼ポイント名と一致している必要があります。
ステップ 6	**serial-number none** 例： `Device(ca-trustpoint)# serial-number none`	**none** キーワードは、証明書要求にシリアル番号が含まれないことを指定します。
ステップ 7	**ip-address none** 例： `Device(ca-trustpoint)# ip-address none`	**none** キーワードは、証明書要求に IP アドレスが含まれないことを指定します。
ステップ 8	**revocation-check crl** 例： `Device(ca-trustpoint)# revocation-check crl`	ピアの証明書が取り消されていないことを確認する方法として CRL を指定します。
ステップ 9	**auto-enroll** *percent* **regenerate** 例： `Device(ca-trustpoint)# auto-enroll 90 regenerate`	自動登録を有効にします。これにより、クライアントはCAから自動的にロールオーバー証明書を要求できます。 自動登録が有効でない場合、証明書の失効時にクライアントを手動で PKI に再登録する必要があります。 デフォルトでは、デバイスのドメイン ネーム システム（DNS）名だけが証明書に含められます。 現行の証明書の有効期間が指定のパーセンテージに達したときに、新しい証明書が要求されるように指定するには、percent 引数を使用します。

	コマンドまたはアクション	目的
		名前付きの鍵がすでに存在する場合でも、証明書の新しい鍵を生成するには、regenerate キーワードを使用します。 ロールオーバー中の鍵ペアがエクスポート可能な場合、新しい鍵ペアもエクスポート可能です。次のコメントがトラストポイント コンフィギュレーションに表示され、鍵ペアがエクスポート可能かどうかが示されます。「! RSA key pair associated with trustpoint is exportable.」 新しい鍵ペアは、セキュリティ上の問題に対処するために生成することを推奨します。
ステップ 10	**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 11	**crypto pki authenticate** *name* 例： `Device(config)# crypto pki authenticate myca`	CA 証明書を取得して、認証します。
ステップ 12	**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 13	**show crypto pki certificate** *trustpoint name* 例： `Device# show crypto pki certificate ka`	信頼ポイントの証明書に関する情報を表示します。

手動による登録の設定

CA が SCEP をサポートしない場合、またはルータと CA 間のネットワーク接続が不可能な場合。手動での証明書登録を設定するには、次の作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *server name*
4. **enrollment url** *url name pem*
5. **rsakeypair** *label*
6. **serial-number none**
7. **ip-address none**
8. **revocation-check crl**

9. **exit**
10. **crypto pki authenticate** *name*
11. **crypto pki enroll** *name*
12. **crypto pki import** *name* **certificate**
13. **end**
14. **show crypto pki certificate** *trustpoint name*

手順の詳細

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** *server name* 例： `Device# crypto pki trustpoint ka`	トラストポイントおよび設定された名前を宣言して、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 4	**enrollment url** *url name pem* 例： `Device(ca-trustpoint)# enrollment url http://url:80`	デバイスが証明書要求を送信する CA の URL を指定します。 URL 内の IPv6 アドレスは括弧で囲む必要があります。たとえば、http:// [2001:DB8:1:1::1]:80 です。 pem キーワードは、証明書要求に Privacy Enhanced Mail（PEM）の境界を追加します。
ステップ 5	**rsakeypair** *label* 例： `Device(ca-trustpoint)# rsakeypair exampleCAkeys`	証明書に関連付けるキー ペアを指定します。
ステップ 6	**serial-number none** 例： `Device(ca-trustpoint)# serial-number none`	**none** キーワードは、証明書要求にシリアル番号が含まれないことを指定します。
ステップ 7	**ip-address none** 例： `Device(ca-trustpoint)# ip-address none`	**none** キーワードは、証明書要求に IP アドレスが含まれないことを指定します。

	コマンドまたはアクション	目的
ステップ 8	**revocation-check crl** 例： `Device(ca-trustpoint)# revocation-check crl`	ピアの証明書が取り消されていないことを確認する方法として CRL を指定します。
ステップ 9	**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 10	**crypto pki authenticate** *name* 例： `Device(config)# crypto pki authenticate myca`	CA 証明書を取得して、認証します。
ステップ 11	**crypto pki enroll** *name* 例： `Device(config)# crypto pki enroll myca`	証明書要求を生成し、証明書サーバーにコピーおよびペーストするために要求を表示します。 プロンプトが表示されたら、登録情報を入力します。たとえば、証明書要求にデバイスの FQDN および IP アドレスを含めるかどうかを指定します。 コンソール端末に対して証明書要求を表示するかについても選択できます。 必要に応じて、Base 64 符号化証明書を PEM ヘッダーを付けて、または付けずに表示します。
ステップ 12	**crypto pki import** *name* **certificate** 例： `Device(config)# crypto pki import myca certificate`	許可された証明書を取得するコンソール端末で、TFTP によって証明書をインポートします。 デバイスは、拡張子が「.req」から「.crt」に変更されたことを除いて、要求の送信に使用した同じファイル名を使用して、許可された証明書を TFTP によって取得しようと試みます。用途鍵証明書の場合、拡張子「-sign.crt」および「-encr.crt」が使用されます。 デバイスは、受信したファイルを解析して証明書を検証し、証明書をスイッチの内部証明書データベースに挿入します。 （注） 一部の CA は、証明書要求の用途鍵情報を無視し、汎用目的の証明書を発行します。ご使用の CA が証明書要求の用途鍵情報を無視する場合は、汎用目的の証明書だけをインポートしてください。ルータは、生成される 2 つの鍵ペアのいずれも使用しません。

	コマンドまたはアクション	目的
ステップ 13	**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 14	**show crypto pki certificate** *trustpoint name* 例： `Device# show crypto pki certificate  ka`	信頼ポイントの証明書に関する情報を表示します。

802.1x 認証の有効化と AAA の設定

手順の概要

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **dot1x system-auth-control**
5. **radius server** *name*
6. **address** *ip_address* **auth-port** *port_number* **acct-port** *port_number*
7. **automate-tester username** *username*
8. **key** *string*
9. **radius-server deadtime** *minutes*
10. **exit**
11. **aaa group server radius** *group_name*
12. **server** *name*
13. **exit**
14. **aaa authentication dot1x default group** *group_name*
15. **aaa authorization network default group** *group_name*

手順の詳細

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 を有効にします。
ステップ 4	**dot1x system-auth-control** 例： `device(config)# dot1x system-auth-control`	デバイス上で 802.1X を有効にします。
ステップ 5	**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。
ステップ 6	**address** *ip_address* **auth-port** *port_number* **acct-port** *port_number* 例： `device(config-radius-server)# address ipv4 10.64.72.90 auth-port 1645 acct-port 1646`	RADIUS サーバー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 7	**automate-tester username** *username* 例： `device(config-radius-server)# automate-tester username dummy`	RADIUS サーバーの自動テスト機能を有効にします。 このようにすると、デバイスは RADIUS サーバーにテスト認証メッセージを定期的に送信し、サーバーからの RADIUS 応答を待機します。成功メッセージは必須ではありません。認証失敗であっても、サーバーが稼働していることを示しているため問題ありません。
ステップ 8	**key** *string* 例： `device(config-radius-server)# key dummy123`	デバイスと RADIUS サーバー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。
ステップ 9	**radius-server deadtime** *minutes* 例： `device(config-radius-server)#radius-server deadtime 2`	いくつかのサーバーが使用不能になったときの RADIUS サーバーの応答時間を短くし、使用不能になったサーバーがすぐにスキップされるようにします。
ステップ 10	**exit**	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	**aaa group server radius** *group_name* 例： `device(config)# aaa group server radius ISEGRP`	異なる RADIUS サーバー ホストを別々のリストと方式にグループ化し、サーバー グループ コンフィギュレーション モードを開始します。
ステップ 12	**server** *name*	

	コマンドまたはアクション	目的
	例： `device(config)#server ise`	
ステップ 13	**exit** 例： `device(config-radius-server)# exit`	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 14	**aaa authentication dot1x default group** *group_name* 例： `device(config)# aaa authentication dot1x default group ISEGRP`	IEEE 802.1x 用にデフォルトの認証サーバ グループを設定します。
ステップ 15	**aaa authorization network default group** *group_name* 例： `device(config)# aaa authorization network default group ISEGRP`	ネットワーク認証のデフォルト グループを設定します。

802.1x MKA MACsec 設定のインターフェイスへの適用

EAP-TLS を使用する MKA MACsec をインターフェイスに適用するには、次のステップを実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *interface_id*
4. **macsec network-link**
5. **authentication periodic**
6. **authentication timer reauthenticate interval**
7. **access-session host-mode multi-domain**
8. **access-session closed**
9. **access-session port-control auto**
10. **dot1x pae both**
11. **dot1x credentials profile**
12. **dot1x supplicant eap profile** *profile_name*
13. **dot1x authenticator eap profile** *profile_name*
14. **service-policy type control subscriber** *control_policy_name*
15. **exit**
16. **show macsec interface**
17. **copy running-config startup-config**

手順の詳細

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 te0/1/2`	MACsec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。インターフェイスは物理インターフェイスでなければなりません。
ステップ 4	**macsec network-link** 例： `device(config)# macsec network-link`	インターフェイス上で MACsec をイネーブルにします。
ステップ 5	**authentication periodic** 例： `device(config)# authentication periodic`	このポートの再認証をイネーブルにします。
ステップ 6	**authentication timer reauthenticate interval** 例： `device(config)# authentication timer reauthenticate interval`	再認証間隔を設定します。
ステップ 7	**access-session host-mode multi-domain** 例： `device(config)# access-session host-mode multi-domain`	ホストにインターフェイスへのアクセスを許可します。
ステップ 8	**access-session closed** 例： `device(config)# access-session closed`	インターフェイスへの事前認証アクセスを防止します。
ステップ 9	**access-session port-control auto** 例： `device(config)# access-session port-control auto`	ポートの認可状態を設定します。

	コマンドまたはアクション	目的
ステップ 10	**dot1x pae both** 例： `device(config)# dot1x pae both`	ポートを 802.1X ポート アクセス エンティティ（PAE）のサプリカントおよびオーセンティケータとして設定します。
ステップ 11	**dot1x credentials profile** 例： `device(config)# dot1x credentials profile`	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割り当てます。
ステップ 12	**dot1x supplicant eap profile** *profile_name* 例： `device(config)# dot1x supplicant eap profile eap1`	EAP-TLS プロファイルをインターフェイスに割り当てます。
ステップ 13	**dot1x authenticator eap profile** *profile_name* 例： `device(config)# dot1x authenticator eap profile eap1`	802.1x 認証時に使用する EAP-TLS プロファイルを割り当てます。
ステップ 14	**service-policy type control subscriber** *control_policy_name* 例： `device(config)# service-policy type control subscriber controlPolicy2`	インターフェイスに加入者制御ポリシーを適用します。
ステップ 15	**exit** 例： `device(config)# exit`	特権 EXEC モードに戻ります。
ステップ 16	**show macsec interface** 例： `device# show macsec interface`	インターフェイスの MACsec の詳細を表示します。
ステップ 17	**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。

PSK を使用する MKA/MACsec のポートチャネルへの設定

事前共有キー（PSK）を使用する MKA ポリシーをインターフェイスに設定するには、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。
ステップ 3	**interface** *interface-id* 例： `Device(config-if)# interface gigabitethernet 1/0/3`	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	**macsec network-link** 例： `Device(config-if)# macsec network-link`	インターフェイス上で MACsec を有効にします。レイヤ 2 およびレイヤ 3 ポートチャネルをサポートします。
ステップ 5	**mka policy** *policy-name* 例： `Device(config-if)# mka policy mka_policy`	MKA ポリシーを設定します。
ステップ 6	**mka pre-shared-key key-chain** *key-chain name* [**fallback key-chain** *key-chain name*] 例： `Device(config-if)# mka pre-shared-key key-chain key-chain-name`	MKA 事前共有キーのキーチェーン名を設定します。 （注） MKA 事前共有キーは、物理インターフェイスまたはサブインターフェイスで設定できますが、両方で設定することはできません。
ステップ 7	**macsec replay-protection window-size** *frame number* 例： `Device(config-if)# macsec replay-protection window-size 0`	リプレイ保護の MACsec ウィンドウ サイズを設定します。
ステップ 8	**channel-group** *channel-group-number* **mode** {**auto** | **desirable**} | {**active** | **passive**} | {**on**} 例： `Device(config-if)# channel-group 3 mode auto active on`	チャネル グループ内にポートを設定し、モードを設定します。 （注） インターフェイスで MACsec を設定しないと、チャネルグループのポートを設定できません。このステップの前に、ステップ 3、4、5、および 6 のコマンド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
		channel-number の指定できる範囲は 1 ～ 4096 です。ポートチャネルがない場合は、このチャネルグループに関連付けられたポートチャネルが自動的に作成されます。モードには、次のキーワードのいずれか 1 つを選択します。 • **auto**：PAgP デバイスが検出された場合に限り、PAgP を有効にします。ポートをパッシブ ネゴシエーション ステートにします。この場合、ポートは受信する PAgP パケットに応答しますが、PAgP パケットネゴシエーションを開始することはありません。 （注） EtherChannel メンバが、スイッチスタックにある異なるスイッチのメンバである場合、**auto** キーワードはサポートされません。 • **desirable**：無条件に PAgP を有効にします。ポートをアクティブ ネゴシエーション ステートにします。この場合、ポートは PAgP パケットを送信することによって、相手ポートとのネゴシエーションを開始します。 （注） EtherChannel メンバが、スイッチスタックにある異なるスイッチのメンバである場合、**desirable** キーワードはサポートされません。 • **on**：PAgP または LACP を使用せずにポートが強制的にチャネル化されます。on モードでは、EtherChannel が存在するのは、**on** モードのポートグループが、**on** モードの別のポートグループに接続する場合だけです。 • **active**：LACP デバイスが検出された場合に限り、LACP を有効にします。ポートをアクティブ ネゴシエーション ステートにします。この場合、ポートは LACP パケットを送信することによって、相手ポートとのネゴシエーションを開始します。 • **passive**：ポート上で LACP を有効にして、ポートをパッシブ ネゴシエーション ステートにします。この場合、ポートは受信する LACP パ

	コマンドまたはアクション	目的
		ケットに応答しますが、LACP パケットネゴシエーションを開始することはありません。
ステップ 9	**end** 例： `Device(config-if)# cend`	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。

レイヤ 2 EtherChannel 用のポートチャネル論理インターフェイスの設定

レイヤ 2 EtherChannel 用のポート チャネル インターフェイスを作成するには、次の作業を行います。

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	**interface port-channel** *channel-group-number* 例： `Device(config)# interface port-channel 1`	ポート チャネル インターフェイスを作成します。 （注） ポートチャネルインターフェイスを削除するには、このコマンドの **no** 形式を使用します。
ステップ 4	**switchport** 例： `Device(config-if)# switchport`	レイヤ 3 モードになっているインターフェイスを、レイヤ 2 設定のレイヤ 2 モードに切り替えます。
ステップ 5	**switchport mode** {**access** | **trunk**} 例： `Device(config-if)# switchport mode access`	すべてのポートをスタティックアクセスポートとして同じVLANに割り当てるか、またはトランクとして設定します。
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。

レイヤ 3 EtherChannel 用のポートチャネル論理インターフェイスの設定

レイヤ 3 EtherChannel 用のポート チャネル インターフェイスを作成するには、次の作業を行います。

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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/2`	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	**no switchport** 例： `Device(config-if)# no switchport`	レイヤ 2 モードになっているインターフェイスを、レイヤ 3 設定用にレイヤ 3 モードに切り替えます。
ステップ 5	**ip address** *ip-address subnet_mask* 例： `Device(config-if)# ip address 10.2.2.3 255.255.255.254`	EtherChannel に IP アドレスおよびサブネット マスクを割り当てます。
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。

MACsec 暗号アナウンスメントの設定

セキュアアナウンスメントの MKA ポリシーの設定

MKA プロトコルポリシーを作成して MKPDU でセキュアアナウンスメントを有効にするには、特権 EXEC モードで次の手順を実行します。デフォルトでは、セキュアアナウンスメントは無効になっています。

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	**mka policy** *policy-name* 例： `Device(config)# mka policy mka_policy`	MKA ポリシーを指定して、MKA ポリシー コンフィギュレーション モードを開始します。ポリシー名の長さは最大で 16 文字です。 （注） MKA ポリシーのデフォルトの MACsec 暗号スイートは GCM-AES-128 です。 デバイスが「GCM-AES-128」および「GCM-AES-256」の両方の暗号方式をサポートしている場合は、ユーザー定義の MKA ポリシーを定義して使用し、必要に応じて、128 および 256 ビット両方の暗号を含めるか、または 256 ビットのみの暗号を含めることを強くお勧めします。
ステップ 4	**key-server** *priority* 例： `Device(config-mka-policy)# key-server priority 200`	MKA キーサーバーオプションを設定し、0 ～ 255 の間で優先順位を設定します。 （注） キーサーバーの優先順位の値を 255 に設定した場合、ピアはキーサーバーになることはできません。キーサーバーの優先順位の値は MKA PSK に対してのみ有効です。これは MKA EAP-TLS には適用されません。
ステップ 5	**send-secure-announcements** 例： `Device(config-mka-policy)# send-secure-announcements`	セキュアアナウンスメントの送信を有効にします。セキュアアナウンスメントの送信を無効にするには、このコマンドの **no** 形式を使用します。デフォルトでは、セキュアアナウンスメントは無効になっています。
ステップ 6	**macsec-cipher-suite** {*gcm-aes-128* | *gcm-aes-256*} 例： `Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128`	128 ビットまたは 256 ビット暗号化により SAK を取得するための暗号スイートを設定します。

	コマンドまたはアクション	目的
ステップ 7	**end** 例： `Device(config-mka-policy)# end`	MKA ポリシー 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。
ステップ 8	**show mka policy** 例： `Device# show mka policy`	MKA ポリシーを表示します。

セキュアアナウンスメントのグローバル設定

特権 EXEC モードから始めて、次の手順に従って、すべての MKA ポリシーにわたって安全なアナウンスメントをグローバルに有効にします。

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	**mka defaults policy send-secure-announcements** 例： `Device(config)# mka defaults policy send-secure-announcements`	MKA ポリシーを介した MKPDU でのセキュアアナウンスメントの送信を有効にします。デフォルトでは、セキュアアナウンスメントは無効になっています。
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。

インターフェイスでの EAPoL アナウンスメントの設定

インターフェイスで EAPoL アナウンスメントを設定するには、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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	MACsec 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。インターフェイスは物理インターフェイスでなければなりません。
ステップ 4	**eapol annoucement** 例： `Device(config-if)# eapol announcement`	EAPoL アナウンスメントを有効にします。EAPoL アナウンスメントを無効にするには、コマンドの **no** 形式を使用します。デフォルトでは、EAPoL アナウンスメントは無効になっています。
ステップ 5	**end** 例： `Device(config-if)# configure terminal`	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。

MACsec 暗号化の設定例

例：MKA および MACsec の設定

次に、MKA ポリシーを作成する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# mka policy mka_policy
Device(config-mka-policy)# key-server priority 200
Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Device(config-mka-policy)# confidentiality-offset 30
Device(config-mka-policy)# ssci-based-on-sci
Device(config-mka-policy)#end
```

次は、インターフェイスに MACsec を設定する例です。

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# switchport access vlan 1
Device(config-if)# switchport mode access
```

```
Device(config-if)# macsec
Device(config-if)#access-session event linksec fail action authorize vlan 1
Device(config-if)# access-session host-mode multi-domain
Device(config-if)# access-session linksec policy must-secure
Device(config-if)# access-session port-control auto
Device(config-if)#authentication periodic
Device(config-if)# authentication timer reauthenticate
Device(config-if)# authentication violation protect
Device(config-if)#mka policy mka_policy
Device(config-if)# dot1x pae authenticator
Device(config-if)# spanning-tree portfast
Device(config-if)#end
```

例：PSK を使用する MACsec MKA の設定

次に、PSK を使用して、MKA MACsec を設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# Key chain keychain1 macsec
Device(config-keychain)# key 1000
Device(config-keychain-key)# cryptographic-algorithm aes-128-cmac
Device(config-keychain-key)# key-string 12345678901234567890123456789012
Device(config-keychain-key)# lifetime local 12:12:00 July 28 2016 12:19:00 July 28 2016
Device(config-keychain-key)# end
```

次に、PSK を使用して、インターフェイスに MACsec MKA を設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# mka policy mka_policy
Device(config-if)# mka pre-shared-key key-chain key-chain-name
Device(config-if)# macsec replay-protection window-size 10
Device(config-if)# end
```

MKA-PSK：CKN 動作の変更

Cisco IOS XE Fuji 16.8.1 リリース以降、MKA PSK セッションの場合、CKN は、固定の 32 バイトではなく、キーの 16 進文字列として設定されている CKN とまったく同じ文字列を使用します。

```
Device> enable
Device# configure terminal
Device(config)# key chain abc macsec
Device(config-keychain)# key 11
Device(config-keychain-key)# cryptographic-algorithm aes-128-cmac
Device(config-keychain-key)# key-string 12345678901234567890123456789013
Device(config-keychain-key)# lifetime local 12:21:00 Sep 9 2015 infinite
Device(config-keychain-key)# end
```

以下は、上記の設定に対する **show mka session** コマンドの出力例です。

```
Device# show mka session

Total MKA Sessions....... 1
Secured Sessions... 1
Pending Sessions... 0
=============================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited         Key-Server
```

```
Port-ID        Peer-RxSCI          MACsec-Peers     Status            CKN

=============================================================================================
Et0/0          aabb.cc00.6600/0002     icv            NO                NO

2              aabb.cc00.6500/0002 1                Secured           11    *Note that
 the CKN key-string is exactly the same that has been configured for the key as
hex-string.*
```

一方でCKN動作が変更され、もう一方でCKN動作が変更されていない2つのイメージ間の相互運用性の場合、キーの16進数文字列は64文字の16進数文字列である必要があります。この文字列は、CKN動作が変更されたイメージを持つデバイスで動作するようにゼロパディングされている必要があります。次の例を参照してください。

CKN キー文字列の動作が変更されていない設定：

```
Device# configure terminal
Device(config)# key chain abc macsec
Device(config-keychain)# key 11
Device(config-keychain-key)# cryptographic-algorithm aes-128-cmac
Device(config-keychain-key)# key-string 12345678901234567890123456789013
Device(config-keychain-key)# lifetime local 12:21:00 Sep 9 2015 infinite
Device(config-keychain-key)# end
```

CKN キー文字列の動作が変更された設定：

```
Device# configure terminal
Device(config)# key chain abc macsec
Device(config-keychain)# key
1100000000000000000000000000000000000000000000000000000000000000
Device(config-keychain-key)# cryptographic-algorithm aes-128-cmac
Device(config-keychain-key)# key-string 12345678901234567890123456789013
Device(config-keychain-key)# lifetime local 12:21:00 Sep 9 2015 infinite
Device(config-keychain-key)# end
```

例：証明書ベース MACsec を使用した MACsec MKA の設定

この例では、証明書ベースMACsecを使用したMACsecMKAの設定方法について説明します。

```
Device> enable
Device# configure terminal
Device(config)# interface Gigabitethernet 1/0/1
Device(config-if)# macsec network-link
Device(config-if)# authentication periodic
Device(config-if)# authentication timer reauthenticate interval
Device(config-if)#access-session host-mode multi-domain
Device(config-if)# access-session closed
Device(config-if)# access-session port-control auto
Device(config-if)# dot1x pae both
Device(config-if)#dot1x credentials profile
Device(config-if)# dot1x supplicant eap profile profile_eap_tls
Device(config-if)#service-policy type control subscriber sub1
Device(config-if)# end
```

例：PSK を使用する MACsec MKA のポートチャネルへ設定

Etherchannel モード - Static/On

次に、EtherChannelモードがオンのデバイス1およびデバイス2の設定例を示します。

```
Device> enable
Device# configure terminal
Device(config)# key chain KC macsec
Device(config-key-chain)# key 1000
Device(config-key-chain)# cryptographic-algorithm aes-128-cmac
Device(config-key-chain)# key-string FC8F5B10557C192F03F60198413D7D45
Device(config-key-chain)# exit
Device(config)# mka policy POLICY
Device(config-mka-policy)# key-server priority 0
Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Device(config-mka-policy)# confidentiality-offset 0
Device(config-mka-policy)# exit
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# channel-group 2 mode on
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# exit
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# channel-group 2 mode on
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# end
```

レイヤ 2 EtherChannel 設定

デバイス 1

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end
```

デバイス 2

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end
```

次に、**show etherchannel summary** コマンドの出力例を示します。

```
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
```

```
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG


 Number of channel-groups in use: 1
 Number of aggregators:           1

 Group  Port-channel  Protocol    Ports

------+-------------+-----------+-----------------------------------------------

 2      Po2(RU)           -        Te1/0/1(P)  Te1/0/2(P)
```

レイヤ 3 EtherChannel 設定

デバイス 1

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 10.25.25.3 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

デバイス 2

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 10.25.25.4 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

次に、**show etherchannel summary** コマンドの出力例を示します。

```
 Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG


 Number of channel-groups in use: 1
 Number of aggregators:           1
```

```
 Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------

 2      Po2(RU)           -        Te1/0/1(P)  Te1/0/2(P)
```

EtherChannel モード - LACP

次に、EtherChannel モードが LACP のデバイス 1 およびデバイス 2 の設定例を示します。

```
Device> enable
Device# configure terminal
Device(config)# key chain KC macsec
Device(config-key-chain)# key 1000
Device(config-key-chain)# cryptographic-algorithm aes-128-cmac
Device(config-key-chain)# key-string FC8F5B10557C192F03F60198413D7D45
Device(config-key-chain)# exit
Device(config)# mka policy POLICY
Device(config-mka-policy)# key-server priority 0
Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Device(config-mka-policy)# confidentiality-offset 0
Device(config-mka-policy)# exit
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# channel-group 2 mode active
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# exit
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# channel-group 2 mode active
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# end
```

レイヤ 2 EtherChannel 設定

デバイス 1

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end
```

デバイス 2

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end
```

次に、**show etherchannel summary** コマンドの出力例を示します。

```
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG


Number of channel-groups in use: 1
Number of aggregators:           1


------+-------------+-----------+-----------------------------------------------

2      Po2(SU)          LACP      Te1/1/1(P)   Te1/1/2(P)
```

レイヤ 3 EtherChannel 設定

デバイス 1

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 10.25.25.3 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

デバイス 2

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 10.25.25.4 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

次に、**show etherchannel summary** コマンドの出力例を示します。

```
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port
```

```
        A - formed by Auto LAG


 Number of channel-groups in use: 1
 Number of aggregators:           1

 Group  Port-channel  Protocol    Ports

------+-------------+-----------+-----------------------------------------------

 2      Po2(RU)          LACP      Te1/1/1(P)  Te1/1/2(P)
```

EtherChannel モード - PAgP

次に、EtherChannel モードが PAgP のデバイス 1 およびデバイス 2 の設定例を示します。

```
Device> enable
Device# configure terminal
Device(config)# key chain KC macsec
Device(config-key-chain)# key 1000
Device(config-key-chain)# cryptographic-algorithm aes-128-cmac
Device(config-key-chain)# key-string FC8F5B10557C192F03F60198413D7D45
Device(config-key-chain)# exit
Device(config)# mka policy POLICY
Device(config-mka-policy)# key-server priority 0
Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Device(config-mka-policy)# confidentiality-offset 0
Device(config-mka-policy)# exit
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# channel-group 2 mode desirable
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# exit
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# channel-group 2 mode desirable
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# end
```

レイヤ 2 EtherChannel 設定

デバイス 1

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end
```

デバイス 2

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# switchport
```

```
Device(config-if)# switchport mode trunk
Device(config-if)# no shutdown
Device(config-if)# end
```

次に、**show etherchannel summary** コマンドの出力例を示します。

```
 Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG


 Number of channel-groups in use: 1
 Number of aggregators:           1


------+------------+----------+-----------------------------------------------

 2      Po2(SU)          PAgP      Te1/1/1(P)  Te1/1/2(P)
```

レイヤ 3 EtherChannel 設定

デバイス 1

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 10.25.25.3 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

デバイス 2

```
Device> enable
Device# configure terminal
Device(config)# interface port-channel 2
Device(config-if)# no switchport
Device(config-if)# ip address 10.25.25.4 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

次に、**show etherchannel summary** コマンドの出力例を示します。

```
 Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
```

```
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG


 Number of channel-groups in use: 1
 Number of aggregators:           1

 Group  Port-channel  Protocol    Ports

------+-------------+-----------+-----------------------------------------------

 2      Po2(RU)          PAgP      Te1/1/1(P)  Te1/1/2(P)
```

アクティブな MKA セッションの表示

次に、すべてのアクティブな MKA セッションを表示します。

```
Device# show mka sessions interface Te1/0/1

=======================================================================================
Interface      Local-TxSCI          Policy-Name      Inherited
Key-Server
Port-ID        Peer-RxSCI           MACsec-Peers     Status
CKN
=======================================================================================
Te1/0/1        00a3.d144.3364/0025 POLICY           NO
NO
37             701f.539b.b0c6/0032 1                Secured
1000
```

例：MACsec 暗号アナウンスメントの設定

次に、セキュアアナウンスメントの MKA ポリシーの設定例を示します。

```
Device> enable
Device# configure terminal
Device(config)# mka policy mka_policy
Device(config-mka-policy)# key-server 2
Device(config-mka-policy)# send-secure-announcements
Device(config-mka-policy)#macsec-cipher-suite gcm-aes-128confidentiality-offset 0
Device(config-mka-policy)# end
```

次に、セキュアアナウンスメントのグローバル設定例を示します。

```
Device> enable
Device# configure terminal
Device(config)# mka defaults policy send-secure-announcements
Device(config)# end
```

次に、インターフェイスでの EAPoL アナウンスメントの設定例を示します。

```
Device> enable
Device# configure terminal
```

```
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# eapol announcement
Device(config-if)# end
```

次に、EAPoL アナウンスメントが有効になっている **show running-config interface** *interface-name* コマンドの出力例を示します。

```
Device# show running-config interface GigabitEthernet 1/0/1

switchport mode access
 macsec
 access-session host-mode multi-host
 access-session closed
 access-session port-control auto
 dot1x pae authenticator
 dot1x timeout quiet-period 10
 dot1x timeout tx-period 5
 dot1x timeout supp-timeout 10
 dot1x supplicant eap profile peap
 eapol announcement
 spanning-tree portfast
 service-policy type control subscriber Dot1X
```

次に、セキュアアナウンスメントが無効になっている **show mka sessions interface** *interface-name* **detail** コマンドの出力例を示します。

```
Device# show mka sessions interface GigabitEthernet 1/0/1 detail


MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI..............  204c.9e85.ede4/002b
Interface MAC Address....  204c.9e85.ede4
MKA Port Identifier......  43
Interface Name...........  GigabitEthernet1/0/1
Audit Session ID.........
CAK Name (CKN)...........
0100000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)...  D46CBEC05D5D67594543CEAE
Message Number (MN)......  89567
EAP Role.................  NA
Key Server...............  YES
MKA Cipher Suite.........  AES-128-CMAC

Latest SAK Status........  Rx & Tx
Latest SAK AN............  0
Latest SAK KI (KN).......  D46CBEC05D5D67594543CEAE00000001 (1)
Old SAK Status...........  FIRST-SAK
Old SAK AN...............  0
Old SAK KI (KN)..........  FIRST-SAK (0)
```

```
SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)
SAK Retire Time.......... 0s (No Old SAK to retire)

MKA Policy Name.......... p2
Key Server Priority...... 2
Delay Protection......... NO
Replay Protection........ YES
Replay Window Size....... 0
Confidentiality Offset... 0
Algorithm Agility........ 80C201
Send Secure Announcement.. DISABLED
SAK Cipher Suite......... 0080C20001000001 (GCM-AES-128)
MACsec Capability........ 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired........... YES

# of MACsec Capable Live Peers............ 1
# of MACsec Capable Live Peers Responded.. 1

Live Peers List:
  MI                       MN         Rx-SCI (Peer)        KS Priority

  ----------------------------------------------------------------------

  38046BA37D7DA77E06D006A9  89555       c800.8459.e764/002a   10

Potential Peers List:
  MI                       MN         Rx-SCI (Peer)        KS Priority

  ----------------------------------------------------------------------

Dormant Peers List:
  MI                       MN         Rx-SCI (Peer)        KS Priority

  ----------------------------------------------------------------------
```

次に、セキュアアナウンスメントが無効になっている **show mka sessions details** コマンドの出力例を示します。

```
Device# show mka sessions details

MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI............. 204c.9e85.ede4/002b
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier...... 43
Interface Name........... GigabitEthernet1/0/1
Audit Session ID.........
CAK Name (CKN)...........
0100000000000000000000000000000000000000000000000000000000000000
```

```
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)..... 89572
EAP Role................ NA
Key Server.............. YES
MKA Cipher Suite........ AES-128-CMAC

Latest SAK Status....... Rx & Tx
Latest SAK AN........... 0
Latest SAK KI (KN)...... D46CBEC05D5D67594543CEAE00000001 (1)
Old SAK Status.......... FIRST-SAK
Old SAK AN.............. 0
Old SAK KI (KN)......... FIRST-SAK (0)

SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)
SAK Retire Time......... 0s (No Old SAK to retire)

MKA Policy Name......... p2
Key Server Priority..... 2
Delay Protection........ NO
Replay Protection....... YES
Replay Window Size...... 0
Confidentiality Offset... 0
Algorithm Agility....... 80C201
Send Secure Announcement.. DISABLED
SAK Cipher Suite........ 0080C20001000001 (GCM-AES-128)
MACsec Capability....... 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired.......... YES

# of MACsec Capable Live Peers............ 1
# of MACsec Capable Live Peers Responded.. 1

Live Peers List:
  MI                      MN         Rx-SCI (Peer)        KS Priority

  ----------------------------------------------------------------------

  38046BA37D7DA77E06D006A9  89560       c800.8459.e764/002a   10

Potential Peers List:
  MI                      MN         Rx-SCI (Peer)        KS Priority

  ----------------------------------------------------------------------

Dormant Peers List:
  MI                      MN         Rx-SCI (Peer)        KS Priority

  ----------------------------------------------------------------------
```

次に、セキュアアナウンスメントが無効になっている **show mka policy** *policy-name* **detail** コマンドの出力例を示します。

```
Device# show mka policy p2 detail

MKA Policy Configuration ("p2")
==========================
MKA Policy Name........ p2
Key Server Priority.... 2
Confidentiality Offset. 0
Send Secure Announcement..DISABLED
Cipher Suite(s)........ GCM-AES-128

Applied Interfaces...
  GigabitEthernet1/0/1
```

例：MKA 情報の表示

次に、**show mka sessions** コマンドの出力例を示します。

```
Device# show mka sessions


Total MKA Sessions....... 1
      Secured Sessions... 1
      Pending Sessions... 0

=============================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited
Key-Server
Port-ID        Peer-RxSCI          MACsec-Peers     Status
CKN
=============================================================================================
Gi1/0/1        204c.9e85.ede4/002b p2               NO
YES
43             c800.8459.e764/002a 1                Secured
0100000000000000000000000000000000000000000000000000000000000000
```

次に、**show mka sessions interface** *interface-name* コマンドの出力例を示します。

```
Device# show mka sessions interface GigabitEthernet 1/0/1

Summary of All Currently Active MKA Sessions on Interface
GigabitEthernet1/0/1...

=============================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited
Key-Server
Port-ID        Peer-RxSCI          MACsec-Peers     Status
CKN
=============================================================================================
Gi1/0/1        204c.9e85.ede4/002b p2               NO
YES
43             c800.8459.e764/002a 1                Secured
0100000000000000000000000000000000000000000000000000000000000000
```

次に、**show mka sessions interface** *interface-name* **detail** コマンドの出力例を示します。

```
Device# show mka sessions interface GigabitEthernet 1/0/1 detail


MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI..............  204c.9e85.ede4/002b
Interface MAC Address....  204c.9e85.ede4
MKA Port Identifier......  43
Interface Name...........  GigabitEthernet1/0/1
Audit Session ID.........
CAK Name (CKN)...........
0100000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)...  D46CBEC05D5D67594543CEAE
Message Number (MN)......  89567
EAP Role.................  NA
Key Server...............  YES
MKA Cipher Suite.........  AES-128-CMAC

Latest SAK Status........  Rx & Tx
Latest SAK AN............  0
Latest SAK KI (KN).......  D46CBEC05D5D67594543CEAE00000001 (1)
Old SAK Status...........  FIRST-SAK
Old SAK AN...............  0
Old SAK KI (KN)..........  FIRST-SAK (0)

SAK Transmit Wait Time...  0s (Not waiting for any peers to respond)
SAK Retire Time..........  0s (No Old SAK to retire)

MKA Policy Name..........  p2
Key Server Priority......  2
Delay Protection.........  NO
Replay Protection........  YES
Replay Window Size.......  0
Confidentiality Offset...  0
Algorithm Agility........  80C201
Send Secure Announcement..  DISABLED
SAK Cipher Suite.........  0080C20001000001 (GCM-AES-128)
MACsec Capability........  3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired...........  YES

# of MACsec Capable Live Peers............  1
# of MACsec Capable Live Peers Responded..  1

Live Peers List:
  MI                      MN         Rx-SCI (Peer)        KS Priority

  ----------------------------------------------------------------------

  38046BA37D7DA77E06D006A9  89555       c800.8459.e764/002a   10
```

```
Potential Peers List:
  MI                        MN          Rx-SCI (Peer)        KS Priority

  ----------------------------------------------------------------------

Dormant Peers List:
  MI                        MN          Rx-SCI (Peer)        KS Priority

  ----------------------------------------------------------------------
```

次に、**show mka sessions details** コマンドの出力例を示します。

```
Device# show mka sessions details

MKA Detailed Status for MKA Session
===================================
Status: SECURED - Secured MKA Session with MACsec

Local Tx-SCI..... 204c.9e85.ede4/002b
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier...... 43
Interface Name........... GigabitEthernet1/0/1
Audit Session ID.........
CAK Name (CKN)...........
0100000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)...... 89572
EAP Role................. NA
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC

Latest SAK Status........ Rx & Tx
Latest SAK AN............ 0
Latest SAK KI (KN)....... D46CBEC05D5D67594543CEAE00000001 (1)
Old SAK Status........... FIRST-SAK
Old SAK AN............... 0
Old SAK KI (KN).......... FIRST-SAK (0)

SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)
SAK Retire Time.......... 0s (No Old SAK to retire)

MKA Policy Name.......... p2
Key Server Priority...... 2
Delay Protection......... NO
Replay Protection........ YES
Replay Window Size....... 0
Confidentiality Offset... 0
Algorithm Agility........ 80C201
Send Secure Announcement.. DISABLED
SAK Cipher Suite......... 0080C20001000001 (GCM-AES-128)
MACsec Capability........ 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired........... YES
```

```
# of MACsec Capable Live Peers..... 1
# of MACsec Capable Live Peers Responded.. 1

Live Peers List:
  MI                      MN          Rx-SCI (Peer)        KS Priority

  ------------------------------------------------------------------------

  38046BA37D7DA77E06D006A9  89560       c800.8459.e764/002a   10

Potential Peers List:
  MI                      MN          Rx-SCI (Peer)        KS Priority

  ------------------------------------------------------------------------

Dormant Peers List:
  MI                      MN          Rx-SCI (Peer)        KS Priority

  ------------------------------------------------------------------------
```

次に、**show mka policy** コマンドの出力例を示します。

```
Device# show mka policy

MKA Policy Summary...

Policy            KS       Delay   Replay  Window Conf   Cipher
   Interfaces
Name              Priority Protect Protect Size   Offset Suite(s)
   Applied
================================================================================
*DEFAULT POLICY*  0        FALSE   TRUE    0      0      GCM-AES-128

p1                1        FALSE   TRUE    0      0      GCM-AES-128

p2                2        FALSE   TRUE    0      0      GCM-AES-128
   Gi1/0/1
```

次に、**show mka policy** *policy-name* コマンドの出力例を示します。

```
Device# show mka policy p2

MKA Policy Summary...

Policy            KS       Delay   Replay  Window Conf   Cipher
   Interfaces
Name              Priority Protect Protect Size   Offset Suite(s)
   Applied
================================================================================
```

```
p2                  2         FALSE    TRUE     0       0       GCM-AES-128
   Gi1/0/1
```

次に、**show mka policy** *policy-name* **detail** コマンドの出力例を示します。

```
Device# show mka policy p2 detail

MKA Policy Configuration ("p2")
==========================
MKA Policy Name........ p2
Key Server Priority.... 2
Confidentiality Offset. 0
Send Secure Announcement..DISABLED
Cipher Suite(s)........ GCM-AES-128

Applied Interfaces...
  GigabitEthernet1/0/1
```

次に、**show mka statistics interface** *interface-name* コマンドの出力例を示します。

```
Device# show mka statistics interface GigabitEthernet 1/0/1


MKA Statistics for Session
==========================
Reauthentication Attempts.. 0

CA Statistics
   Pairwise CAKs Derived... 0
   Pairwise CAK Rekeys..... 0
   Group CAKs Generated.... 0
   Group CAKs Received..... 0

SA Statistics
   SAKs Generated.......... 1
   SAKs Rekeyed............ 0
   SAKs Received........... 0
   SAK Responses Received.. 1

MKPDU Statistics
   MKPDUs Validated & Rx... 89585
      "Distributed SAK".. 0
      "Distributed CAK".. 0
   MKPDUs Transmitted...... 89596
      "Distributed SAK".. 1
      "Distributed CAK".. 0
```

次に、**show mka summary** コマンドの出力例を示します。

```
Device# show mka summary

Total MKA Sessions....... 1
      Secured Sessions... 1
      Pending Sessions... 0
```

```
=============================================================================
Interface       Local-TxSCI          Policy-Name      Inherited
Key-Server
Port-ID         Peer-RxSCI           MACsec-Peers     Status
CKN
=============================================================================
Gi1/0/1         204c.9e85.ede4/002b p2                NO
YES
43              c800.8459.e764/002a 1                 Secured
0100000000000000000000000000000000000000000000000000000000000000



MKA Global Statistics
=====================
MKA Session Totals
   Secured..................... 1
   Reauthentication Attempts.. 0

   Deleted (Secured).......... 0
   Keepalive Timeouts......... 0

CA Statistics
   Pairwise CAKs Derived...... 0
   Pairwise CAK Rekeys........ 0
   Group CAKs Generated....... 0
   Group CAKs Received........ 0

SA Statistics
   SAKs Generated............. 1
   SAKs Rekeyed............... 0
   SAKs Received.............. 0
   SAK Responses Received..... 1

MKPDU Statistics
   MKPDUs Validated & Rx...... 89589
      "Distributed SAK"..... 0
      "Distributed CAK"..... 0
   MKPDUs Transmitted......... 89600
      "Distributed SAK"..... 1
      "Distributed CAK"..... 0

MKA Error Counter Totals
========================
Session Failures
   Bring-up Failures................ 0
   Reauthentication Failures........ 0
   Duplicate Auth-Mgr Handle........ 0

SAK Failures
```

```
   SAK Generation..................... 0
   Hash Key Generation.............. 0
   SAK Encryption/Wrap.............. 0
   SAK Decryption/Unwrap............ 0
   SAK Cipher Mismatch.............. 0

CA Failures
   Group CAK Generation............. 0
   Group CAK Encryption/Wrap........ 0
   Group CAK Decryption/Unwrap...... 0
   Pairwise CAK Derivation.......... 0
   CKN Derivation................... 0
   ICK Derivation................... 0
   KEK Derivation................... 0
   Invalid Peer MACsec Capability... 0
MACsec Failures
   Rx SC Creation................... 0
   Tx SC Creation................... 0
   Rx SA Installation............... 0
   Tx SA Installation............... 0

MKPDU Failures
   MKPDU Tx......................... 0
   MKPDU Rx Validation.............. 0
   MKPDU Rx Bad Peer MN............. 0
   MKPDU Rx Non-recent Peerlist MN.. 0
```

MACsec 暗号化に関する追加情報

標準および RFC

標準/RFC	タイトル
IEEE 802.1AE-2006	*Media Access Control*（*MAC*）セキュリティ
IEEE 802.1X-2010	ポート ベースのネットワーク アクセス コントロール
IEEE 802.1AEbw-2013	*Media Access Control*（*MAC*）セキュリティ（*IEEE 802.1AE-2006* の修正）：*Extended Packet Numbering*（*XPN*）
IEEE 802.1Xbx-2014	ポートベースのネットワークアクセス コントロール（*IEEE 802.1X-2010* の修正）

標準/RFC	タイトル
RFC 4493	*AES-CMAC* アルゴリズム

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。	Cisco.com の [Support & Downloads] ページ

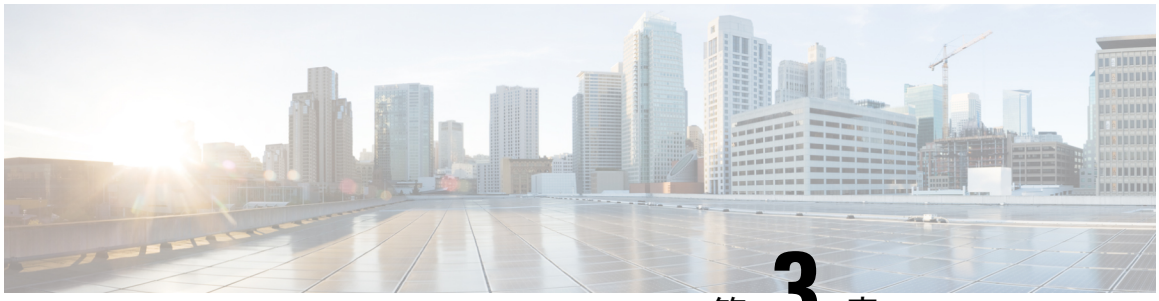
MACsec 暗号化の機能履歴

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 XE 17.13.1	証明書ベースの MACsec 暗号化	この機能のサポートは、このリリースで Cisco Catalyst ESS9300 エンベデッド シリーズ スイッチに導入されました。
Cisco IOS XE Cupertino 17.8.x	MACSec 暗号化	MACsec は 2 台の MACsec 対応デバイス間のパケットの認証と暗号化の IEEE 802.1AE 規格です。 この機能のサポートは、このリリースで Cisco Catalyst IE9300 高耐久性シリーズ スイッチに導入されました。

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

Network Edge Access Topology（NEAT）

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スイッチ

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。認証サーバーがスイッチ ポートに接続する各クライアントを認証したうえで、スイッチまたは LAN が提供するサービスを利用できるようにします。設定情報を含む、802.1x の詳細については、「Configuring IEEE 802.1x Port-Based Authentication」を参照してください。

Network Edge Access Topology（NEAT）機能は、ワイヤリングクローゼット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。NEAT では、サプリカントスイッチとオーセンティケータ間でクライアントの MAC と VLAN 情報を伝播するために Client Information Signalling Protocol（CISP）が使用されます。CISP および NEAT は L2 ポートでのみサポートされ、L3 ポートではサポートされません。Cisco Catalyst IE9300 高耐久性シリーズ スイッチで NEAT を設定できます。

- 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、別のスイッチ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、スイッチがワイヤリング クローゼット外にあり、トランク ポートを介してアップストリーム スイッチに接続される場合に役に立ちます。802.1x スイッチ サプリカント機能を使用して設定されたスイッチは、セキュアな接続のためにアップストリーム スイッチで認証します。サプリカント スイッチが認証に成功すると、オーセンティケータ スイッチでポー

トモードがアクセスからトランクに変更されます。サプリカントスイッチでは、CISP を有効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 スイッチ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 ガードが有効にされたオーセンティケータ スイッチにサプリカントのスイッチを接続する場合、オーセンティケータのポートはサプリカント スイッチ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、errdisable 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 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をブロックします。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 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のスイッチ ポートで有効になっている場合、サプリカント スイッチで **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スイッチで BPDU ガードを有効にした場合、サプリカントスイッチで **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。

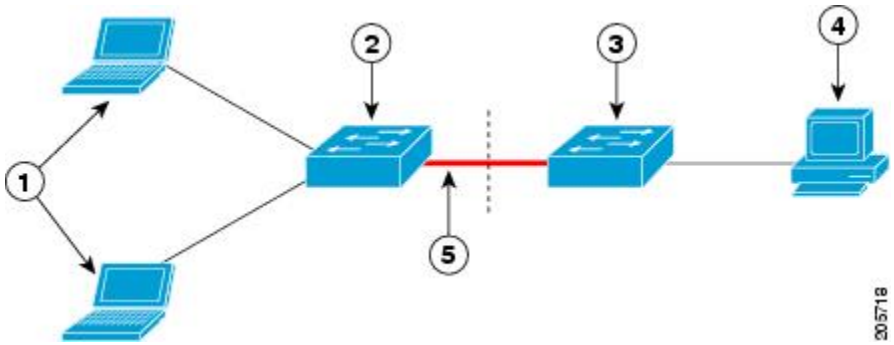
1 つ以上のサプリカントスイッチに接続するオーセンティケータ スイッチ インターフェイスで MDA または multiauth モードを有効にできます。マルチホスト モードはオーセンティケータ スイッチ インターフェイスではサポートされていません。

インターフェイスで有効になっているシングルホスト モードでオーセンティケータ スイッチをリブートすると、インターフェイスが認証前に err-disabled 状態に移行する場合があります。err-disabled 状態から回復するには、オーセンティケータ 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 NEAT のサプリカントスイッチで使用します。

- ホスト許可：許可済み（サプリカントでスイッチに接続する）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。これらのスイッチは、CISP を使用して、サプリカントスイッチに接続する MAC アドレスをオーセンティケータスイッチに送信します。
- 自動有効化：オーセンティケータ スイッチでのトランク コンフィギュレーションを自動的に有効化します。これにより、サプリカント スイッチから着信する複数の VLAN のユーザー 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 *5 : CISP* を使用したオーセンティケータまたはサプリカントスイッチ



1	ワークステーション（クライアント）
2	サプリカント スイッチ（ワイヤリング クローゼット外）
3	オーセンティケータ スイッチ
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 スイッチ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。

注意事項と制約事項

次に、NEAT の設定および使用に関する注意事項と制約事項を示します。

- シスコの Identity Server Engine（ISE）などの RADIUS サーバーが必要です。
- CISP および NEAT は L2 ポートでのみサポートされ、L3 ポートではサポートされません。
- NEAT および 802.1x は、EtherChannel ポートではサポートされません。
- NEAT はダイナミックポートではサポートされません。
- MACsec は NEAT でサポートされます。
- NEAT は PTP と併用できます。

- 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。

NEAT を使用したオーセンティケータスイッチの設定

この機能を設定するには、ワイヤリング クローゼット外の 1 つのスイッチがサプリカントとして設定され、オーセンティケータ スイッチに接続されている必要があります。

（注）
- *cisco-av-pairs* は、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。

スイッチ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。

手順の概要

1. **enable**
2. **configure terminal**
3. **cisp enable**
4. **interface** *interface-id*
5. **switchport mode access**
6. **authentication port-control auto**
7. **dot1x pae authenticator**
8. **spanning-tree portfast**
9. **end**

手順の詳細

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	**cisp enable** 例： `Device(config)# cisp enable`	CISP を有効にします。
ステップ 4	**interface** *interface-id* 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`	インターフェイスを、複数の VLAN のメンバーであるインターフェイスのスパニングツリー フォワーディング ステートにすばやく移行できるようにします。このコマンドは、スイッチ間接続がレイヤ 2 ループの一部ではないことが確実な場合にのみ使用します。
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。

NEAT を使用したサプリカントスイッチの設定

スイッチ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。

手順の概要

1. **enable**
2. **configure terminal**
3. **cisp enable**
4. **eap profile** *profile-name*
5. **method** *type*
6. **exit**
7. **dot1x credentials** *profile*
8. **username** *suppswitch*
9. **password** *password*
10. **dot1x supplicant force-multicast**
11. **interface** *interface-id*
12. **switchport trunk encapsulation dot1q**
13. **switchport mode trunk**
14. **dot1x pae supplicant**
15. **dot1x credentials** *profile-name*
16. **dot1x supplicant eap profile** *profile-name*
17. **end**

手順の詳細

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	**cisp enable** 例：	CISP を有効にします。

	コマンドまたはアクション	目的
	`Device(config)# cisp enable`	
ステップ 4	**eap profile** *profile-name* 例： `Device(config)# eap profile CISP`	Extensible Authentication Protocol（EAP）プロファイルを作成し、EAP 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	**method** *type* 例： `Device(config-eap-profile)# method md5`	EAP 認証方式を指定します。
ステップ 6	**exit** 例： `Device(config-eap-profile)# exit`	EAP プロファイル コンフィギュレーション モードを終了します。
ステップ 7	**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 8	**username** *suppswitch* 例： `Device(config)# username suppswitch`	ユーザ名を作成します。
ステップ 9	**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 10	**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スイッチ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 スイッチで機能できるようにもなります。
ステップ 11	**interface** *interface-id* 例：	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
	`Device(config)# interface gigabitethernet1/0/1`	
ステップ 12	**switchport trunk encapsulation dot1q** 例： `Device(config-if)# switchport trunk encapsulation dot1q`	ポートをトランク モードに設定します。
ステップ 13	**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。
ステップ 14	**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 15	**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 16	**dot1x supplicant eap profile** *profile-name* 例： `Device(config-if)# dot1x supplicant eap profile cisp`	EAP-TLS プロファイルを 802.1X インターフェイスに割り当てます。
ステップ 17	**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。

設定の確認

Client Information Signaling Protocol（CISP）および Network Edge Access Topology（NEAT）の設定に関する情報を確認するには、次の show コマンドを使用します。

- show cisp interface <interface name>

- show cisp clients
- show cisp summary
- show cisp registrations

次に、**show cisp** コマンドの出力例を示します。GigabitEthernet 1/0/1 はオーセンティケータとして設定され、GigabitEthernet 1/0/2 はサプリカントとして設定されます。

```
Auth# show cisp interface Gi1/0/2

CISP Status for interface Gi1/0/2
 -------------------------------------
Version: 1
Mode: Supplicant Peer
Mode: Authenticator
Supp State: Idle

Auth# show cisp clients

Authenticator Client Table:
 ------------------------
 MAC Address VLAN Interface
 ---------------------------------
 0050.5695.4de8 1 Gi1/0/10
6c03.09e7.3947 1 Gi1/0/10
6c03.09e7.3954 11 Gi1/0/10
6c03.09e7.4485 1 Gi1/0/10
 9077.ee4a.8567 1 Gi1/0/10
e41f.7ba1.bbd4 1 Gi1/0/10

Supplicant Client Table:
------------------------
MAC Address VLAN Interface
---------------------------------
9077.ee4a.856b 11 Vl11
9077.ee4a.8572 1 Ap1/1
e41f.7bc7.2f03 1 Gi1/0/9

Auth# show cisp summary

CISP is running on the following interface(s):
---------------------------------------------
Gi1/0/2 (Authenticator)

Supp# show cisp summary

CISP is running on the following interface(s):
---------------------------------------------
Gi1/0/1 (Supplicant)

Auth# show cisp registrations

Interface(s) with CISP registered user(s):
------------------------------------------
Gi1/0/2
Auth Mgr (Authenticator)

Supp# show cisp registration

Interface(s) with CISP registered user(s):
------------------------------------------
```

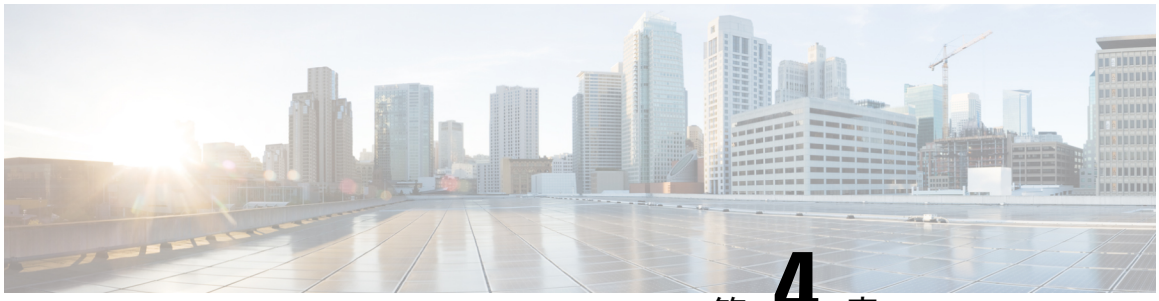
```
Gi1/0/1
802.1x Sup (Supplicant)
```

CISP および NEAT をトラブルシューティングするには、次の debug コマンドを使用します。

- debug access-session errors
- debug access-session event
- debug dot1x errors
- debug dot1x packets
- debug dot1x events

機能の履歴

機能名	リリース	機能情報
Network Edge Access Topology（NEAT）	Cisco IOS XE 17.8.1	Cisco Catalyst IE9300 高耐久性シリーズ スイッチでの最初のサポート



第 4 章

レイヤ 2 ネットワークアドレス変換

レイヤ 2 ネットワークアドレス変換

1 対 1 レイヤ 2 NAT（ネットワークアドレス変換）は、固有のパブリック IP アドレスを既存のプライベート IP アドレス（エンドデバイス）に割り当てるサービスです。この割り当てにより、エンドデバイスがプライベートサブネットおよびパブリックサブネット上で通信できます。このサービスは、NAT 対応デバイスで設定され、エンドデバイスに物理的にプログラムされた IP アドレスのパブリックでの「エイリアス」です。これは、通常 NAT デバイスでテーブルとして表されます。

レイヤ 2 NAT はテーブルを使用して、IPv4 アドレスをパブリックからプライベートおよびプライベートからパブリックの両方にラインレートで変換します。レイヤ 2 NAT は、一貫した高レベルの（bump-in-the-wire）ワイヤスピードの性能を提供するハードウェアベースの機能です。またこの機能は、拡張されたネットワーク セグメンテーション用の NAT 境界で複数の VLAN をサポートします。

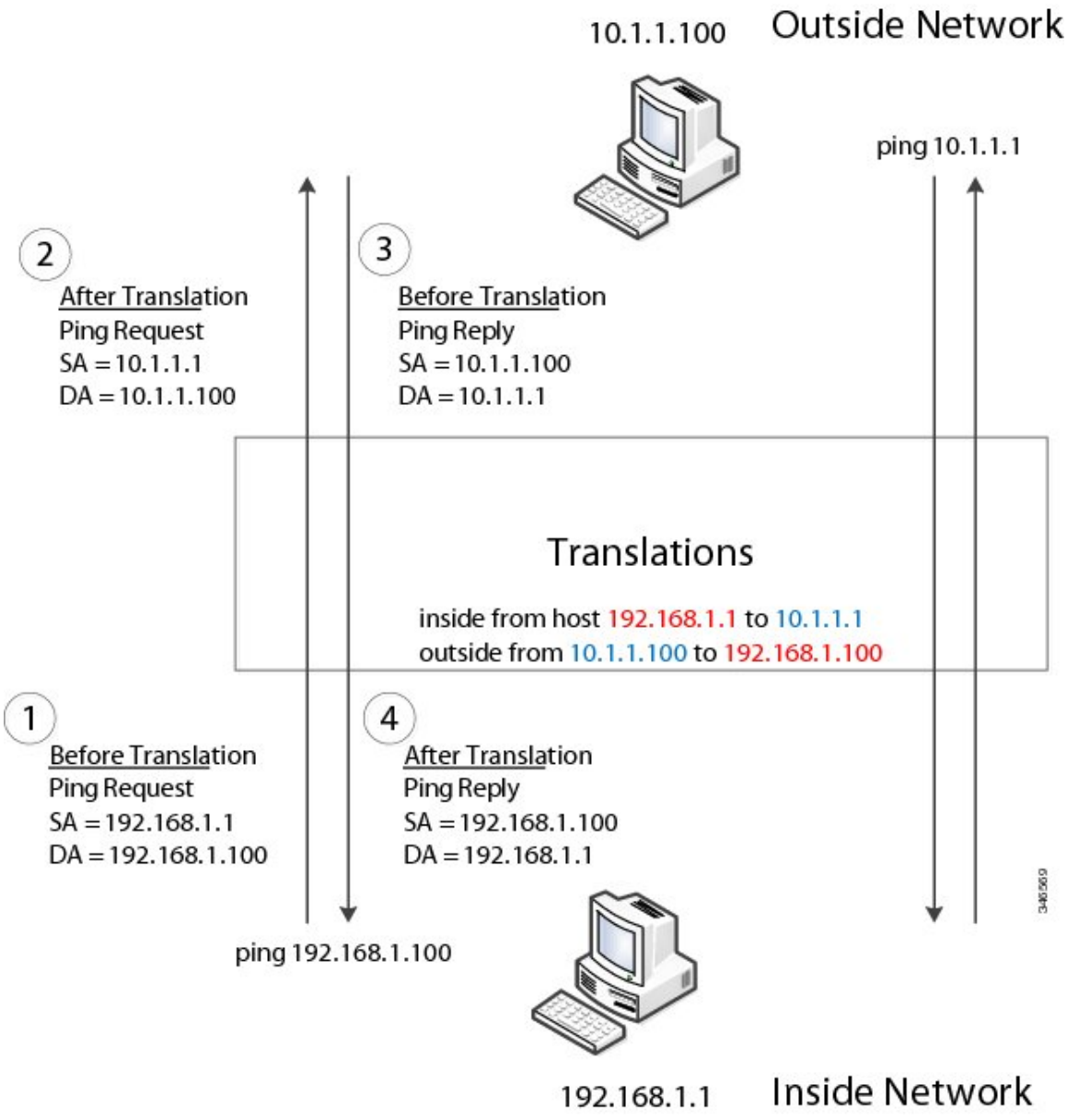
次に、レイヤ 2 NAT で 192.168.1.x ネットワークのセンサーと 10.1.1.x ネットワークの通信制御装置間のアドレスを変換する例を示します。

1. 192.168.1.x ネットワークは内側/内部 IP アドレス空間、10.1.1.x ネットワークは外側または外部 IP アドレス空間です。
2. 192.168.1.1 のセンサーが、「内部」アドレス 192.168.1.100 を使用して通信制御装置に ping 要求を送信します。

レイヤ 2 ネットワークアドレス変換

3. パケットが内部ネットワークから送信される前に、レイヤ 2 NAT は送信元アドレス（SA）を 10.1.1.1 へ、宛先アドレス（DA）を 10.1.1.100 へと変換します。
4. 通信制御装置は 10.1.1.1 へ ping 応答を送信します。
5. パケットが内部ネットワークで受信されると、レイヤ 2 NAT は送信元アドレスを 192.168.1.100 へ、宛先アドレスを 192.168.1.1 へ変換します。

図 *6:* ネットワーク間のアドレス変換

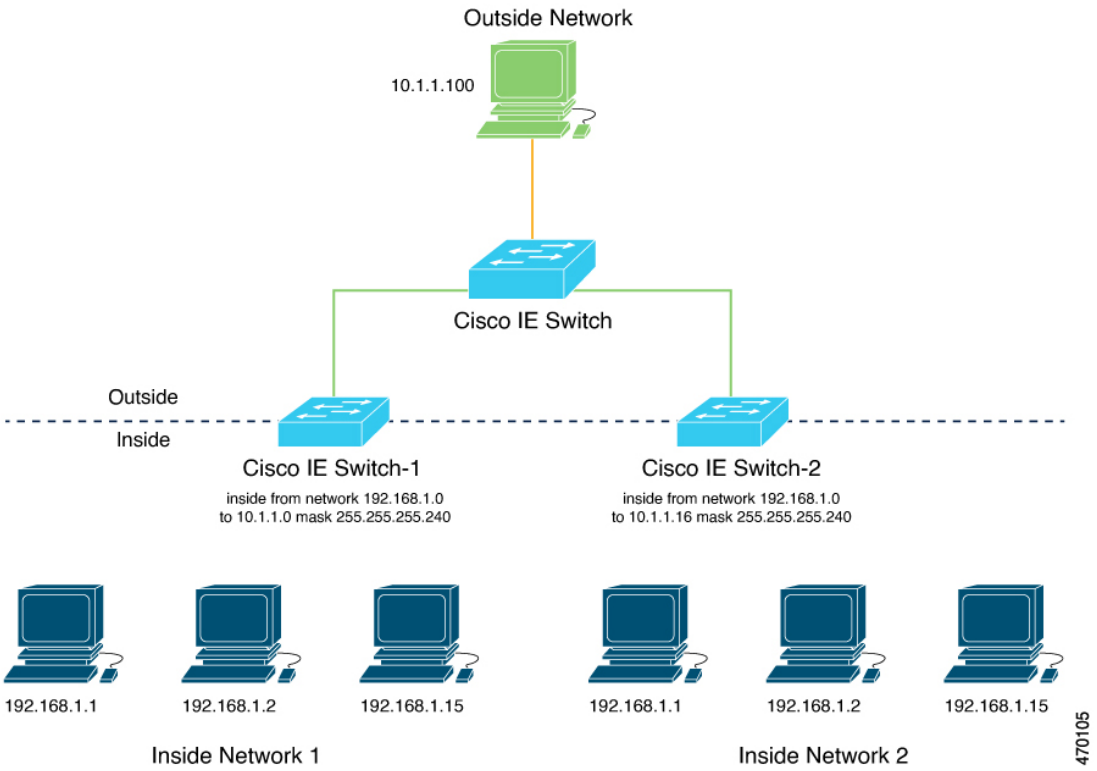


多数のノードに対して、サブネット内のすべてのデバイスの変換をまとめて有効にできます。この場合、内部ネットワーク 1 からのアドレスは 10.1.1.0/28 サブネットで外部アドレスに変換することができ、内部ネットワーク 2 からのアドレスは 10.1.1.16/28 サブネットで外部アドレ

レイヤ 2 ネットワークアドレス変換

スに変換することができます。各サブネットのアドレスはすべて 1 つのコマンドを使って変換できます。サブネットベースの変換を使用すると、レイヤ L2 NAT 規則を節約できます。スイッチには、レイヤ 2 NAT 規則の数に制限があります。サブネットを含む規則では、1 つの規則で複数のエンドデバイスを変換できます。

図 7: 内部-外部アドレス変換



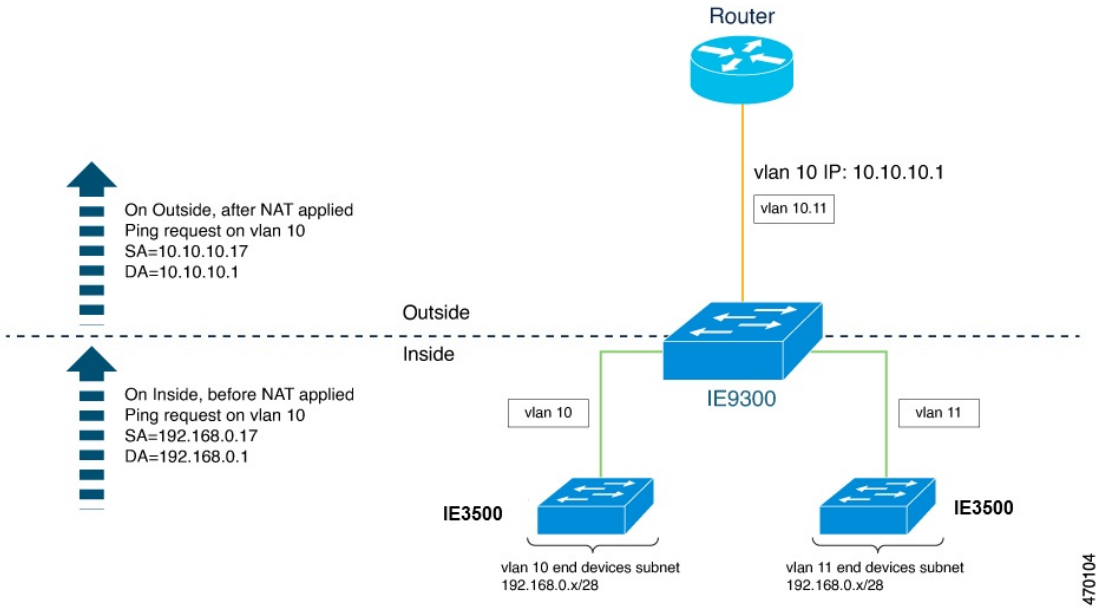
次の図は、レイヤ 2 MAC アドレスに基づいてイーサネットパケットを転送するアグリゲーションレイヤのスイッチを示しています。この例では、ルータはすべてのサブネットと VLAN のレイヤ 3 ゲートウェイです。

L2NAT インスタンス定義では、**network** コマンドを使用して、同じサブネット内の複数のデバイスの変換行を定義します。この場合、IP アドレスの最後のバイトが 16 で始まり 31 で終わる /28 サブネットです。VLAN のゲートウェイは、IP アドレスの最後のバイトが .1 で終わるルータです。外部ホスト変換は、ルータに提供されます。レイヤ 2 NAT 定義の **network** コマンドは、1 つのコマンドでサブネットに相当するホストを変換し、レイヤ 2 NAT 変換レコードを節約します。

Gi1/1 アップリンク インターフェイスには、VLAN 10 および VLAN 11 サブネット用のレイヤ 2 NAT 変換インスタンスがあります。インターフェイスは、複数のレイヤ 2 NAT インスタンス定義をサポートできます。

下流のスイッチは、レイヤ 2 NAT を実行せず上流のアグリゲーション レイヤ スイッチのレイヤ 2 NAT に依存するアクセスレイヤスイッチの例です。

図 *8 :* スイッチの *NAT*



次の例は、上の図の NAT 設定を示しています。

```
!
l2nat instance Subnet10-NAT
 instance-id 1
 permit all
 fixup all
 outside from host 10.10.10.1 to 192.168.0.1
 inside from network 192.168.0.0 to 10.10.10.16 mask 255.255.255.240
!
l2nat instance Subnet11-NAT
 instance-id 1
 permit all
 fixup all
 outside from host 10.10.11.1 to 192.168.0.1
 inside from network 192.168.0.0 to 10.10.11.16 mask 255.255.255.240
!
interface GigabitEthernet1/1
 switchport mode trunk
 l2nat Subnet10-NAT 10
 l2nat Subnet11-NAT 11
!
Interface vlan 1
  ip address 10.10.1.2
```

注意事項と制約事項

次のリストに、スイッチでレイヤ 2 NAT を使用する場合のガイドラインと制限事項を示します。

（注） 規模の詳細については、このガイドの「NATの性能と拡張性（120ページ）」セクションを参照してください。

- レイヤ 2 NAT は、スタンドアロンスイッチでサポートされます。
- レイヤ 2 NAT はデフォルトでは無効です。設定すると有効になります。このガイドの レイヤ 2 NAT の設定 （120 ページ）を参照してください。
- レイヤ 2 NAT はユニキャストトラフィックにのみ適用されます。変換されないユニキャストトラフィック、マルチキャストトラフィック、およびIGMP トラフィックは許可されます。
- レイヤ 2 NAT は、アップリンクポートでのみサポートされ、Network Essentials ライセンスと Network Advantage ライセンスの両方で使用できます。
- レイヤ 2 NAT は、外部 IP アドレスと内部 IP アドレス間の 1 対 1 のマッピングをサポートしています。
- レイヤ 2 NAT は、アクセスモードまたはトランクモードのアップリンク インターフェイスに適用できます。
- レイヤ 2 トラフィックの IPv4 アドレスのみを変換できます。
- 内部ネットワーク変換でサポートされるサブネットマスクは、/24、/25、/26、/27、/28、および /32 のみです。
- 外部変換規則は、ホスト変換のみをサポートします。
- ARP はレイヤ 2 NAT で透過的に機能しません。ただし、スイッチは、IP パケットのペイロードに埋め込まれている IP アドレスを、プロトコルが機能するように変更します。埋め込まれた IP アドレスは変換されません。
- デバッグの統計情報には、各変換のエントリ、各インスタンスおよび各インターフェイスの変換済み入力と出力の合計が含まれます。また、ARP フィックスアップ統計情報と、ハードウェアに割り当てられた変換エントリの数も含まれます。
- レイヤ 2 NAT は、1 対多および多対 1 の IP アドレスのマッピングをサポートしていません。
- パブリックからプライベートへの変換は 1 対 1 であるため、レイヤ 2 NAT ではパブリック IP アドレスを節約できません。1:N NAT ではありません。
- レイヤ 2 NAT のホストの変換を設定する場合は、DHCP クライアントとして設定しないでください。
- レイヤ 2 NAT を使用して内部アドレスを外部アドレスに変換する場合は、変換された IP アドレスがグローバルネットワークでアクセスできないことを確認します。
- 管理インターフェイスはレイヤ2NAT機能の背後にあります。そのためこのインターフェイスはプライベート ネットワーク VLAN 上に置かないようにしてください。プライベー

トネットワーク VLAN 上に存在する場合は、内部アドレスを割り当て、内部の変換を設定します。

- レイヤ 2 NAT は外部アドレスと内部アドレスを分けるように設計されているため、同じサブネットのアドレスを外部アドレスと内部アドレスの両方に設定しないでください。
- レイヤ 2 NAT はレイヤ 2 トラフィック専用です。ルーティング中のパケットには使用しないでください。
- レイヤ 2 NAT は、CPU 宛てのパケットと CPU から送信されるパケットを変換しません。管理トラフィックは、プライベートネットワーク VLAN とは異なる VLAN 上にある必要があります。
- レイヤ 2 NAT カウンタはポートに基づいていません。同じレイヤ 2 NAT インスタンスが複数のインターフェイスに適用されると、対応するレイヤ 2 NAT カウンタがそれらすべてのインターフェイスに表示されます。

NAT の性能と拡張性

レイヤ 2 NAT 変換および転送は、ハードウェアでラインレートで実行されます。サポートされるレイヤ 2 NAT 規則の数は、ハードウェアでサポートできるハードウェアエントリの数によって異なります。

拡張性は、内部/外部の組み合わせの数によって異なります。次に、拡張性の例を示します。

- 内部規則のみを持つインスタンスには、合計 128 個の変換規則を設定できます。
- 1 つの内部規則を持つ複数のインスタンスでは、合計 128 個のインスタンスを 128 個の異なる VLAN に適用できます。
- 1 つの内部規則と 1 つの外部規則を持つ複数のインスタンスには、最大 64 個のインスタンスを含めることができます。
- 1 つの外部規則を持つ 1 つのインスタンスには、最大 100 個の内部規則を設定できます。サポートできる内部規則の数は、外部規則の数が増えると減少します。

（注） 規則の数を節約するために、ネットワーク変換規則を使用することをお勧めします。

レイヤ 2 NAT の設定

アドレス変換を指定するレイヤ 2 NAT インスタンスを設定する必要があります。レイヤ 2 NAT インスタンスを物理イーサネットインターフェイスに接続し、インスタンスを適用する VLAN を設定します。レイヤ 2 NAT インスタンスは、管理インターフェイス（CLI/SNMP）から設定

できます。送受信されたパケットに関する詳細な統計情報を確認できます。このガイドの設定の確認 （123 ページ）セクションを参照してください。

レイヤ 2 NAT を設定するには、次の手順を実行します。詳細については、このガイドで「基本的な内部から外部への通信：例 （125 ページ）」と「重複する IP アドレスの例 （127 ページ）」の例を参照してください。

手順

ステップ 1 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。

configure terminal

ステップ 2 新しいレイヤ 2 NAT インスタンスを作成します。

l2nat instance *instance_name* インスタンスを作成した後、そのインスタンスのサブモードを開始する場合もこのコマンドを使用します。

ステップ 3 内部アドレスを外部アドレスへ変換します。

inside from [*host* | *range* | *network*] *original ip* to *translated ip* [*mask*] *number* | *mask*

単一のホストアドレス、ホストアドレスの範囲、またはサブネット内のすべてのアドレスを変換できます。発信トラフィックの送信元アドレスと着信トラフィックの宛先アドレスを変換します。

ステップ 4 外部アドレスを内部アドレスへ変換します。

outside from [*host* | *range* | *network*] *original ip* to *translated ip* [*mask*] *number* | *mask*

単一のホストアドレス、ホストアドレスの範囲、またはサブネット内のアドレスを変換できます。発信トラフィックの宛先アドレスと着信トラフィックの送信元アドレスを変換します。

ステップ 5 config-l2nat モードを終了します。

exit

ステップ 6 指定したインターフェイス（IE 3400 のアップリンクポートのみ）のインターフェイス コンフィギュレーション モードにアクセスします。

interface *interface-id*

ステップ 7 VLAN または VLAN 範囲に指定されたレイヤ 2 NAT のインスタンスを適用します。このパラメータが欠落している場合、レイヤ 2 NAT インスタンスはネイティブ VLAN に適用されます。

l2nat *instance_name* [*vlan* | *vlan_range*]

ステップ 8 インターフェイス コンフィギュレーション モードを終了します。

end

ポートチャネルでのレイヤ 2 NAT サポートの設定

（注） レイヤ2NATは、ポートチャネルの論理インターフェイスではサポートされますが、メンバーインターフェイスではサポートされません。

LACP は IEEE 802.3ad で定義されており、シスコデバイスが IEEE 802.3ad プロトコルに適合したデバイス間のポートチャネルを管理できるようにします。LACP を使用すると、イーサネットポート間で LACP パケットを交換することにより、ポートチャネルを自動的に作成できます。

スイッチまたはスイッチ スタックはLACP を使用することによって、LACP をサポートできるパートナーの識別情報、および各ポートの機能を学習します。次に、設定が類似しているポートを単一の倫理リンク（チャネルまたは集約ポート）に動的にグループ化します。設定が類似しているポートをグループ化する場合の基準は、ハードウェア、管理、およびポートパラメータ制約です。たとえば、LACP は速度、デュプレックス モード、ネイティブ VLAN、VLAN 範囲、トランキング ステータス、およびトランキング タイプが同じポートをグループとしてまとめます。リンクをまとめてポートチャネルを形成した後で、LACP は単一デバイスポートとして、スパニングツリーにそのグループを追加します。

LACP モードでは、ポートが LACP パケットを送信できるか、LACP パケットの受信のみができるかどうかを指定します。

アクティブモード：ポートをアクティブ ネゴシエーション ステートにします。この場合、ポートは LACP パケットを送信することによって、相手ポートとのネゴシエーションを開始します。

パッシブモード：ポートをパッシブ ネゴシエーション ステートにします。この場合、ポートは受信する LACP パケットに応答しますが、LACP パケットネゴシエーションを開始することはありません。これにより、LACP パケットの送信を最小限に抑えます。

アクティブおよびパッシブLACPモードはともに、相手ポートとネゴシエーションして、ポート速度などの条件に基づいて（レイヤ 2 EtherChannel の場合は、トランク状態および VLAN 番号などの基準に基づいて）、ポートでポートチャネルを形成できるようにします。

ポート チャネルで許可されるバンドル化された LACP ポートの最大数を指定すると、ポートチャネル内の残りのポートがホット スタンバイ ポートとして指定されます。ポートチャネルの LACP ポートの最大数を設定するには、特権 EXEC モードで開始して、次の手順に従います。この手順は任意です。

手順

ステップ 1 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。

device configure

ステップ 2 A-LC という新しいレイヤ 2 NAT インスタンスを作成します。

device # l2nat instance A-LC

ステップ 3 A1 の内部アドレスを外部アドレスへ変換します。

Device(config-l2nat)# inside from host 192.168.1.1 to 10.1.1.1

ステップ 4 A2 の内部アドレスを外部アドレスへ変換します。

Device(config-l2nat)# inside from host 192.168.1.2 to 10.1.1.2

ステップ 5 A3 の内部アドレスを外部アドレスへ変換します。

Device(config-l2nat)# inside from host 192.168.1.3 to 10.1.1.3

ステップ 6 LC の外部アドレスを内部アドレスへ変換します。

Device(config-l2nat)# outside from host 10.1.1.200 to 192.168.1.250

ステップ 7 config-l2nat モードを終了します。

Device(config-l2nat)# exit

ステップ 8 ポートチャネルのインターフェイス設定モードにアクセスします。

Device(config)# interface port-channel

ステップ 9 このインターフェイスのネイティブ VLAN に、先ほどのレイヤ 2 NAT インスタンスを適用します。

Device(config-if)#l2nat A-LC

（注）
トランク上のタグ付き通信の場合は、インターフェイスへインスタンスを適用するときに、次のように VLAN 番号を追加します。`l2nat instance vlan`

ステップ 10 特権 EXEC モードに戻ります。

Device# end

設定の確認

手順

レイヤ 2 NAT 設定を確認するには、次のコマンドを実行します。

コマンド	目的
show l2nat instance	指定されたレイヤ 2 NAT インスタンスの設定の詳細を表示します。

コマンド	目的
show l2nat interface	1 つまたは複数のインターフェイスでのレイヤ 2 NAT インスタンスの設定の詳細を表示します。
show l2nat statistics	すべてのインターフェイスのレイヤ 2 NAT 統計情報を表示します。
show l2nat statistics interface	指定したインターフェイスのレイヤ 2 NAT 統計情報を表示します。
debug l2nat	設定が適用されたときにリアルタイムでのレイヤ 2 NAT 設定の詳細の表示を有効にします。
show platform hardware fed switch 1 fwd-asic resource tcam table pbr record 0 format 0 -	ハードウェアエントリを表示します。
-show platform hardware fed switch active fwd-asic resource tcam utilization | in PBR	ハードウェアリソース使用率を表示します。

次に、**show l2nat instance** および **show l2nat statistics** コマンドの出力例を示します。

```
switch#show l2nat instance
l2nat instance test
fixup  : all
outside from host    10.10.10.200 to 192.168.1.200
inside  from host    192.168.1.1 to 10.10.10.1
l2nat instance test2
fixup  : all
inside  from host    1.1.1.1 to 2.2.2.2
outside from host    2.2.2.200 to 1.1.1.200

Switch#show l2nat interface
FOLLOWING INSTANCE(S) AND VLAN(s) ATTACHED TO ALL INTERFACES
============================================================
l2nat Gi1/1 test
============================================================

Switch#show l2nat statistics

STATS FOR INSTANCE: test (IN PACKETS)

TRANSLATED STATS (IN PACKETS)
================================================================================
INTERFACE DIRECTION VLAN   TRANSLATED
Gi1/1     EGRESS    50    0
Gi1/1     INGRESS   50    0
--------------------------------------------------------------------------------

PROTOCOL FIXUP STATS (IN PACKETS)
================================================================================
INTERFACE DIRECTION VLAN    ARP
Gi1/1     REPLY    50    0
Gi1/1     REQUEST   50    0
--------------------------------------------------------------------------------
```

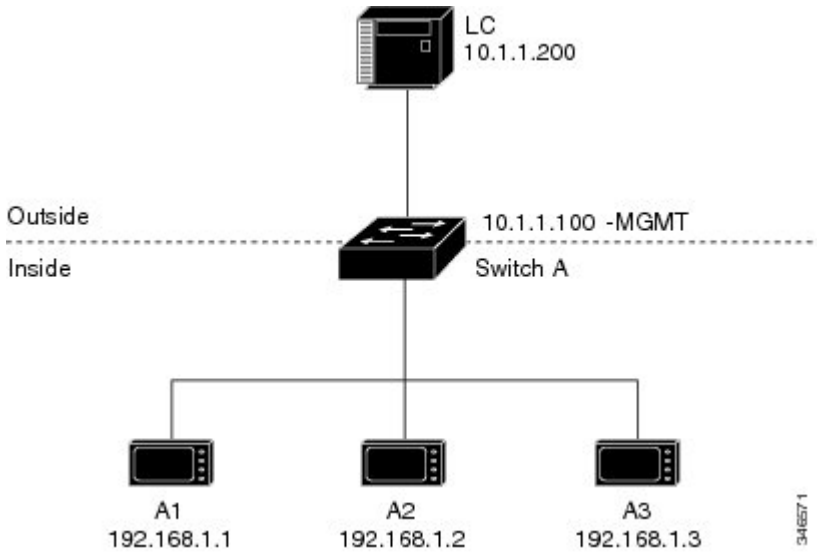
```
PER TRANSLATION STATS (IN PACKETS)
================================================================================
TYPE    DIRECTION SA/DA ORIGINAL IP     TRANSLATED IP   COUNT
OUTSIDE INGRESS   SA    10.10.10.200    192.168.1.200   0
OUTSIDE EGRESS    DA    192.168.1.200   10.10.10.200    0
INSIDE  EGRESS    SA    192.168.1.1     10.10.10.1      0
INSIDE  INGRESS   DA    10.10.10.1      192.168.1.1     0
--------------------------------------------------------------------------------
================================================================================

TOTAL TRANSLATIONS ENTRIES IN HARDWARE: 4
TOTAL INSTANCES ATTACHED : 1
================================================================================
GLOBAL NAT STATISTICS
================================================================================
Total Number of TRANSLATED NAT  Packets   = 0
Total Number of ARP      FIX UP Packets   = 0
================================================================================
ad
```

基本的な内部から外部への通信：例

この例では、A1 はアップリンクポートに直接接続されたロジックコントローラ（LC）と通信する必要があります。レイヤ 2 NAT インスタンスは、外部ネットワーク（10.1.1.1）上での A1 のアドレスと内部ネットワーク（192.168.1.250）上での LC のアドレスを提供するように設定されています。

図 *9*：基本的な内部から外部への通信



ここで次の通信が発生します。

1. A1 が「SA: 192.168.1.1DA: 192.168.1.250」という ARP 要求を送信します。

2. Cisco スイッチ A は「SA:10.1.1.1DA: 10.1.1.200」という ARP 要求をフィックスアップします。
3. LC は要求を受信し、10.1.1.1 の MAC アドレスを学習します。
4. LC が「SA: 10.1.1.200DA: 10.1.1.1」という応答を送信します。
5. Cisco スイッチ A は「SA: 192.168.1.250DA: 192.168.1.1」という ARP 応答をフィックスアップします。
6. A1 は 192.168.1.250 の MAC アドレスを学習し、通信を開始します。

（注）
- スイッチの管理インターフェイスは内部ネットワーク 192.168.1.x. とは別の VLAN に属している必要があります。
- このセクションの例を設定するタスクについては、「基本的な内部から外部への通信：設定 （126 ページ）」セクションを参照してください。

基本的な内部から外部への通信：設定

このセクションでは、前のセクションで説明した内部から外部への通信を設定する手順について説明します。レイヤ 2 NAT インスタンスを作成し、変換エントリを 2 つ追加して、このインスタンスをインターフェイスに適用します。ARP フィックスアップはデフォルトで有効です。

始める前に

「基本的な内部から外部への通信：例 （125 ページ）」セクションの内容を読んで理解してください。

手順

ステップ 1 コンフィギュレーション モードを入力します。

例：

```
switch# configure
```

ステップ 2 A-LC という新しいレイヤ 2 NAT インスタンスを作成します。

例：

```
switch(config)# l2nat instance A-LC
```

ステップ 3 A1 の内部アドレスを外部アドレスへ変換します。

例：

```
switch(config-l2nat)# inside from host 192.168.1.1 to 10.1.1.1
```

ステップ 4 A2 の内部アドレスを外部アドレスへ変換します。

例：

```
switch(config-l2nat)# inside from host 192.168.1.2 to 10.1.1.2
```

ステップ 5 A3 の内部アドレスを外部アドレスへ変換します。

例：

```
switch(config-l2nat)# inside from host 192.168.1.3 to 10.1.1.3
```

ステップ 6 LC 外部アドレスを内部アドレスへ変換します。

例：

```
switch(config-l2nat)# outside from host 10.1.1.200 to 192.168.1.250
```

ステップ 7 config-l2nat モードを終了します。

例：

```
switch(config-l2nat)# exit
```

ステップ 8 アップリンクポートのインターフェイス コンフィギュレーション モードにアクセスします。

例：

```
witch(config)# interface Gi1/1
```

ステップ 9 このインターフェイスのネイティブ VLAN に、先ほどのレイヤ 2 NAT インスタンスを適用します。

例：

```
switch(config-if)# l2nat A-LC
```

（注）
トランク上のタグ付きトラフィックの場合は、インターフェイスへインスタンスを適用するときに、次のように VLAN 番号を追加します。

l2nat *instance vlan*

ステップ 10 特権 EXEC モードに戻ります。

例：

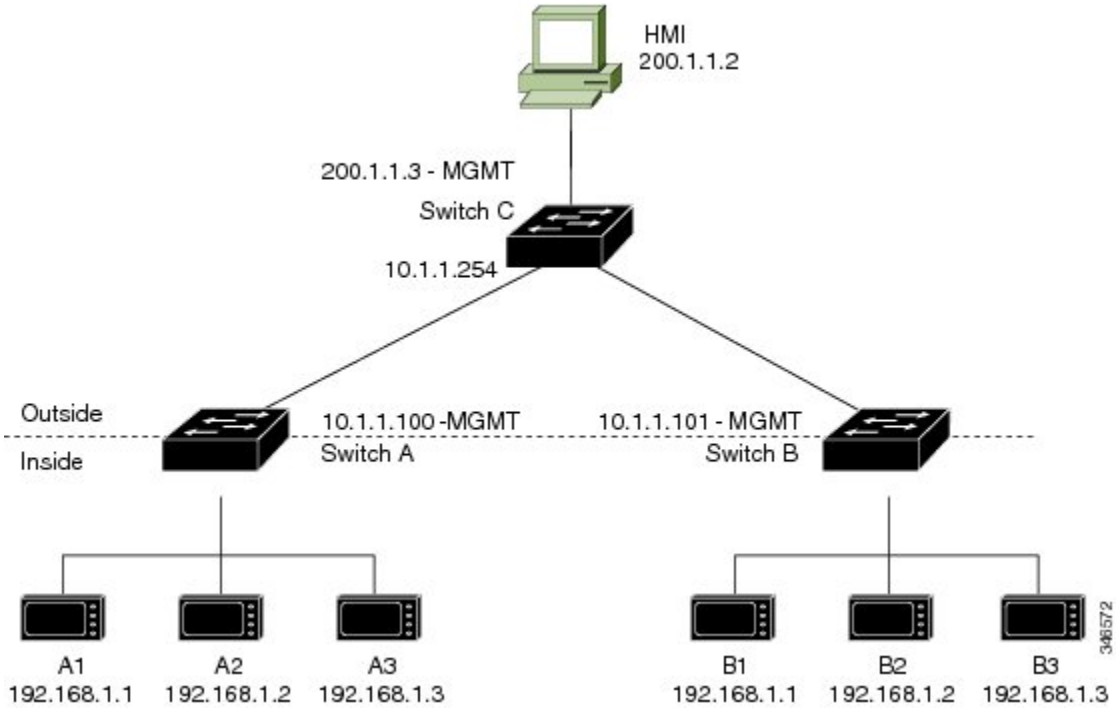
```
switch# end
```

重複する IP アドレスの例

ここでは、2 台のマシンノードで 192.168.1.x 領域のアドレスが事前設定されています。レイヤ 2 NAT により、これらのアドレスが外部ネットワークの別のサブネット上で一意のアドレスに

変換されます。また、マシン間の通信では、ノード A のマシンはノード B の領域で一意のアドレスを必要とし、ノード B のマシンはノード A の領域で一意のアドレスが必要です。

図 *10 : IP* アドレスの重複



- スイッチ C は 192.168.1.x 領域でのアドレスが必要です。パケットがノード A またはノード B で受信されると、スイッチ C の 10.1.1.254 というアドレスが 192.168.1.254 に変換されます。パケットがノード A またはノード B から送信されると、スイッチ C の 192.168.1.254 というアドレスは 10.1.1.254 に変換されます。
- ノード A とノード B のマシンは 10.1.1.x 領域で一意のアドレスが必要です。設定の容易さと使いやすさを実現するために、10.1.1.x 領域は 10.1.1.0、10.1.1.16、10.1.1.32 などのサブネットに分割されます。各サブネットは異なるノードに使用できます。この例では、10.1.1.16 はノード A に使用され、10.1.1.32 はノード B に使用されます。
- ノード A とノード B のマシンはデータを交換するための一意のアドレスが必要です。使用可能なアドレスはサブネットに分割されます。便宜上、ノード A のマシンの 10.1.1.16 サブネットアドレスは、ノード B の 192.168.1.16 サブネットアドレスに変換され、ノード B のマシンの 10.1.1.32 サブネットアドレスはノード A の 192.168.1.32 アドレスに変換されます。
- マシンは各ネットワークで一意のアドレスを持ちます。

表 *10 : IP* アドレスの変換

ノード	ノード A のアドレス	外部ネットワークのアドレス	ノード B のアドレス
スイッチ A のネットワークアドレス	192.168.1.0	10.1.1.16	192.168.1.16
A1	192.168.1.1	10.1.1.17	192.168.1.17
A2	192.168.1.2	10.1.1.18	192.168.1.18
A3	192.168.1.3	10.1.1.19	192.168.1.19
Cisco スイッチ B のネットワークアドレス	192.168.1.32	10.1.1.32	192.168.1.0
B1	192.168.1.33	10.1.1.33	192.168.1.1
B2	192.168.1.34	10.1.1.34	192.168.1.2
B3	192.168.1.35	10.1.1.35	192.168.1.3
スイッチ C	192.168.1.254	10.1.1.254	192.168.1.254

重複する IP アドレスの設定：スイッチ A

このセクションでは、内部ネットワーク内の 1 つのマシンノードの重複 IP アドレスを外部ネットワークのサブネット上の一意のアドレスに変換するようにレイヤ 2 NAT を設定する手順について説明します。この手順は、「重複する IP アドレスの例 （127 ページ）」セクションのスイッチ A を対象としています。

始める前に

「重複する IP アドレスの例 （127 ページ）」セクションの内容を読んで理解してください。

手順

ステップ 1 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。

例：

```
switch# configure
```

ステップ 2 A-Subnet という新しいレイヤ 2 NAT インスタンスを作成します。

例：

```
switch(config)# l2nat instance A-Subnet
```

ステップ 3 ノード A マシンの内部アドレスを 10.1.1.16 255.255.255.240 サブネットのアドレスへ変換します。

例：

```
switch(config-l2nat)# inside from network 192.168.1.0 to 10.1.1.16 mask 255.255.255.240
```

ステップ 4 スイッチ C の外部アドレスを内部アドレスへ変換します。

例：

```
switch(config-l2nat)# outside from host 10.1.1.254 to 192.168.1.254
```

ステップ 5 ノード B マシンの外部アドレスを内部アドレスへ変換します。

例：

```
switch(config-l2nat)# outside from host 10.1.1.32 to 192.168.1.32
outside from host 10.1.1.33 to 192.168.1.33
outside from host 10.1.1.34 to 192.168.1.34
outside from host 10.1.1.35 to 192.168.1.35
```

ステップ 6 config-l2nat モードを終了します。

例：

```
switch(config-l2nat)# exit
```

ステップ 7 アップリンクポートのインターフェイス コンフィギュレーション モードにアクセスします。

例：

```
switch(config)# interface Gi1/1
```

ステップ 8 このインターフェイスのネイティブ VLAN に、先ほどのレイヤ 2 NAT インスタンスを適用します。

例：

```
switch(config-if)# l2nat A-Subnet
```

（注）
トランク上のタグ付きトラフィックの場合は、インターフェイスへインスタンスを適用するときに、次のように VLAN 番号を追加します。

l2nat *instance vlan*

ステップ 9 特権 EXEC モードに戻ります。

例：

```
switch# end
```

次のタスク

「重複する IP アドレスの例 （127 ページ）」セクションのスイッチ B の重複 IP アドレスを変換するようにレイヤ 2 NAT を設定します。重複する IP アドレスの設定：スイッチ B （131 ページ）を参照してください。

重複する IP アドレスの設定：スイッチ B

このセクションでは、内部ネットワーク内の 1 つのマシンノードの重複 IP アドレスを外部ネットワークのサブネット上の一意のアドレスに変換するようにレイヤ 2 NAT を設定する手順について説明します。この手順は、「重複する IP アドレスの例 （127 ページ）」セクションのスイッチ B を対象としています。

始める前に

「重複する IP アドレスの例 （127 ページ）」セクションの内容を読んで理解してください。

手順

ステップ 1 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。

例：

```
switch# configure
```

ステップ 2 B-Subnet という新しいレイヤ 2 NAT インスタンスを作成します。

例：

```
switch(config)# l2nat instance B-Subnet
```

ステップ 3 ノード B マシンの内部アドレスを 10.1.1.32 255.255.255.240 サブネットのアドレスへ変換します。

例：

```
switch(config-l2nat)# inside from network 192.168.1.0 to 10.1.1.32 255.255.255.240
```

ステップ 4 スイッチ C の外部アドレスを内部アドレスへ変換します。

例：

```
switch(config-l2nat)# outside from host 10.1.1.254 to
```

ステップ 5 ノード A マシンの外部アドレスを内部アドレスへ変換します。

例：

```
switch(config-l2nat)# outside from host 10.1.1.16 to 192.168.1.16
outside from host 10.1.1.17 to 192.168.1.17
outside from host 10.1.1.18 to 192.168.1.18
outside from host 10.1.1.19 to 192.168.1.19
```

ステップ 6 config-l2nat モードを終了します。

例：

```
switch(config-l2nat)# exit
```

ステップ 7 アップリンクポートのインターフェイス コンフィギュレーション モードにアクセスします。

例：

```
switch(config)# interface Gi1/1
```

ステップ 8 このインターフェイスのネイティブ VLAN に、先ほどのレイヤ 2 NAT インスタンスを適用します。

例：

```
switch(config-if)# l2nat name1
```

（注）
トランク上のタグ付きトラフィックの場合は、インターフェイスへインスタンスを適用するときに、次のように VLAN 番号を追加します。

l2nat *instance vlan*

ステップ 9 指定されたレイヤ 2 NAT インスタンスの設定の詳細を表示します。

例：

```
switch# show l2nat instance name1
```

ステップ 10 レイヤ 2 NAT の統計情報を表示します。

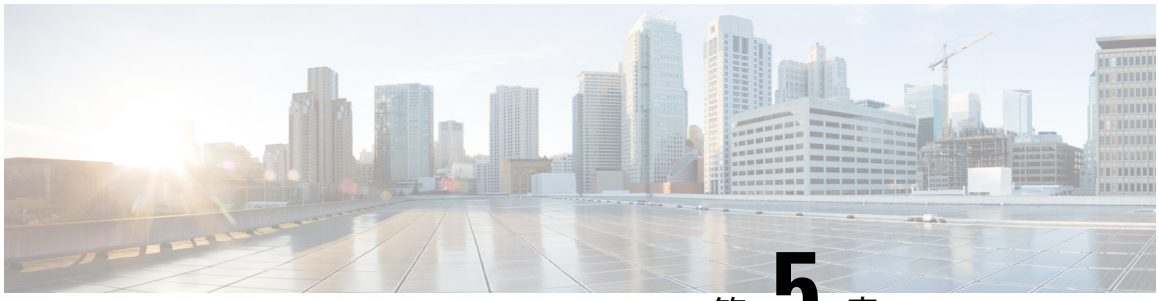
例：

```
switch# show l2nat statistics
```

ステップ 11 特権 EXEC モードに戻ります。

例：

```
switch# end
```



第 5 章

有線ダイナミック PVLAN の設定

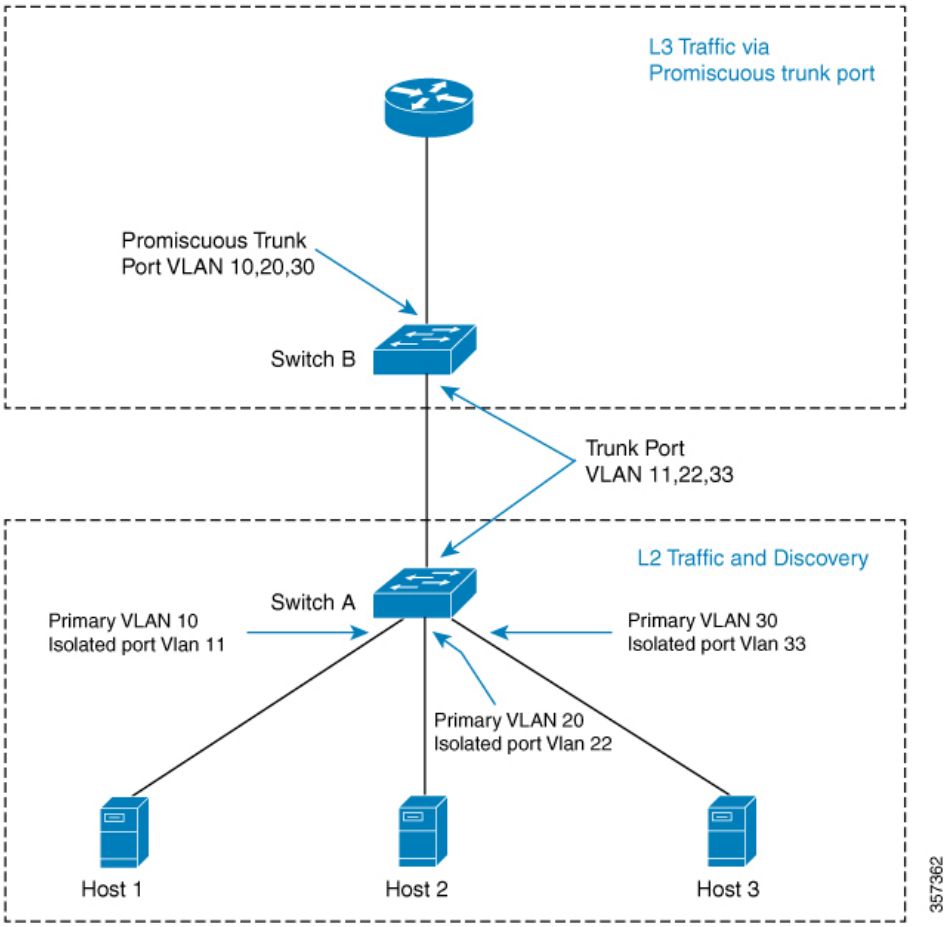
有線ダイナミック PVLAN の制約事項

- 有線ダイナミック PVLAN では、ハイアベイラビリティはサポートされません。
- 音声 VLAN 設定は、この機能と共存できません。
- ローカル We b認証（LWA）および中央 Web 認証（CWA）は、この機能では使用できません。
- ダイナミック PVLANインターフェイステンプレートを使用するすべての有線クライアントは、データクライアントとしてプログラムされます。
- PVLAN テンプレートをサポートするのは、既存のアクセスまたは PVLAN ホストスイッチポートモードのインターフェイスのみです。
- ダイナミックテンプレートのサポートには、Identity Based Networking Services 2.0（IBNS 2.0）を使用する必要があります。

有線ダイナミック PVLAN に関する情報

有線ダイナミック PVLAN は、AAA 許可のあるプライベート VLAN を使用してクライアントを分離し、ゼロトラストを提供する機能です。これは、サブネット/VLAN 内のピアツーピア通信をブロックする方法です。ここで、クライアントは PVLAN に割り当てられ、1 つのポートに接続された有線クライアントをレイヤ 2 の他のすべてのポートから分離し、レイヤ 3 通信は無差別ポートを介して行われます。この機能では、ポイントツーポイントブロッキングを保証するために、ポートインターフェイスごとに単一の有線データクライアントがサポートされます。

（注） 同じインターフェイス上の複数のクライアントからのトラフィックはブロックされません。



このトポロジでは、ホストはスイッチAに接続されており、スイッチの無差別トランクポートとのみ通信できます。PVLANは、中間スイッチを追加することで、複数のスイッチにまたがって拡張できます。上記のトポロジでスイッチ A とスイッチ B の間にスイッチ（スイッチ C）がある場合は、中間リンクにレイヤ 2 トランクポートを設定する必要があります。コミュニティ VLAN の場合、同じコミュニティ VLAN 内の他のホストでパケットを確認できます。

ホストがケーブルでスイッチポートに接続されている場合、そのホストは他のホストを検出できない独立 PVLAN に配置されます。その後、ホストは RADIUS サーバーによって認証されます。もう 1 つのシナリオは、ポートがクローズモードになり、ポートが認証されていない場合、Extensible Authentication Protocol over LAN（EAPoL）パケットのみが許可される場合です。ポートが認証されると、そのポートは独立 VLAN に動的に配置されます。ホストは最初に RADIUS サーバで認証されるため、ホストのポートに適用されるダイナミック インターフェイス テンプレートの名前を送信します。このインターフェイステンプレートには、ポートで PVLAN プライマリ VLAN とセカンダリ VLAN を有効にするための設定が含まれています。テ

ンプレートがホストに適用されると、スイッチポートモードが変更され、ポートがアクセスモードから PVLAN モードにフラップします。

（注） AAA 許可によって参照されるインターフェイステンプレートと同じ名前のものをスイッチで設定する必要があります。

インターフェイステンプレートが適用されると、スティッキータイマーで設定された時間だけポートは物理的にダウンし、再びアップします。RADIUS サーバーがインターフェイステンプレートを 2 回送信すると、変換が完了したため無視されます。その後、ポートは隔離されたままの PVLAN に割り当てられます。ホストは許可を完了し、準備完了状態になります。

access-session interface-template sticky timer *time* コマンドを使用して、インターフェイス テンプレート情報をポートから削除する前に保持するキープタイムを設定します。

有線ダイナミック PVLAN の設定

有線ダイナミック PVLAN を設定するには、ユーザーデバイス（上記のトポロジのスイッチ A）で次の手順を実行します。

始める前に

ユーザーデバイスで dot1x aaa が設定されていることを確認します。

手順の概要

1. **enable**
2. **configure terminal**
3. **vlan** *vlan-id*
4. **private-vlan isolated**
5. **exit**
6. **vlan** *vlan-id*
7. **private-vlan primary**
8. **private-vlan association** [**add** | **remove**] *secondary_vlan_list*
9. **exit**
10. **template** *template-name*
11. **switchport mode private-vlan host**
12. **switchport private-vlan host-association** *primary_vlan_id secondary_vlan_id*
13. **exit**
14. **access-session interface-template sticky timer** *time*
15. **interface** *interface-id*
16. **access-session interface-template sticky timer** *time*
17. **end**

有線ダイナミック PVLAN の設定

手順の詳細

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	**vlan** *vlan-id* 例： `Device(config)# vlan 200`	（任意）VLAN コンフィギュレーション モードを開始して、独立 VLAN となる VLAN を指定または作成します。VLAN ID の範囲は 2 ～ 1001 および 1006 ～ 4094 です。
ステップ 4	**private-vlan isolated** 例： `Device(config-vlan)# private-vlan isolated`	VLAN を独立 VLAN として指定します。
ステップ 5	**exit** 例： `Device(config-vlan)# exit`	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 6	**vlan** *vlan-id* 例： `Device(config)# vlan 100`	VLAN コンフィギュレーション モードを開始して、プライマリ VLAN となる VLAN を指定または作成します。VLAN ID の範囲は 2 ～ 1001 および 1006 ～ 4094 です。
ステップ 7	**private-vlan primary** 例： `Device(config-vlan)# private-vlan primary`	VLAN をプライマリ VLAN として指定します。

	コマンドまたはアクション	目的
ステップ 8	**private-vlan association** [**add** | **remove**] *secondary_vlan_list* 例： `Device(config-vlan)# private-vlan association 200`	セカンダリ VLAN をプライマリ VLAN に関連付けます。単一のプライベート VLAN ID でも、またはハイフンで連結したプライベート VLAN ID でもかまいません。 • *secondary_vlan_list* パラメータには、スペースを含めないでください。カンマで区切った複数の項目を含めることができます。各項目として入力できるのは、単一のプライベート VLAN ID またはハイフンで連結したプライベート VLAN ID です。 • *secondary_vlan_list* パラメータには複数のコミュニティ VLAN ID を含められますが、独立 VLAN ID は 1 つだけです。 • *secondary_vlan_list* を入力するか、または *secondary_vlan_list* で **add** キーワードを指定し、セカンダリ VLAN とプライマリ VLAN を関連付けます。 • セカンダリ VLAN とプライマリ VLAN 間の関連付けをクリアするには、*secondary_vlan_list* に **remove** キーワードを使用します。 • このコマンドは、VLAN コンフィギュレーション モードを終了するまで機能しません。
ステップ 9	**exit** 例： `Device(config-vlan)# exit`	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 10	**template** *template-name* 例： `Device(config)# template PVLAN100_200_CFG`	ユーザーテンプレートを作成し、テンプレート コンフィギュレーション モードを開始します。
ステップ 11	**switchport mode private-vlan host** 例： `Device(config-template)# switchport mode private-vlan host`	レイヤ 2 ポートを PVLAN ホストポートとしてテンプレートに設定します。

	コマンドまたはアクション	目的
ステップ 12	**switchport private-vlan host-association** *primary_vlan_id secondary_vlan_id* 例： `Device(config-template)# switchport private-vlan host-association 100 200`	テンプレートの PVLAN とレイヤ 2 ポートの関連付けを設定します。
ステップ 13	**exit** 例： `Device(config-template)# exit`	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 14	**access-session interface-template sticky timer** *time* 例： `Device(config)# access-session interface-template sticky timer 60`	テンプレートの保持時間をグローバルに設定します。最後のクライアントが離れると、設定された保持時間の後にテンプレートがポートから削除されます。 （注） スティッキータイマーを 60 秒に設定することをお勧めします。
ステップ 15	**interface** *interface-id* 例： `Device(config)# interface GigabitEthernet1/0/1`	インターフェイス設定モードに入り、インターフェイスを指定します。
ステップ 16	**access-session interface-template sticky timer** *time* 例： `Device(config-if)# access-session interface-template sticky timer 60`	インターフェイス上のテンプレートの保持時間を設定します。最後のクライアントが離れると、設定された保持時間の後にテンプレートがポートから削除されます。 （注） スティッキータイマーを 60 秒に設定することをお勧めします。
ステップ 17	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

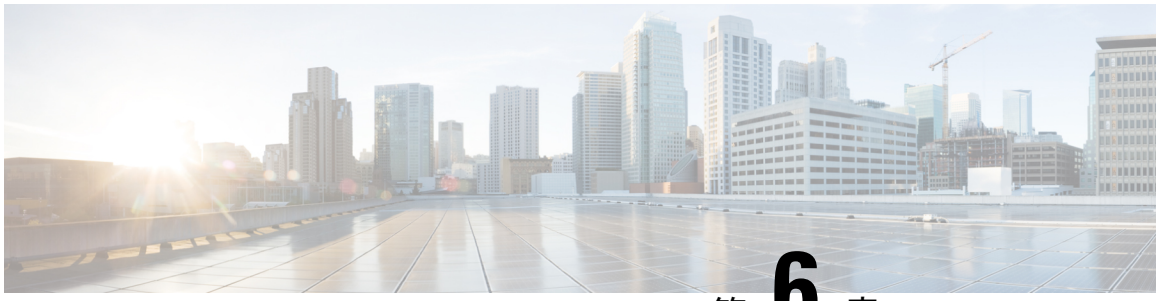
次のタスク

上記の手順の後、Identity Services Engine（ISE）またはその他の RADIUS サーバーを設定して、クライアントが正常に認証された後にクライアントのポートインターフェイスにテンプレートを割り当てます。

ISE を使用している場合、[Policy] > [Policy Elements] > [Authorization] > [Authorization Profile] ページの順にアクセスします。[Interface Template] チェックボックスをオンにして、クライアント インターフェイスに割り当てるテンプレートの名前を入力します。

別の RADIUS サーバーを使用している場合は、最初のクライアント認証が完了した後に、属性 **Cisco-AVpair="interface:template=name"** をスイッチにプッシュする必要があります。

有線ダイナミック PVLAN の設定



第 6 章

IPv4 ACL

IPv4 アクセスコントロールリストの制約事項

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。

- 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。また、VLAN マップでも名前を指定できます。
- 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 を管理ポートに設定することはできません。
- 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。
- プロトコルの TCAM をプログラムしないインターフェイスと、アンロードされた ACL にスケール ACL を適用すると、他のプロトコルのトラフィックの既存の通常移動に影響を与える可能性があります。この制限は、に該当します。
- ルータ ACL は、CPU 生成トラフィックを含むすべてのタイプのトラフィックに適用されます。

- 出力方向の ACL ロギングは、デバイスのコントロールプレーンから生成されたパケットではサポートされません。
- 存続可能時間（TTL）分類は、ACL ではサポートされていません。
- ダウンロード可能な ACL に重複するエントリが含まれている場合、エントリは自動的にマージされません。その結果、802.1X セッション許可は失敗します。ダウンロード可能な ACL が、同じポートのポートベースのエントリや名前ベースのエントリなど、重複するエントリなしで最適化されていることを確認します。
- ソフトウェアによって転送される、注入されたトラフィックでは、出力 ACL ルックアップはサポートされていません。
- ACL は、レイヤ 3 インターフェイス（ルーテッドインターフェイスや VLAN インターフェイスなど）およびサブインターフェイスのみをサポートします。

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。
- VLAN に属しているレイヤ 2 インターフェイスに ACL を適用した場合、レイヤ 2（ポート）ACL は VLAN インターフェイスに適用された入力方向のレイヤ 3 ACL、または VLAN に適用された VLAN マップよりも優先します。
- レイヤ 3 インターフェイスに ACL が適用され、スイッチ上でルーティングがイネーブルになっていない場合は、SNMP、Telnet、Web トラフィックなど、CPU で処理されるパケットだけがフィルタリングされます。
- パケットをフィルタリングするために **preauth_ipv4_acl** ACL が設定されている場合、ACL は認証後に削除されます。
- レイヤ 2 インターフェイスに ACL を適用する場合、ルーティングを有効にする必要はありません。

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

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、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。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、転送するすべてのパケット上で ACL を使用できます。

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、デバイス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、デバイス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。

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。

ACL でサポートされるタイプ

デバイスは、IP ACL とイーサネット（MAC）ACL をサポートしています。

- IP ACL は、TCP、ユーザー 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。

このデバイスは、Quality of Service（QoS）分類 ACL もサポートしています。

サポートされる ACL

スイッチでは、トラフィックをフィルタリングするために、次に示す 3 種類の ACL がサポートされています。

- 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。アクセスリストタイプ（IPv4、IPv6、および MAC）のどの方向に対してでも、レイヤ 2 インターフェイスにポート ACL を適用できます。
- ルータ ACL は、VLAN 間でルーティングされたトラフィックのアクセスを制御し、レイヤ 3 インターフェイスで特定の方向（インバウンドまたはアウトバウンド）に適用されます。

ACL 優先順位

VLAN マップ、ポート ACL、およびルータ ACL が同じスイッチに設定されている場合、入力トラフィックの場合のフィルタの優先順位は上からポート ACL、VLAN マップ、およびルータ ACL です。出力トラフィックの場合、フィルタの優先順位は、ルータ ACL、VLAN マップ、ポート ACL です。

次の例で、簡単な使用例を説明します。

- 入力ポート ACL と VLAN マップが両方とも適用されている場合に、ポート ACL が適用されたポートにパケットが着信すると、このパケットはポート ACL によってフィルタリングされます。その他のパケットは、VLAN マップによってフィルタリングされます。
- スイッチ仮想インターフェイス（SVI）に入力ルータ ACL および入力ポート ACL が設定されている場合に、ポート ACL が適用されているポートにパケットが着信すると、このパケットはポート ACL によってフィルタリングされます。他のポートで受信した着信のルーティング IP パケットには、ルータ ACL のフィルタが適用されます。他のパケットはフィルタリングされません。
- SVI に出力ルータ ACL および入力ポート ACL が設定されている場合に、ポート ACL が適用されているポートにパケットが着信すると、このパケットはポート ACL によってフィルタリングされます。発信するルーティング IP パケットには、ルータ ACL のフィルタが適用されます。他のパケットはフィルタリングされません。

- SVI に VLAN マップ、入力ルータ ACL、および入力ポート ACL が設定されている場合に、ポート ACL が適用されているポートにパケットが着信すると、このパケットはポート ACL だけによってフィルタリングされます。他のポートで受信した着信のルーティング IP パケットには、VLAN マップおよびルータ ACL のフィルタが適用されます。他のパケットには、VLAN マップのフィルタだけが適用されます。
- SVI に VLAN マップ、出力ルータ ACL、および入力ポート ACL が設定されている場合に、ポート ACL が適用されているポートにパケットが着信すると、このパケットはポート ACL だけによってフィルタリングされます。発信するルーティング IP パケットには、VLAN マップおよびルータ ACL のフィルタが適用されます。他のパケットには、VLAN マップのフィルタだけが適用されます。

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 は、物理インターフェイスおよび EtherChannel インターフェイス上でサポートされていますが、EtherChannel メンバーインターフェイスではサポートされていません。ポート ACL は、インバウンド方向とアウトバウンド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 *11 : ACL* によるネットワーク内のトラフィックの制御

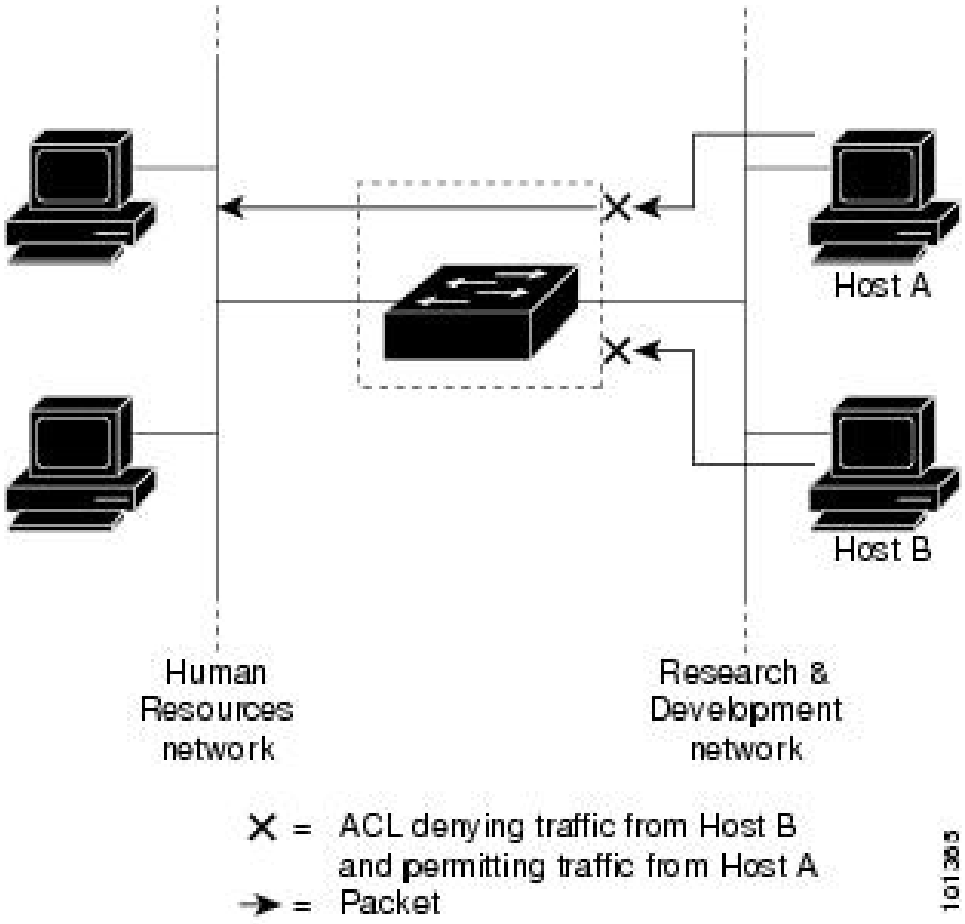


図 *12 : ACL* によるネットワーク内のトラフィックの制御

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。

ルータ ACL

VLAN へのレイヤ 3 インターフェイスであるスイッチ仮想インターフェイス（SVI）、物理層 3 インターフェイス、およびレイヤ 3 EtherChannel インターフェイスに、ルータ ACL を適用できます。ルータ ACL はインターフェイスの特定の方向（着信または発信）に対して適用されます。1 つのインターフェイスの方向ごとに、ルータ ACL を 1 つ適用できます。

スイッチは、IPv4 トラフィックの次のアクセス リストをサポートしています。

- 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 リストは、送信元アドレス、宛先アドレス、およびオプションのプロトコル タイプ情報を使用して一致処理を行います。

ポート ACL の場合と同様、スイッチはインターフェイスに設定されている機能に関連付けられている ACL が照合されます。パケットがスイッチのインターフェイスに着信すると、そのインターフェイスに設定されているすべての着信機能に対応する ACL が照合されます。パケットがルーティングされてからネクストホップに転送されるまでの間に、出力インターフェイスに設定された発信機能に対応するすべての ACL が照合されます。

ACL は ACL 内のエントリとパケットの一致結果に応じて、パケット転送を許可するか、拒否するかを決めます。ACL を使用すると、ネットワーク全体またはネットワークの一部に対するアクセス コントロールが行えます。

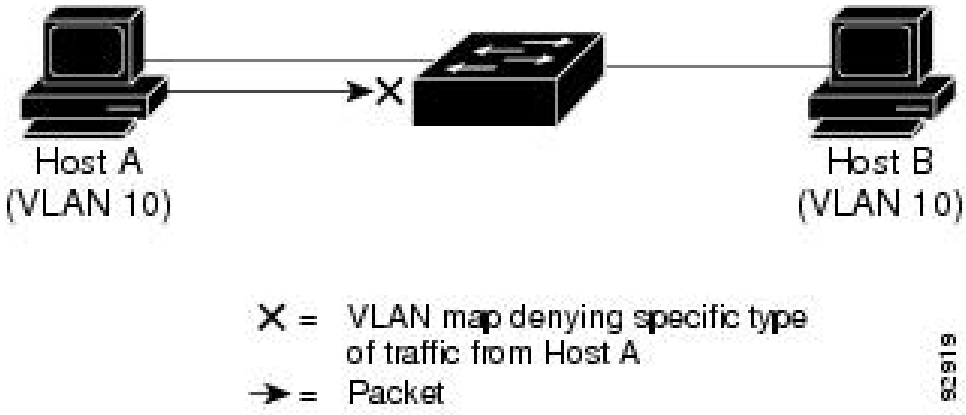
VLAN マップ

VLAN ACL または VLAN マップは、VLAN 内のネットワークトラフィックを制御するために使用されます。スイッチの VLAN 内でブリッジングされるすべてのパケットに VLAN マップを適用できます。VACL は、セキュリティ パケット フィルタリングおよび特定の物理インターフェイスへのトラフィックのリダイレクトだけを目的としたものです。VACL は方向（入力または出力）で定義されることはありません。

すべての非 IP プロトコルは、MAC VLAN マップを使用して、MAC アドレスおよび Ethertype によってアクセス コントロールされます（IP トラフィックは、MAC VLAN マップではアクセス制御されません）。VLAN マップはスイッチを通過するパケットにだけ適用できます。ハブ上またはこのスイッチに接続された別のスイッチ上のホスト間のトラフィックには、VLAN マップを適用できません。

VLAN マップを使用すると、マップに指定されたアクションに基づいてパケットの転送が許可または拒否されます。

図 *13 : VLAN* マップによるトラフィックの制御



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。

標準 IPv4 ACL および拡張 IPv4 ACL

ACLは、許可条件と拒否条件の順序付けられた集まりです。デバイスは、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、デバイスがパ

ケットを受け入れるか拒否するかが決定されます。デバイス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、デバイス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

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
- IP アカウンティング
- 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。

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 *11* : 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

アクセス リスト番号	タイプ	サポートあり
1000 ～ 1099	IPX SAP アクセス リスト	なし
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。

show ip access-list *acl_name* または **show run section** *acl_name* コマンドを使用すると、ACE は、シーケンス番号に従って昇順で表示されます。

作成した番号付き標準 IPv4 ACL を VLAN、端末回線、またはインターフェイスに適用できます。

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。

このデバイスは、ダイナミックまたはリフレクシブ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、このデバイスは以下の 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、デバイス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。
- VLAN マップには、標準 ACL または拡張 ACL（名前付きまたは番号付き）を使用できます。

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、デバイス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） ACL ロギングは、Unicast Reverse Path Forwarding（uRPF）で使用される ACL ではサポートされません。ACL ロギングは、ルータ ACL でのみサポートされ、ポート ACL ではサポートされません。

（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デバイス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。

ハードウェアおよびソフトウェアによる IP ACL の処理

ACL 処理はハードウェアで実行されます。ハードウェアで ACL の設定を保存する領域が不足すると、そのインターフェイス上のすべてのパケットがドロップします。

（注） デバイス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、デバイスに着信した該当 VLAN 内の通信だけです。

ルータ ACL の場合は、次の場合にパケットが CPU に送信されることがあります。

- **log** キーワードの使用
- ICMP 到達不能メッセージを生成する。

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。スイッチドパケットおよびルーテッドパケットに関するハードウェアの ACL の基本的な統計情報を取得する場合は、**show platform software fed switch** { *switch_num* | **active** | **standby** } **acl counters hardware** 特権 EXEC コマンドを使用します。

ルータ 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。
- ルータ 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 が許可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およびルーティングされます。

VLAN マップの設定時の注意事項

VLANマップは、VLAN内でフィルタリングを制御する唯一の方法です。VLANマップには方向の指定がありません。VLANマップを使用して、特定の方向のトラフィックをフィルタリングするには、特定の送信元または宛先アドレスが指定されたACLを追加する必要があります。VLAN マップ内に該当パケット タイプ（IP または MAC）に対する match 句がある場合、デフォルトでは、マップ内のどのエントリにも一致しないパケットはドロップされます。該当パケット タイプに対する match コマンドがない場合、デフォルトでは、パケットが転送されます。

次は、VLAN マップ設定の注意事項です。

- インターフェイスでトラフィックを拒否するように設定された ACL がなく、VLAN マップが設定されていない場合、すべてのトラフィックが許可されます。
- 各 VLAN マップは一連のエントリで構成されます。VLAN マップのエントリの順序は重要です。デバイスに着信したパケットは、VLANマップの最初のエントリに対してテストされます。一致した場合は、VLANマップのその部分に指定されたアクションが実行され

ます。一致しなかった場合、パケットはマップ内の次のエントリに対してテストされます。

- 該当パケット タイプ（IP または MAC）に対する match 句が VLAN マップに 1 つまたは複数ある場合でも、パケットがそれらの match 句に一致しない場合、デフォルトではパケットがドロップされます。該当パケット タイプに対する match 句が VLAN マップ内にない場合、デフォルトではパケットが転送されます。
- VLAN マップのロギングはサポートされていません。
- レイヤ 2 インターフェイスに適用された IP アクセスリストまた MAC アクセスリストがデバイスにあって、ポートが属する VLAN に VLAN マップを適用する場合、ポート ACL が VLAN マップよりも優先されます。
- ハードウェアに VLAN マップの設定を適用できない場合は、その VLAN 内のすべてのパケットがドロップします。

VLAN マップとルータ ACL

ブリッジングされたトラフィックおよびルーティングされたトラフィックの両方に対してアクセス コントロールを行うには、VLAN マップを単独で使用するか、またはルータ ACL と VLAN マップを組み合わせて使用します。入力と出力両方のルーテッド VLAN インターフェイスでルータ ACL を定義したり、ブリッジングされたトラフィックのアクセスをコントロールする VLAN マップを定義したりできます。

パケット フローが ACL 内 VLAN マップの deny ステートメントと一致した場合、ルータ ACL の設定に関係なく、パケット フローは拒否されます。

（注） ルータ ACL を VLAN マップと組み合わせて使用し、ルータ ACL でのロギングを必要とするパケットが VLAN マップで拒否された場合、これらのパケットはロギングされません。

該当パケット タイプ（IP または MAC）に対する match 句が VLAN マップにある場合でも、パケットがそのタイプに一致しない場合、デフォルトではパケットがドロップされます。VLAN マップ内に match 句がなく、アクションが指定されていない場合、どの VLAN マップ エントリとも一致しないパケットは転送されます。

VLAN マップとルータ ACL の設定時の注意事項

ここに記載された注意事項は、ルータ ACL および VLAN マップを同じ VLAN 上で使用する必要がある設定に適用されます。ルータ ACL および VLAN マップを異なる VLAN に割り当てる設定には、これらの注意事項は適用されません。

ルータ ACL および VLAN マップを同じ VLAN に設定する必要がある場合は、ルータ ACL と VLAN マップの両方の設定に関し、ここで説明する注意事項に従ってください。

- VLAN インターフェイス上の各方向（入力および出力）に VLAN マップおよびルータの ACL を 1 つずつに限り設定できます。

- 可能な限り、すべてのエントリのアクションが同一で、末尾のデフォルトアクションだけが反対のタイプとなるように ACL を記述します。次のいずれかの形式を使用して、ACL を記述します。

  - permit...

    permit...

    permit...

    deny ip any any

  または

  - deny...

    deny...

    deny...

    permit ip any any

- ACL 内で複数のアクション（許可、拒否）を定義する場合は、それぞれのアクション タイプをまとめて、エントリ数を削減します。
- ACL 内にレイヤ 4 情報を指定しないでください。レイヤ 4 情報を追加すると、統合プロセスが複雑になります。ACL のフィルタリングが、full-flow（送信元 IP アドレス、宛先 IP アドレス、プロトコル、およびプロトコル ポート）でなく、IP アドレス（送信元および宛先）に基づいて行われる場合に、最適な統合結果が得られます。可能な限り、IP アドレスには *don't care* ビットを使用してください。

  IP ACE とレイヤ 4 情報を含む TCP/UDP/ICMP ACE が両方とも ACL 内に存在し、full-flow モードを指定する必要があるときは、レイヤ 4 ACE をリストの末尾に配置します。この結果、IP アドレスに基づくトラフィックのフィルタリングが優先されます。

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、デバイス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）を使用してデバイス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、パケットの受信後デバイスはパケットを ACL と照合します。ACL がパケットを許可する場合、デバイス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、デバイスはパケットを廃棄します。

アウトバウンド ACL の場合、パケットを受信し制御対象インターフェイスにルーティングした後、デバイスはパケットを ACL と照合します。ACL がパケットを許可した場合は、デバイスはパケットを送信します。ACL がパケットを拒否する場合、デバイスはパケットを廃棄します。

デフォルトでは、パケットが廃棄された場合は、その原因が入力インターフェイスの ACL または発信インターフェイスの ACL のいずれであっても、常に入力インターフェイスから ICMP 到達不能メッセージが送信されます。ICMP 到達不能メッセージは通常、入力インターフェイス 1 つにつき、0.5 秒ごとに 1 つだけ生成されます。ただし、この設定は **ip icmp rate-limit unreachable** 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変更できます。

未定義の ACL をインターフェイスに適用すると、デバイスは 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。

IPv4 アクセスコントロールリストの設定方法

IPv4 ACL の設定

このスイッチで IP ACL を使用する手順は次のとおりです。

手順の概要

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。

2. その ACL をインターフェイスまたは端末回線に適用します。標準および拡張 IP ACL を VLAN マップに適用することもできます。

手順の詳細

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 その ACL をインターフェイスまたは端末回線に適用します。標準および拡張 IP ACL を VLAN マップに適用することもできます。

番号付き標準 ACL の作成

番号付き標準 ACL を作成するには、次の手順に従ってください。

手順の概要

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} *source source-wildcard*]
4. **end**

手順の詳細

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	**access-list** *access-list-number* {**deny** | **permit**} *source source-wildcard*] 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。

	コマンドまたはアクション	目的
		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。 （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。
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。

番号付き拡張 ACL の作成

番号付き拡張 ACL を作成するには、次の手順に従ってください。

手順の概要

1. **enable**
2. **configure terminal**
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*]
4. **access-list** *access-list-number* {**deny** | **permit**} **tcp** *source source-wildcard* [*operator port*] *destination destination-wildcard* [*operator port*] [**established**] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**time-range** *time-range-name*] [**dscp** *dscp*] [*flag*]
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*]

6. **access-list** *access-list-number* {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* | [[*icmp-type icmp-code*] | [*icmp-message*]] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**time-range** *time-range-name*] [**dscp** *dscp*]
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*]
8. **end**

手順の詳細

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	**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： `Device(config)# access-list 101 permit ip host 10.1.1.2 any precedence 0 tos 0 log`	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。 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。 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**。 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）、

	コマンドまたはアクション	目的
		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確認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*] [**established**] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**time-range** *time-range-name*] [**dscp** *dscp*] [*flag*] 例： `Device(config)# access-list 101 permit tcp any any eq 500`	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。 （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。

	コマンドまたはアクション	目的
		• **established**：確立された接続と照合する場合に入力します。このキーワードは、**ack** または **rst** フラグでの照合と同じ機能を果たします。 • *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]] ポート番号またはポート名は、UDP ポートの番号または名前を指定する必要があります。また、UDP では、**flag** キーワードと **established** 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* | [[*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`	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。 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。

	コマンドまたはアクション	目的
		igmp-type：IGMP メッセージタイプと照合するには、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`	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。

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、次の手順に従ってください。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip access-list standard** *name*
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**]
5. **end**

手順の詳細

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** *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。

	コマンドまたはアクション	目的

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`	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 モードに戻ります。

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、次の手順に従ってください。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *name*
4. {**deny** | **permit**} *protocol* {*source* [*source-wildcard*] | **host** *source* | **any**} {*destination* [*destination-wildcard*] | host *destination* | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**] [**time-range** *time-range-name*]
5. **end**

手順の詳細

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：	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	**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*] [**established**] [**log**] [**time-range** *time-range-name*] 例： `Device(config-ext-nacl)# permit 0 any 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。 • **host** *destintation*：宛先および宛先ワイルドカードの値である *destination* 0.0.0.0。 • **any**：送信元および送信元ワイルドカード、または宛先および宛先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-ext-nacl)# end`	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 モードに戻ります。

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 は、インターフェイスまたは VLAN に適用できます。

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。

手順の概要

1. **enable**
2. **configure terminal**
3. **time-range** *time-range-name*
4. 次のいずれかを使用します。
   - **absolute** [**start** *time date*] [**end** *time date*]
   - **periodic** *day-of-the-week hh:mm to* [*day-of-the-week*] *hh:mm*
   - **periodic** {**weekdays** | **weekend** | **daily**} *hh:mm to hh:mm*
5. **end**

手順の詳細

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	**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*]
	• **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`
		適用対象の機能がいつ動作可能になるかを指定します。
		• 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。

	コマンドまたはアクション	目的
ステップ 5	**end** 例： `Device(config-time-range)# end`	時間範囲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

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。

手順の概要

1. **enable**
2. **configure terminal**
3. **line** [**console** | **vty**] *line-number*
4. **access-class** *access-list-number* {**in** | **out**}
5. **end**

手順の詳細

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	**line** [**console**	**vty**] *line-number* 例： `Device(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。

	コマンドまたはアクション	目的
		• **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** | **out**} 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および発信接続を制限します。
ステップ 5	**end** 例： `Device(config-line)# end`	回線コンフィギュレーション モードを終了します。続いて、特権 EXEC モードに戻ります。

インターフェイスへの IPv4 ACL の適用

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管理には、特権EXECモードで次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **ip access-group** {*access-list-number* | *name*} {**in** | **out**}
5. **end**

手順の詳細

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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet1/0/1`	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）またはレイヤ 3 インターフェイス（ルータ ACL）を指定できます。
ステップ 4	**ip access-group** {*access-list-number* | *name*} {**in** | **out**} 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 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。

名前付き MAC 拡張 ACL の作成

VLAN またはレイヤ 2 インターフェイスで非 IPv4 トラフィックをフィルタリングするには、MAC アドレスおよび名前付き MAC 拡張 ACL を使用します。その手順は、他の名前付き拡張 ACL を設定する場合と同様です。

名前付き MAC 拡張 ACL を作成するには、次の手順に従ってください。

手順の概要

1. **enable**
2. **configure terminal**
3. **mac access-list extended** *name*
4. {**deny** | **permit**} {**any** | **host** *source MAC address* | *source MAC address mask*} {**any** | **host** *destination MAC address* | *destination MAC address mask*} [*type mask* | **lsap** *lsap mask* | **aarp** | **amber** | **dec-spanning** | **decnet-iv** | **diagnostic** | **dsm** | **etype-6000** | **etype-8042** | **lat** | **lavc-sca** | **mop-console** | **mop-dump** | **msdos** | **mumps** | **netbios** | **vines-echo** | **vines-ip** | **xns-idp** | 0-*65535*] [**cos** *cos*]
5. **end**

手順の詳細

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	**mac access-list extended** *name* 例： `Device(config)# mac access-list extended mac1`	名前を使用して MAC 拡張アクセス リストを定義します。

ステップ 4	{**deny** | **permit**} {**any** | **host** *source MAC address* | *source MAC address mask*} {**any** | **host** *destination MAC address* | *destination MAC address mask*} [*type mask* | **lsap** *lsap mask* | **aarp** | **amber** | **dec-spanning** | **decnet-iv** | **diagnostic** | **dsm** | **etype-6000** | **etype-8042** | **lat** | **lavc-sca** | **mop-console** | **mop-dump** | **msdos** | **mumps** | **netbios** | **vines-echo** | **vines-ip** | **xns-idp** | 0-*65535*] [**cos** *cos*]
	例：
	`Device(config-ext-macl)# deny any any decnet-iv`
	または
	`Device(config-ext-macl)# permit any any`
		拡張 MAC アクセス リスト コンフィギュレーション モードでは、すべての送信元 MAC アドレス、マスク付き送信元 MAC アドレス、または特定の**host**の送信元 MAC アドレスと、**any**の宛先 MAC アドレス、マスク付き宛先 MAC アドレス、または特定の宛先 MAC アドレスに、**permit** または **deny** を指定します。
		（任意）次のオプションを入力することもできます。
		• *type mask*：Ethernet II または SNAP でカプセル化されたパケットの任意の EtherType 番号。10 進数、16 進数、または 8 進数で表記できます。一致検査の前に、任意で指定できる *don't care* ビットのマスクが EtherType に適用されます。
		• **lsap** *lsap mask*：IEEE 802.2 でカプセル化されたパケットの LSAP 番号。10 進数、16 進数、または 8 進数で表記できます。任意で *don't care* ビットのマスクを指定できます。
		• **aarp** | **amber** | **dec-spanning** | **decnet-iv** | **diagnostic** | **dsm** | **etype-6000** | **etype-8042** | **lat** | **lavc-sca** | **mop-console** | **mop-dump** | **msdos** | **mumps** | **netbios** | **vines-echo** | **vines-ip** | **xns-idp**：非 IP プロトコル。
		• **cos** *cos*：プライオリティを設定する 0 ～ 7 の IEEE 802.1Q CoS 番号。

ステップ 5	**end** 例：	拡張 MAC 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 モードに戻ります。

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

レイヤ 2 インターフェイスへの MAC ACL の適用

レイヤ 2 インターフェイスへのアクセスを制御するために MAC アクセス リストを適用するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **mac access-group** {*name*} {**in** | **out** }
5. **end**
6. **show mac access-group** [**interface** *interface-id*]

手順の詳細

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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet1/0/1`	特定の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。指定するインターフェイスは物理レイヤ 2 インターフェイス（ポート ACL）でなければなりません。
ステップ 4	**mac access-group** {*name*} {**in**	**out** } 例： `Device(config-if)# mac access-group mac1 in`	MAC アクセス リストを使用して、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。 ポート ACL はアウトバウンドおよびインバウンド方向でサポートされます。
ステップ 5	**end** 例：	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。

	コマンドまたはアクション	目的
	`Device(config-if)# end`	
ステップ 6	**show mac access-group** [**interface** *interface-id*] 例： `Device# show mac access-group interface gigabitethernet1/1`	そのインターフェイスまたはすべてのレイヤ 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。

デバイスは、パケットを受信すると、インバウンド ACL とパケットを照合します。ACL がパケットを許可する場合、デバイス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、デバイス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、デバイスは ACL がインターフェイスに適用されていないものとして、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。

テンプレートモードでの IPv4 ACL の設定

（注） **ip access-group** コマンドはテンプレート コンフィギュレーション モードで設定できます。**source template** コマンドは、インターフェイスに対して 1 回だけ設定できます。

ACL をテンプレートで設定するには、特権 EXEC モードで次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *name*
4. **ip access-list extended** {*name*|*number*|*ext_number*}
5. **exit**
6. **template**
7. **ip access-group** {*access-list-number* | *name*} {**in** | **out**}
8. **exit**
9. **interface** *interface-id*
10. **ip access-group** {*access-list-number* | *name*} {**in** | **out**}
11. **source template** *name*
12. **end**

手順の詳細

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 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。 *name* を入力して、アクセスリスト名を定義します。 *number* を入力して、拡張 IP アクセスリスト番号を定義します。有効な範囲は 100 ～ 199 です。 *ext_number* を入力して、拡張 IP アクセスリスト番号を定義します。拡張範囲は 2000 ～ 2699 です。
ステップ 4	**ip access-list extended** {*name*|*number*|*ext_number*} 例： `Device(config)# ip access-list extended 151`	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。 *name* を入力して、アクセスリスト名を定義します。 *number* を入力して、拡張 IP アクセスリスト番号を定義します。有効な範囲は 100 ～ 199 です。 *ext_number* を入力して、拡張 IP アクセスリスト番号を定義します。拡張範囲は 2000 ～ 2699 です。
ステップ 5	**exit** 例： `Device(config-ext-nacl)# exit`	アクセス リスト コンフィギュレーション モードを終了します。
ステップ 6	**template** 例： `Device# template test`	ユーザーテンプレートを作成し、テンプレート コンフィギュレーション モードを開始します。
ステップ 7	**ip access-group** {*access-list-number* | *name*} {**in** | **out**} 例：	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。

	コマンドまたはアクション	目的
	`Device(config-template)# ip access-group 150 in`	*access-list-number* を入力して、アクセスリストを定義します。アクセスリストには番号を指定できます。 *name* を入力して、アクセスリストを定義します。アクセスリストには名前を指定できます。 **in** を入力して、インターフェイスの着信方向にアクセスリストを送信します。 **out** を入力して、インターフェイスの発信方向にアクセスリストを送信します。
ステップ 8	**exit** 例： `Device(config-template)# exit`	テンプレートの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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet1/0/1`	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）またはレイヤ 3 インターフェイス（ルータ ACL）を指定できます。
ステップ 10	**ip access-group** {*access-list-number* | *name*} {**in** | **out**} 例： `Device(config-if)# ip access-group 151 out`	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。 *access-list-number* を入力して、アクセスリストを定義します。アクセスリストには番号を指定できます。 *name* を入力して、アクセスリストを定義します。アクセスリストには名前を指定できます。 **in** を入力して、インターフェイスの着信方向にアクセスリストを送信します。 **out** を入力して、インターフェイスの発信方向にアクセスリストを送信します。
ステップ 11	**source template** *name* 例： `Device(config)# source template test`	インターフェイス テンプレートをターゲットに適用します。アクセスリスト 150 は、設定されている着信アクセスリストです。
ステップ 12	**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。

VLAN マップの設定

VLANマップを作成して、1つまたは複数のVLANに適用するには、次のステップを実行します。

始める前に

VLAN に適用する標準 IPv4 ACL または拡張 IP ACL、または名前付き MAC 拡張 ACL を作成します。

手順の概要

1. **enable**
2. **configure terminal**
3. **vlan access-map** *name* [**number**]
4. **match** {**ip** | **mac**} **address** {*name* | *number*} [*name* | *number*]
5. IP パケットまたは非 IP パケットを（既知の 1 MAC アドレスのみを使って）指定し、1 つ以上の ACL（標準または拡張）とそのパケットを照合するには、次のコマンドのいずれかを入力します。
   - **action** { **forward**}

     ```
     Device(config-access-map)# action forward
     ```

   - **action** { **drop**}

     ```
     Device(config-access-map)# action drop
     ```

6. **exit**
7. **vlan filter** *mapname* **vlan-list** *list*
8. **end**

手順の詳細

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	**vlan access-map** *name* [**number**] 例： `Device(config)# vlan access-map map1 20`	VLANマップを作成し、名前と、任意で番号を付けます。番号は、マップ内のエントリのシーケンス番号です。

	コマンドまたはアクション	目的
		同じ名前の VLAN マップを作成すると、10 ずつ増加する番号が順に割り当てられます。マップを変更または削除するときは、該当するマップエントリの番号を入力できます。 VLAN マップでは、特定の permit または deny キーワードを使用しません。VLAN マップを使用してパケットを拒否するには、パケットを照合する ACL を作成して、アクションをドロップに設定します。ACL 内の permit は、一致するという意味です。ACL 内の deny は、一致しないという意味です。 このコマンドを入力すると、アクセス マップ コンフィギュレーション モードに変わります。
ステップ 4	**match** {**ip** | **mac**} **address** {*name* | *number*} [*name* | *number*] 例： `Device(config-access-map)# match ip address ip2`	1 つまたは複数の標準または拡張アクセス リストに対してパケットを照合します（IP または MAC アドレスを使用）。パケットの照合は、対応するプロトコル タイプのアクセス リストに対してだけ行われます。IP パケットは、標準または拡張 IP アクセス リストに対して照合されます。非 IP パケットは、名前付き MAC 拡張アクセス リストに対してだけ照合されます。 （注） パケット タイプ（IP または MAC）に対する match 句が VLAN マップに設定されている場合で、そのマップ アクションがドロップの場合は、そのタイプに一致するすべてのパケットがドロップされます。match 句が VLAN マップになく、設定されているアクションがドロップの場合は、すべての IP およびレイヤ 2 パケットがドロップされます。

ステップ 5	IP パケットまたは非 IP パケットを（既知の 1 MAC アドレスのみを使って）指定し、1 つ以上の ACL（標準または拡張）とそのパケットを照合するには、次のコマンドのいずれかを入力します。
	
	• **action** { **forward**}
	`Device(config-access-map)# action forward`
	
	• **action** { **drop**}
	`Device(config-access-map)# action drop`
		マップエントリに対するアクションを設定します。

	コマンドまたはアクション	目的
ステップ 6	**exit** 例： `Device(config-access-map)# exit`	アクセスマップ 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	**vlan filter** *mapname* **vlan-list** *list* 例： `Device(config)# vlan filter map1 vlan-list 20-22`	VLAN マップを 1 つまたは複数の VLAN に適用します。 list には単一の VLAN ID（22）、連続した範囲（10 ～ 22）、または VLAN ID のストリング（12、22、30）を指定できます。カンマやハイフンの前後にスペースを挿入することもできます。
ステップ 8	**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。

VLAN への VLAN マップの適用

VLAN マップを 1 つまたは複数の VLAN に適用するには、次の手順に従います。

手順の概要

1. **enable**
2. **configure terminal**
3. **vlan filter** *mapname* **vlan-list** *list*
4. **end**

手順の詳細

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	**vlan filter** *mapname* **vlan-list** *list* 例：	VLAN マップを 1 つまたは複数の VLAN に適用します。

	コマンドまたはアクション	目的
	`Device(config)# vlan filter map 1 vlan-list 20-22`	list には単一の VLAN ID（22）、連続した範囲（10 ～ 22）、または VLAN ID のストリング（12、22、30）を指定できます。カンマやハイフンの前後にスペースを挿入することもできます。
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。

IPv4 ACL のモニタリング

デバイスに設定されている ACL、およびインターフェイスと VLAN に適用された ACL を表示して IPv4 ACL をモニターできます。

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 *12:* 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

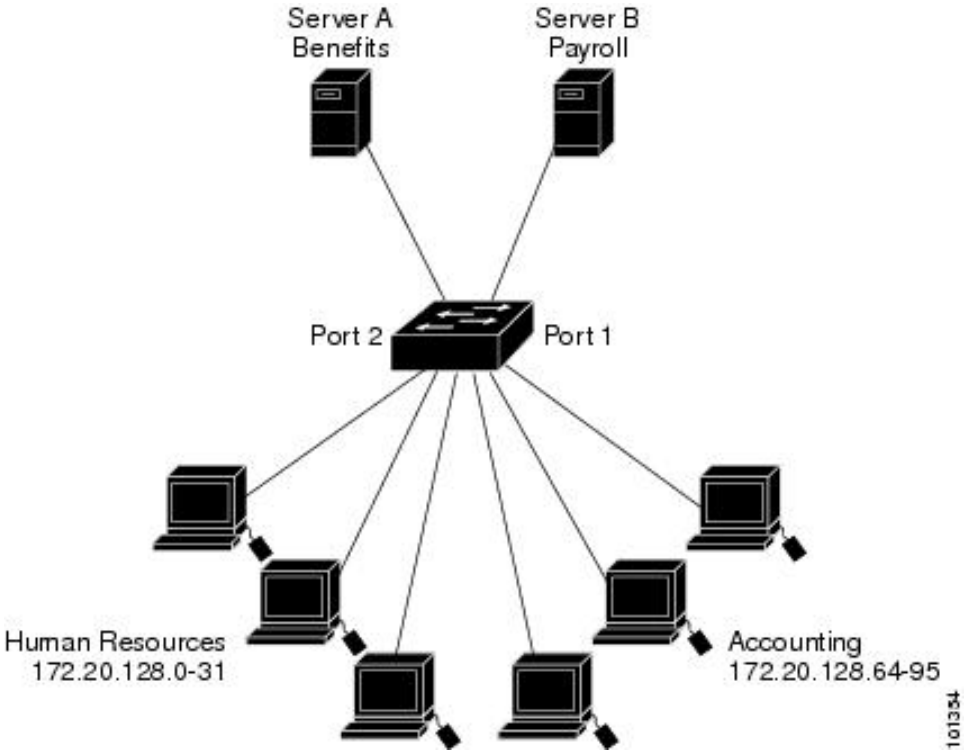
コマンド	目的
show access-lists [*number* | *name*]	最新の IP および MAC アドレス アクセス リストの全体やその一部、または特定のアクセス リスト（番号付きまたは名前付き）の内容を表示します。
show ip access-lists [*number* | *name*]	最新の IP アクセス リスト全体、または特定の IP アクセス リスト（番号付きまたは名前付き）を表示します。
show ip interface *interface-id*	インターフェイスの詳細設定およびステータスを表示します。IP がイネーブルになっているインターフェイスに、**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*]	デバイス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。

IPv4 アクセスコントロールリストの設定例

小規模ネットワークが構築されたオフィス用の ACL

図 *14 :* ルータ *ACL* によるトラフィックの制御



ルータ ACL を使用して上記のように設定するには、次のいずれかの方法を使用します。

- 標準 ACL を作成し、ポート 1 からサーバーに着信するトラフィックをフィルタリングします。
- 拡張 ACL を作成し、サーバーからポート 1 に着信するトラフィックをフィルタリングします。

例：小規模ネットワークが構築されたオフィスの ACL

次に、標準 ACL を使用してポートからサーバー B に着信するトラフィックをフィルタリングし、経理部の送信元アドレス 172.20.128.64 ～ 172.20.128.95 から送信されるトラフィックだけを許可する例を示します。この ACL は、指定された送信元アドレスを持つルーテッド ポート 1 から送信されるトラフィックに適用されます。

```
Device> enable
Device# configure terminal
Device(config)# access-list 6 permit 172.20.128.64 0.0.0.31
Device(config)# exit
Device# show access-lists

Standard IP access list 6
    10 permit 172.20.128.64, wildcard bits 0.0.0.31

Device# configure terminal
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group 6 out
Device(config-if)# end
```

次に、拡張 ACL を使用してサーバー B からポートに着信するトラフィックをフィルタリングし、任意の送信元アドレス（この場合はサーバー B）から経理部の宛先アドレス 172.20.128.64 ～ 172.20.128.95 に送信されるトラフィックだけを許可する例を示します。この ACL は、ルーテッドポート 1 に着信するトラフィックに適用され、指定の宛先アドレスに送信されるトラフィックだけを許可します。拡張 ACL を使用する場合は、送信元および宛先情報の前に、プロトコル（IP）を入力する必要があります。

```
Device(config)# access-list 106 permit ip any 172.20.128.64 0.0.0.31
Device(config)# exit
Device# show access-lists

Extended IP access list 106
    10 permit ip any 172.20.128.64 0.0.0.31


Device# configure terminal
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group 106 in
Device(config-if)# end
```

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> enable
Device# configure terminal
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 gigabitethernet1/0/1
Device(config-if)# ip access-group 2 in
Device(config-if)# end
```

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 行目で、ホスト 172.16.0.0 の Simple Mail Transfer Protocol（SMTP）ポートへの着信 TCP 接続を許可しています。3 番目の行は、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> enable
Device# configure terminal
Device(config)# access-list 102 permit tcp any 172.16.0.0 0.0.255.255 gt 1023
Device(config)# access-list 102 permit tcp any host 172.16.1.2 eq 25
Device(config)# access-list 102 permit icmp any any
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group 102 in
Device(config-if)# end
```

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 システムでは常にポート 25 でのメール接続が使用されているため、着信サービスは個別に制御されます。

```
Device> enable
Device# configure terminal
Device(config)# access-list 102 permit tcp any 172.16.0.0 0.0.255.255 eq 23
Device(config)# access-list 102 permit tcp any 172.16.0.0 0.0.255.255 eq 25
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group 102 in
Device(config-if)# end
```

次の例では、ネットワークはアドレスが 172.16.0.0 のクラス B ネットワークで、メールホストのアドレスは 172.16.1.2 です。**established** キーワードは、確立された接続を表示する TCP 専用のキーワードです。TCP データグラムに ACK または RST ビットが設定され、パケットが既存の接続に属していることが判明すると、一致と見なされます。ギガビット イーサネット インターフェイス 1 は、デバイスをインターネットに接続するインターフェイスです。

```
Device> enable
Device# configure terminal
Device(config)# access-list 102 permit tcp any 172.16.0.0 0.0.255.255 established
Device(config)# access-list 102 permit tcp any host 172.16.1.2 eq 25
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip access-group 102 in
Device(config-if)# end
```

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 10.2.3.4 から送信されるすべてのトラフィックを許可します。

```
Device> enable
Device# configure terminal
Device(config)# ip access-list standard Internet_filter
Device(config-ext-nacl)# permit 10.2.3.4
Device(config-ext-nacl)# exit
Device(config-ext-nacl)# end
```

marketing_group ACL は、宛先アドレスとワイルドカードの値 172.16.0.0 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 172.16.0.0 ～ 172.16.255.255 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> enable
Device# configure terminal
Device(config)# ip access-list extended marketing_group
Device(config-ext-nacl)# permit tcp any 172.16.0.0 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 172.16.0.0 0.0.255.255 lt 1024
Device(config-ext-nacl)# deny ip any any log
Device(config-ext-nacl)# end
```

Internet_filter ACL は発信トラフィックに適用され、*marketing_group* ACL はレイヤ 3 ポートの着信トラフィックに適用されます。

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# no switchport
Device(config-if)# ip address 10.0.5.1 255.255.255.0
Device(config-if)# ip access-group Internet_filter out
Device(config-if)# ip access-group marketing_group in
Device(config-if)# end
```

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> enable
Device# configure terminal
Device(config)# ip access-list extended border-list
Device(config-ext-nacl)# no permit ip host 10.1.1.3 any
Device(config-ext-nacl)# end
```

例：ACL ロギング

ルータ 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> enable
Device# configure terminal
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 gigabitethernet1/0/1

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> enable
Device# configure terminal
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 gigabitethernet1/0/1

Device(config-if)# ip access-group ext1 in
Device(config)# end
```

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：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> enable
Device# configure terminal
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
Device(config)# end
```

（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。

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> enable
Device# configure terminal
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)# exit
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> enable
Device# configure terminal
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> enable
Device# configure terminal
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 gigabitethernet1/0/1

Device(config-if)# ip access-group strict in
Device(config-if)# end
```

例：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。

次の例では、user1 のワークステーションにはアクセスを許可し、user2 のワークステーションにはアクセスを許可しません。

```
Device> enable
Device# configure terminal
Device(config)# access-list 1 remark Permit only user1 workstation through
Device(config)# access-list 1 permit 171.69.2.88
Device(config)# access-list 1 remark Do not allow user2 through
Device(config)# access-list 1 deny 171.69.3.13
Device(config)# end
```

名前付き IP ACL のエントリには、**remark** アクセスリスト 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。

次の例では、サブネット subnet1 にはアウトバウンド Telnet の使用が許可されません。

```
Device> enable
Device# configure terminal
Device(config)# ip access-list extended telnetting
Device(config-ext-nacl)# remark Do not allow subnet1 subnet to telnet out
Device(config-ext-nacl)# deny tcp host 171.69.2.88 any eq telnet
Device(config-ext-nacl)# end
```

例：パケットを拒否する ACL および VLAN マップの作成

ここでは、パケットを拒否する ACL および VLAN マップを作成する例を示します。最初のマップでは、*ip1* ACL（TCP パケット）に一致するすべてのパケットがドロップされます。最初に、すべての TCP パケットを許可し、それ以外のパケットをすべて拒否する *ip1* ACL を作成します。VLAN マップには IP パケットに対する match 句が存在するため、デフォルトのアクションでは、どの match 句とも一致しない IP パケットがすべてドロップされます。

```
Device> enable
Device# configure terminal
Device(config)# ip access-list extended ip1
Device(config-ext-nacl)# permit tcp any any
Device(config-ext-nacl)# exit
Device(config)# vlan access-map map_1 10
Device(config-access-map)# match ip address ip1
Device(config-access-map)# action drop
Device(config-access-map)# end
```

例：パケットを許可する ACL および VLAN マップの作成

次に、パケットを許可する VLAN マップを作成する例を示します。ACL *ip2* は UDP パケットを許可し、*ip2* ACL と一致するすべてのパケットが転送されます。このマップでは、これ以前のどの ACL とも一致しないすべての IP パケット（TCP でも UDP でもないパケット）がドロップされます。

```
Device> enable
Device# configure terminal
Device(config)# ip access-list extended ip2
Device(config-ext-nacl)# permit udp any any
Device(config-ext-nacl)# exit
Device(config)# vlan access-map map_1 20
Device(config-access-map)# match ip address ip2
Device(config-access-map)# action forward
Device(config-access-map)# exit
```

例：IP パケットのドロップおよび MAC パケットの転送のデフォルトアクション

次の例の VLAN マップでは、デフォルトで IP パケットがドロップされ、MAC パケットが転送されます。標準の ACL 101 および名前付き拡張アクセス リスト **igmp-match** および **tcp-match** をこのマップと組み合わせて使用すると、次のようになります。

- すべての UDP パケットが転送されます。

- すべての IGMP パケットがドロップされます。
- すべての TCP パケットが転送されます。
- その他のすべての IP パケットがドロップされます。
- すべての非 IP パケットが転送されます。

```
Device> enable
Device# configure terminal
Device(config)# access-list 101 permit udp any any
Device(config)# ip access-list extended igmp-match
Device(config-ext-nacl)# permit igmp any any
Device(config)# action forward
Device(config-ext-nacl)# permit tcp any any
Device(config-ext-nacl)# exit
Device(config)# vlan access-map drop-ip-default 10
Device(config-access-map)# match ip address 101
Device(config-access-map)# action forward
Device(config-access-map)# exit
Device(config)# vlan access-map drop-ip-default 20
Device(config-access-map)# match ip address igmp-match
Device(config-access-map)# action drop
Device(config-access-map)# exit
Device(config)# vlan access-map drop-ip-default 30
Device(config-access-map)# match ip address tcp-match
Device(config-access-map)# action forward
Device(config-access-map)# end
```

例：MAC パケットのドロップおよび IP パケットの転送のデフォルト アクション

次の例の VLAN マップでは、デフォルトで MAC パケットがドロップされ、IP パケットが転送されます。MAC 拡張アクセス リスト **good-hosts** および **good-protocols** をこのマップと組み合わせて使用すると、次のようになります。

- ホスト 0000.0c00.0111 および 0000.0c00.0211 からの MAC パケットが転送されます。
- decnet-iv または vines-ip プロトコルを使用する MAC パケットが転送されます。
- その他のすべての非 IP パケットがドロップされます。
- すべての IP パケットが転送されます。

```
Device> enable
Device# configure terminal
Device(config)# mac access-list extended good-hosts
Device(config-ext-macl)# permit host 000.0c00.0111 any
Device(config-ext-macl)# permit host 000.0c00.0211 any
Device(config-ext-nacl)# exit
Device(config)# action forward
Device(config-ext-macl)# mac access-list extended good-protocols
Device(config-ext-macl)# permit any any vines-ip
Device(config-ext-nacl)# exit
Device(config)# vlan access-map drop-mac-default 10
Device(config-access-map)# match mac address good-hosts
Device(config-access-map)# action forward
```

```
Device(config-access-map)# exit
Device(config)# vlan access-map drop-mac-default 20
Device(config-access-map)# match mac address good-protocols
Device(config-access-map)# action forward
Device(config-access-map)# end
```

例：すべてのパケットをドロップするデフォルト アクション

次の例の VLAN マップでは、デフォルトですべてのパケット（IP および非 IP）がドロップされます。例 2 および例 3 のアクセス リスト **tcp-match** および **good-hosts** をこのマップと組み合わせて使用すると、次のようになります。

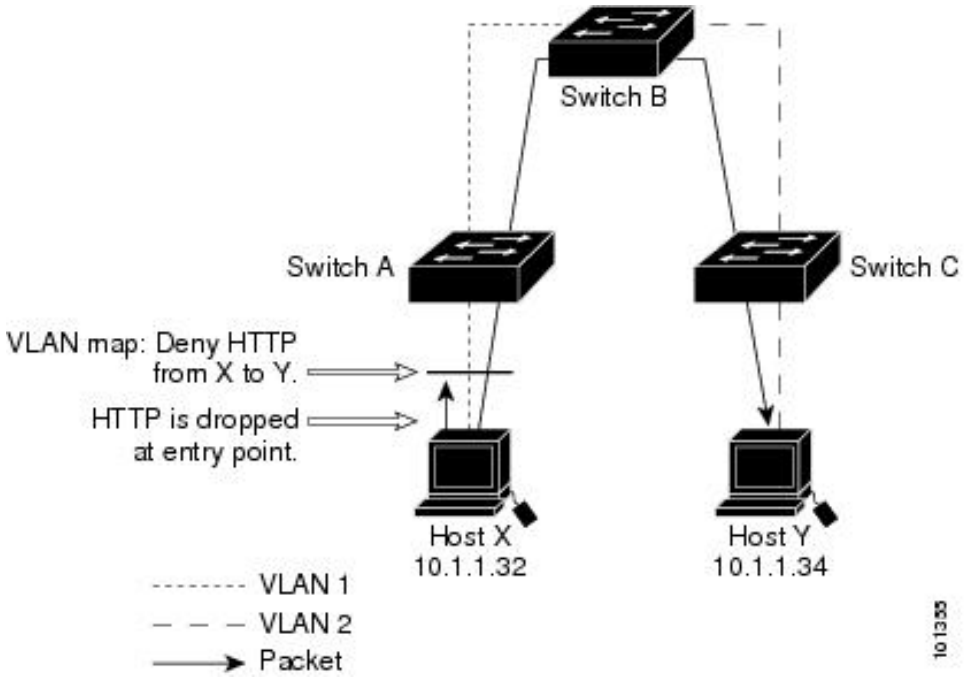
- すべての TCP パケットが転送されます。
- ホスト 0000.0c00.0111 および 0000.0c00.0211 からの MAC パケットが転送されます。
- その他のすべての IP パケットがドロップされます。
- その他のすべての MAC パケットがドロップされます。

```
Device> enable
Device# configure terminal
Device(config)# vlan access-map drop-all-default 10
Device(config-access-map)# match ip address tcp-match
Device(config-access-map)# action forward
Device(config-access-map)# exit
Device(config)# vlan access-map drop-all-default 20
Device(config-access-map)# match mac address good-hosts
Device(config-access-map)# action forward
Device(config-access-map)# end
```

例：ネットワークでの VLAN マップの使用

例：ワイヤリング クローゼットの設定

図 *15 :* ワイヤリング クローゼットの設定



HTTP トラフィックをホスト X からホスト Y へスイッチングしない場合は、ホスト X（IP アドレス 10.1.1.32）からホスト Y（IP アドレス 10.1.1.34）に向かうすべての HTTP トラフィックがスイッチ A でドロップされ、スイッチ B にブリッジングされないように、スイッチ A の VLAN マップを設定できます。

最初に、HTTP ポート上ですべての TCP トラフィックを許可（一致）する IP アクセス リスト *http* を定義します。

```
Device> enable
Device# configure terminal
Device(config)# ip access-list extended http
Device(config-ext-nacl)# permit tcp host 10.1.1.32 host 10.1.1.34 eq www
Device(config-ext-nacl)# end
```

次に、*http* アクセス リストと一致するトラフィックがドロップされ、その他のすべての IP トラフィックが転送されるように、VLAN アクセス マップ *map2* を作成します。

```
Device> enable
Device# configure terminal
Device(config)# vlan access-map map2 10
Device(config-access-map)# match ip address http
Device(config-access-map)# action drop
Device(config-access-map)# exit
Device(config)# ip access-list extended match_all
Device(config-ext-nacl)# permit ip any any
Device(config-ext-nacl)# exit
```

```
Device(config)# vlan access-map map2 20
Device(config-access-map)# match ip address match_all
Device(config-access-map)# action forward
Device(config-access-map)# end
```

次に、VLAN アクセス マップ *map2* を VLAN 1 に適用します。

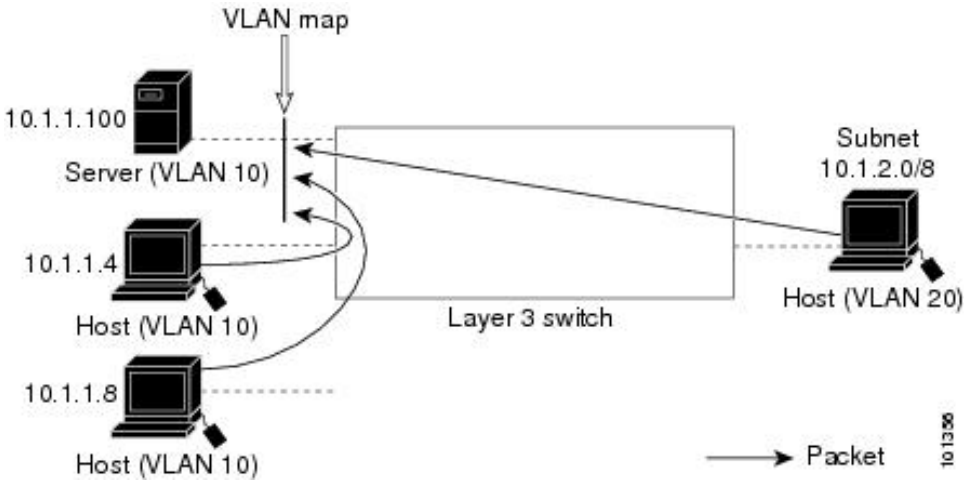
```
Device> enable
Device# configure terminal
Device(config)# vlan filter map2 vlan 1
Device(config)# end
```

例：別の VLAN にあるサーバーへのアクセスの制限

図 *16* : 別の *VLAN* 上のサーバーへのアクセスの制限

別の VLAN にあるサーバーへのアクセスを制限できます。たとえば、VLAN 10 内のサーバー 10.1.1.100 では、次のホストへのアクセスを拒否する必要があります。

- VLAN 20 内のサブネット 10.1.2.0/8 にあるホストのアクセスを禁止します。
- VLAN 10 内のホスト 10.1.1.4 および 10.1.1.8 のアクセスを禁止します。



例：別の VLAN にあるサーバーへのアクセスの拒否

次に、サブネット 10.1.2.0.8 内のホスト、ホスト 10.1.1.4、およびホスト 10.1.1.8 のアクセスを拒否し、その他の IP トラフィックを許可する VLAN マップ SERVER1-ACL を作成して、別の VLAN 内のサーバーへのアクセスを拒否する例を示します。最後のステップでは、マップ SERVER1 を VLAN 10 に適用します。

正しいパケットと一致する IP ACL を定義します。

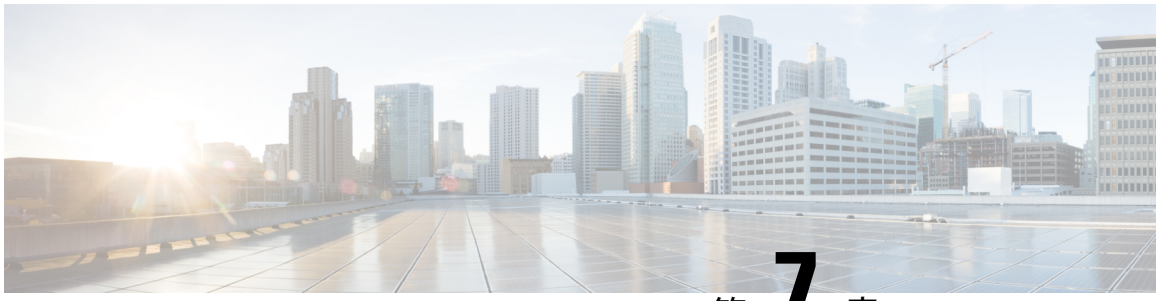
```
Device> enable
Device# configure terminal
Device(config)# ip access-list extended SERVER1_ACL
Device(config-ext-nacl)# permit ip 10.1.2.0 0.0.0.255 host 10.1.1.100
Device(config-ext-nacl)# permit ip host 10.1.1.4 host 10.1.1.100
Device(config-ext-nacl)# permit ip host 10.1.1.8 host 10.1.1.100
Device(config-ext-nacl)# end
```

SERVER1_ACL と一致する IP パケットをドロップして、この ACL と一致しない IP パケットを転送する ACL を使用して、VLAN マップを定義します。

```
Device> enable
Device# configure terminal
Device(config)# vlan access-map SERVER1_MAP
Device(config-access-map)# match ip address SERVER1_ACL
Device(config-access-map)# action drop
Device(config)# vlan access-map SERVER1_MAP 20
Device(config-access-map)# action forward
Device(config-access-map)# end
```

VLAN 10 に VLAN マップを適用します。

```
Device> enable
Device# configure terminal
Device(config)# vlan filter SERVER1_MAP vlan-list 10
Device(config)# end
```



第 7 章

IPv6 ACL

IPv6 ACL の制限

IPv6 がサポートするのは名前付き ACL だけです。IPv4 ACL 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。

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、**flowlabel**、**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。
- スイッチは再起 ACL（**reflect** キーワード）をサポートしません。
- **vrf-also** キーワードは、IPv6 access-class 行コマンドと相互に排他的です。
- 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、制約事項はありません。ハードウェア転送が必要なインターフェイス（物理ポートまたは SVI）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。ACL がインターフェイスでサポートされていない場合、ACL 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。

- プロトコルの TCAM をプログラムしないインターフェイスと、アンロードされた ACL にスケール ACL を適用すると、他のプロトコルのトラフィックの既存の通常移動に影響を与える可能性があります。この制限は、に該当します。
- 存続可能時間（TTL）分類は、ACL ではサポートされていません。
- ダウンロード可能な ACL に重複するエントリが含まれている場合、エントリは自動的にマージされません。その結果、802.1X セッション許可は失敗します。ダウンロード可能な ACL が、同じポートのポートベースのエントリや名前ベースのエントリなど、重複するエントリなしで最適化されていることを確認します。
- ソフトウェアによって転送される、注入されたトラフィックでは、出力 ACL ルックアップはサポートされていません。
- ACL は、レイヤ 3 インターフェイス（ルーテッドインターフェイスや VLAN インターフェイスなど）のみをサポートします。

IPv6 ACL の概要

ここでは、IPv6 ACL について説明します。

IPv6 ACL の概要

このトピックでは、IPv6 ACL の概要を示します。

アクセスコントロールリスト（ACL）とは、特定のインターフェイスへのアクセスを制限するために使用されるルールセットのことです。ACL はデバイスに設定され、管理インターフェイスおよび任意の動的インターフェイスに適用されます。

Web 認証用に事前認証 ACL を作成することもできます。このような ACL は、認証が完了するまでに特定のタイプのトラフィックを許可するために使用されます。

IPv6 ACL は、送信元、宛先、送信元ポート、宛先ポートなど、IPv4 ACL と同じオプションをサポートします。

サポートされる ACL

スイッチでは、トラフィックをフィルタリングするために、次に示す 3 種類の ACL がサポートされています。

- 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。アクセスリストタイプ（IPv4、IPv6、および MAC）のどの方向に対してでも、レイヤ 2 インターフェイスにポート ACL を適用できます。
- ルータ ACL は、VLAN 間でルーティングされたトラフィックのアクセスを制御し、レイヤ 3 インターフェイスで特定の方向（インバウンドまたはアウトバウンド）に適用されます。

ACL のタイプ

次のセクションでは ACL のタイプについて説明します。

ユーザー単位 IPv6 ACL

ユーザーあたりの ACL の場合、テキスト文字列としての完全なアクセス コントロール エントリ（ACE）が Cisco Secure Access Control Server（Cisco Secure ACS）で設定されます。

フィルタ ID IPv6 ACL

filter-Id ACL の場合、完全な ACE および `acl name(filter-id)` がデバイスで設定され、`filter-id` のみが次に設定されます。 Cisco Secure ACS で設定されます。

ACL 優先順位

ポート ACL、およびルータ ACL が同じスイッチに設定されている場合、入力トラフィックの場合のフィルタの優先順位は上からポート ACL、およびルータ ACL です。出力トラフィックの場合、フィルタの優先順位は、ルータ ACL、ポート ACL です。

次の例で、簡単な使用例を説明します。

- スイッチ仮想インターフェイス（SVI）に入力ルータ ACL および入力ポート ACL が設定されている場合に、ポート ACL が適用されているポートにパケットが着信すると、このパケットはポート ACL によってフィルタリングされます。他のポートで受信した着信のルーティング IP パケットには、ルータ ACL のフィルタが適用されます。他のパケットはフィルタリングされません。
- SVI に出力ルータ ACL および入力ポート ACL が設定されている場合に、ポート ACL が適用されているポートにパケットが着信すると、このパケットはポート ACL によってフィルタリングされます。発信するルーティング IP パケットには、ルータ ACL のフィルタが適用されます。他のパケットはフィルタリングされません。

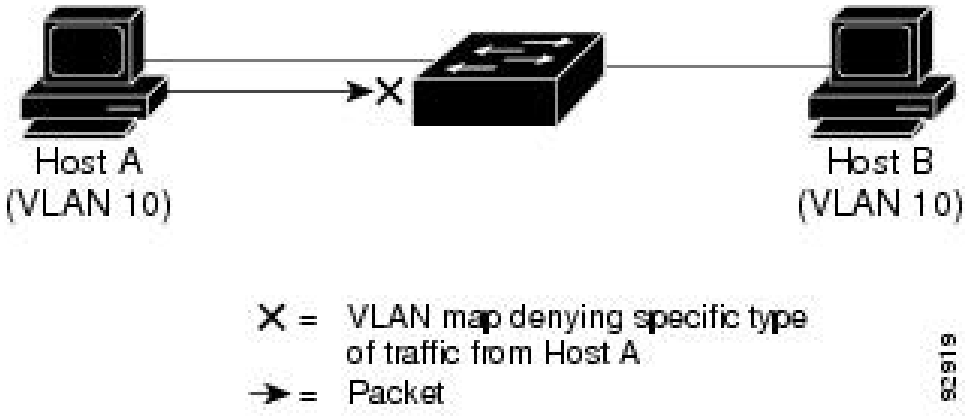
VLAN マップ

VLAN ACL または VLAN マップは、VLAN 内のネットワークトラフィックを制御するために使用されます。スイッチの VLAN 内でブリッジングされるすべてのパケットに VLAN マップを適用できます。VACL は、セキュリティ パケット フィルタリングおよび特定の物理インターフェイスへのトラフィックのリダイレクトだけを目的としたものです。VACL は方向（入力または出力）で定義されることはありません。

すべての非 IP プロトコルは、MAC VLAN マップを使用して、MAC アドレスおよび Ethertype によってアクセス コントロールされます（IP トラフィックは、MAC VLAN マップではアクセス制御されません）。VLAN マップはスイッチを通過するパケットにだけ適用できます。ハブ上またはこのスイッチに接続された別のスイッチ上のホスト間のトラフィックには、VLAN マップを適用できません。

VLANマップを使用すると、マップに指定されたアクションに基づいてパケットの転送が許可または拒否されます。

図 *17 : VLAN* マップによるトラフィックの制御



他の機能およびスイッチとの相互作用

- IPv6 ルータ ACL がパケットを拒否するよう設定されている場合、パケットはルーティングされません。パケットのコピーがインターネット制御メッセージ プロトコル（ICMP）キューに送信され、フレームに ICMP 到達不能メッセージが生成されます。
- 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 インターフェイスまたはレイヤ 3 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 メモリに空きがない場合、パケットはインターフェイスでドロップされ、アンロードのエラー メッセージが記録されます。

  ハードウェア メモリが満杯の場合、設定済みの ACL を追加すると、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。ハードウェアが一杯になると、ACL がアンロードされたことを示すメッセージがコンソールに出力され、パケットはインターフェイスでドロップされます。

IPv6 ACL の設定方法

ここでは、IPv6 ACL の設定方法に関する情報を示します。

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
```

IPv6 ACL の設定

IPv6 トラフィックをフィルタリングするには、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。 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* | **log-update threshold** | **role-based** *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 threshold**| **host** *source-ipv6-address*} [operator [*port-number*]] { *destination-ipv6-prefix/ prefix-length* | **any** | **host** *destination-ipv6-address*}	IPv6 ACL の許可条件または拒否条件を指定します。 • protocol には、IP の名前または番号を入力します。 **ahp**、**esp**、**icmp**、**ipv6**、**pcp**、**stcp**、**tcp**、

	コマンドまたはアクション	目的
	[operator [*port-number*]][**dscp** *value*] [**fragments**] [**log**] [**log-input**][**sequence** *value*] [**time-range** *name*] 例： `Device(config-ipv6-acl)# permit tcp 2001:DB8:0300:0201::/32 eq telnet any`	**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）、および **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

	コマンドまたはアクション	目的
		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。ロギングはルータ ACL でだけサポートされます。 • （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、拒否または許可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*] [**established**] [**fin**] [**log**] [**log-input**] [**neq** {**port** | protocol}] [**psh**] [**range** {**port** | **protocol**}] [**rst**] [**sequence** *value*] [**syn**] [**time-range** *name*] [**urg**] 例： `Device(config-ipv6-acl)# deny tcp host 2001:DB8:1::1 any log-input`	IPv6 ACL の許可条件または拒否条件を指定します。 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（ACK）ビットセット。 • **established**：確立された接続。TCP データグラムに ACK または RST ビットが設定されている場合、照合が行われます。 • **fin**：終了ビットセット。送信者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

	コマンドまたはアクション	目的
ステップ 6	**end** 例： `Device(config-ipv6-acl)# end`	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、特権 EXEC モードに戻ります。
ステップ 7	**show ipv6 access-list** 例： `Device# show ipv6 access-list`	IPv6 ACL が正しく設定されていることを確認します。

インターフェイスへの IPv6 ACL の付加

レイヤ 3 インターフェイスで発信または着信トラフィックに ACL を、あるいはレイヤ 2 インターフェイスで着信トラフィックに を適用できます。レイヤ 3 インターフェイスで着信トラフィックにだけ ACL を適用できます。

インターフェイスへのアクセスを制御するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **no switchport**
5. **ipv6 address** *ipv6-address*
6. **ipv6 traffic-filter** *access-list-name* {**in** | **out**}
7. **end**

手順の詳細

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	**interface** *interface-id* 例：	アクセスリストを適用するレイヤ 2 インターフェイス（ポート ACL 用）またはレイヤ 3 インターフェ

	コマンドまたはアクション	目的
	`Device(config)# interface gigabitethernet1/0/1`	イス（ルータ ACL 用）を指定して、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 4	**no switchport** 例： `Device(config-if)# no switchport`	インターフェイスをルーテッドインターフェイスの状態に戻して、レイヤ 2 の詳細設定をすべて削除します。
ステップ 5	**ipv6 address** *ipv6-address* 例： `Device(config-if)# ipv6 address 2001:DB8::1`	レイヤ3インターフェイス（ルータ ACL 用）で Ipv6 アドレスを設定します。
ステップ 6	**ipv6 traffic-filter** *access-list-name* {**in** | **out**} 例： `Device(config-if)# ipv6 traffic-filter acl1 in`	インターフェイスの着信トラフィックまたは発信トラフィックにアクセス リストを適用します。
ステップ 7	**end** 例： `Device(config-ipv6-acl)# 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。

テンプレートモードでの IPv6 ACL の設定

（注） **ipv6 traffic-filter** コマンドはテンプレート コンフィギュレーション モードで設定できます。**source template** コマンドは、インターフェイスに対して 1 回だけ設定できます。

ACL をテンプレートで設定するには、特権 EXEC モードで次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** {*list-name* | **log-update threshold** | **role-based** *list-name*}
4. **ipv6 access-list** {*list-name* | **log-update threshold** | **role-based** *list-name*}
5. **exit**
6. **template**
7. **ipv6 traffic-filter** {*access-list-number* | *name*} {**in** | **out**}
8. **exit**
9. **interface** *interface-id*
10. **ipv6 traffic-filter** {*access-list-number* | *name*} {**in** | **out**}
11. **source template** *name*
12. **end**

手順の詳細

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* | **log-update threshold** | **role-based** *list-name*} 例： `Device(config)# ipv6 access-list v6acl10`	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	**ipv6 access-list** {*list-name* | **log-update threshold** | **role-based** *list-name*} 例： `Device(config-ipv6-acl)#ipv6 access-list v6acl11`	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 5	**exit** 例： `Device(config-ipv6-acl)#exit`	アクセス リスト コンフィギュレーション モードを終了します。
ステップ 6	**template** 例： `Device(config)# template test`	ユーザーテンプレートを作成し、テンプレート コンフィギュレーション モードを開始します。
ステップ 7	**ipv6 traffic-filter** {*access-list-number* | *name*} {**in** | **out**} 例： `Device(config-template)# ipv6 traffic-filter v6acl10 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。 *access-list-number* を入力して、アクセスリストを定義します。アクセスリストには番号を指定できます。 *name* を入力して、アクセスリストを定義します。アクセスリストには名前を指定できます。 **in** を入力して、インターフェイスの着信方向にアクセスリストを送信します。 **out** を入力して、インターフェイスの発信方向にアクセスリストを送信します。

	コマンドまたはアクション	目的
ステップ 8	**exit** 例： `Device(config-template)# exit`	テンプレートの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	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet1/0/1`	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）またはレイヤ 3 インターフェイス（ルータ ACL）を指定できます。
ステップ 10	**ipv6 traffic-filter** {*access-list-number* | *name*} {**in** | **out**} 例： `Device(config-if)# ipv6 traffic-filter v6acl11 out`	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。 *access-list-number* を入力して、アクセスリストを定義します。アクセスリストには番号を指定できます。 *name* を入力して、アクセスリストを定義します。アクセスリストには名前を指定できます。 **in** を入力して、インターフェイスの着信方向にアクセスリストを送信します。 **out** を入力して、インターフェイスの発信方向にアクセスリストを送信します。
ステップ 11	**source template** *name* 例： `Device(config)# source template test`	インターフェイス テンプレートをターゲットに適用します。アクセスリスト *v6acl10* は、設定されている着信アクセスリストです。
ステップ 12	**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。

VLAN マップの設定

VLAN マップを作成して、1 つまたは複数の VLAN に適用するには、次のステップを実行します。

始める前に

VLAN に適用する IPv6 ACL を作成します。

手順の概要

1. **enable**

2. **configure terminal**
3. **vlan access-map** *name* [**number**]
4. **match** {**ip** | **ipv6** | **mac**} **address** {*name* | *number*} [*name* | *number*]
5. IP パケットまたは非 IP パケットを（既知の 1 MAC アドレスのみを使って）指定し、1 つ以上の ACL とそのパケットを照合するには、次のコマンドのいずれかを入力します。
   - **action** { **forward**}

     ```
     Device(config-access-map)# action forward
     ```

   - **action** { **drop**}

     ```
     Device(config-access-map)# action drop
     ```

6. **vlan filter** *mapname* **vlan-list** *list*
7. **end**

手順の詳細

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	**vlan access-map** *name* [**number**] 例： `Device(config)# vlan access-map map_1 20`	VLANマップを作成して、VLANアクセスマップ コマンド モードを開始します。 VLANマップには、名前または（オプションで）番号を指定できます。番号は、マップ内のエントリのシーケンス番号です。 同じ名前の VLAN マップを作成すると、10 ずつ増加する番号が順に割り当てられます。マップを変更または削除するときは、該当するマップエントリの番号を入力できます。 VLAN マップでは、特定の permit または deny キーワードを使用しません。VLANマップを使用してパケットを拒否するには、パケットを照合する ACL を作成して、アクションをドロップに設定します。ACL 内の permit は、一致するという意味です。ACL 内の deny は、一致しないという意味です。

	コマンドまたはアクション	目的
ステップ 4	**match** {**ip** | **ipv6** | **mac**} **address** {*name* | *number*} [*name* | *number*] 例： `Device(config-access-map)# match ipv6 address ip_net`	パケットを 1 つまたは複数のアクセスリストと照合します。パケットの照合は、対応するプロトコルタイプのアクセスリストに対してだけ行われます。IP パケットは、IP アクセスリストに対して照合されます。非 IP パケットは、名前付き MAC アクセスリストに対してだけ照合されます。 （注） パケット タイプ（IP または MAC）に対する match 句が VLAN マップに設定されている場合で、そのマップ アクションがドロップの場合は、そのタイプに一致するすべてのパケットがドロップされます。match 句が VLAN マップになく、設定されているアクションがドロップの場合は、すべての IP およびレイヤ 2 パケットがドロップされます。

ステップ 5	IP パケットまたは非 IP パケットを（既知の 1 MAC アドレスのみを使って）指定し、1 つ以上の ACL とそのパケットを照合するには、次のコマンドのいずれかを入力します。
	• **action** { **forward**}
	`Device(config-access-map)# action forward`
	• **action** { **drop**}
	`Device(config-access-map)# action drop`
		マップエントリに対するアクションを設定します。

ステップ 6	**vlan filter** *mapname* **vlan-list** *list* 例： `Device(config)# vlan filter map 1 vlan-list 20-22`	VLAN マップを 1 つまたは複数の VLAN に適用します。 list には単一の VLAN ID（22）、連続した範囲（10 ～ 22）、または VLAN ID のストリング（12、22、30）を指定できます。カンマやハイフンの前後にスペースを挿入することもできます。
ステップ 7	**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。

VLAN への VLAN マップの適用

VLAN マップを 1 つまたは複数の VLAN に適用するには、次の手順に従います。

手順の概要

1. **enable**
2. **configure terminal**
3. **vlan filter** *mapname* **vlan-list** *list*
4. **end**

手順の詳細

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	**vlan filter** *mapname* **vlan-list** *list* 例： `Device(config)# vlan filter map 1 vlan-list 20-22`	VLAN マップを 1 つまたは複数の VLAN に適用します。 list には単一の VLAN ID（22）、連続した範囲（10 ～ 22）、または VLAN ID のストリング（12、22、30）を指定できます。カンマやハイフンの前後にスペースを挿入することもできます。
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。

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。

表 *13 : show ACL* コマンド

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。

コマンド	目的
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。
show vlan access-map [*map-name*]	VLAN アクセス マップ設定を表示します。
show vlan filter [**access-map** *access-map* | **vlan** *vlan-id*]	VACL と VLAN 間のマッピングを表示します。

IPv6 ACL の設定例

ここでは、IPv6 ACL の設定例を示します。

例：IPv6 ACL の作成

この例では、IPv6-ACL という名前の IPv6 アクセスリストを設定します。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。

（注） ロギングは、レイヤ 3 インターフェイスでのみサポートされます。

```
Device> enable
Device(config)# ipv6 access-list IPv6_ACL
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
Device(config-ipv6-acl)# end
```

例：IPv6 ACL の表示

次に、**show access-lists** コマンドの出力例を示します。出力には、デバイス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** 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
```

例：VLAN アクセスマップ設定の表示

次に、**show vlan access-map** 特権 EXEC コマンドの出力例を示します。

```
Device# show vlan access-map

Vlan access-map "m1"  10
  Match clauses:
    ipv6 address: ip2
  Action: drop
```

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
```

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。