アクセス リストおよびプレフィックス リストの実装

アクセス コントロール リスト（ACL）は、ネットワーク トラフィック プロファイルをまとめて定義する 1 つ以上のアクセス コントロール エントリ（ACE）です。 このプロファイルはその後、トラフィック フィルタリング、ルート フィルタリング、QoS 分類、アクセス コントロールなど、Cisco IOS XR ソフトウェアの機能で参照できます。 各 ACL には、送信元アドレス、宛先アドレス、プロトコル、およびプロトコルに固有のパラメータなどの基準に基づく、アクション要素（許可または拒否）やフィルタ要素が含まれています。

プレフィックス リストはルート マップおよびルート フィルタリング操作に使用されるほか、ボーダー ゲートウェイ プロトコル（BGP）の多くのルート フィルタリング コマンドではアクセス リストの代わりに使用できます。 プレフィックスは IP アドレスの一部であり、左端のオクテットの左端のビットから始まります。 アドレスの何ビットがプレフィックスに属するかを正確に指定すると、プレフィックスを使用してアドレスを集約し、そのアドレスに対して再配布（フィルタ ルーティング アップデート）などの機能を実行できるようになります。

この章では、次の製品にアクセス リストおよびプレフィックス リストを実装するのに必要な新規のタスクおよび改訂されたタスクについて説明します： Cisco ASR 9000 シリーズ ルータ

（注） この章に記載されているアクセス リストおよびプレフィックス リストのコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』を参照してください。この章で使用される他のコマンドの説明については、コマンド リファレンスのマスター索引を参照するか、またはオンラインで検索してください。

アクセス リストおよびプレフィックス リストの実装の機能履歴

リリース	変更内容
リリース 3.7.2	この機能が導入されました。
リリース 4.2.1	BVI インターフェイス上の IPv6 ACL 機能が追加されました。

リリース	変更内容
リリース 4.2.1	クラス マップの ACL 機能が追加されました。

アクセスリストおよびプレフィックスリストの実装の前提条件

アクセス リストおよびプレフィックス リストの実装には、次の前提条件が適用されます。

すべてのコマンド タスク ID は、それぞれのコマンド リファレンスと、『Cisco IOS XR Task ID Reference Guide』に記載されています。タスク グループの割り当てについて支援が必要である場合は、システム管理者にお問い合わせてください。

アクセスリストおよびプレフィックスリストの実装の制約事項

アクセス リストおよびプレフィックス リストの実装には、次の制約事項が適用されます。

- IPv4 ACL は、ループバック インターフェイスおよびインターフレックス インターフェイスではサポートされません。

- IPv6 ACL は、ループバック、インターフレックス、および L2 イーサネット フロー ポイント（EFP）のメインまたはサブインターフェイスではサポートされません。

ACL ベース転送（ABF）の実装には、次の制約事項が適用されます。

- ネクスト ホップ オプションを持つ ACL を出方向に接続する設定、ネクスト ホップを持ち出方向に接続された ACL を変更する設定、ネクスト ホップを持つ ACE を拒否する設定のネクスト ホップ設定はサポートされていません。
- リリース 4.2.0 では、A9K-SIP-700 LC および ASR 9000 Enhanced Ethernet LC は ABFv4 および ABFv6 をサポートします。 リリース 4.2.0 では、ASR 9000 Ethernet LC は ABFv6 をサポートせず、ABFv4 のみをサポートします。
- ABFv4 は ASR 9000 Enhanced Ethernet ラインカードの BVI インターフェイスでサポートされます。 ASR 9000 Ethernet ラインカードではサポートされません。

  ABFv6 は両方のラインカードでサポートされません。

（注） A9K-SIP-700 ラインカード、ASR 9000 Ethernet ラインカード、または GRE や BVI などの仮想インターフェイス上のネクスト ホップ出力は、ABFv4 が BVI インターフェイス用に設定されている場合にサポートされます。

（注） これには例外が 1 つあります。 IP to TAG の場合、入力 LC が（ABF ネクスト ホップに基づいて）ラベルを提供するため、パケットはタグ パケットとしてファブリックを横断します。 このようなパケットは、A9K-SIP-700 によって問題なく処理されます。

- 低速パスでは ABF がサポートされないため、NPU から LC CPU へと入力方向にパントされたパケットは ABF では処理されません。
- フラグメンテーションを必要とする IP パケットは、ABF で処理されません。 そのようなパケットは、従来の方法で転送されます。 フラグメント化されたパケットは受信後、ABF によって処理されます。

ハードウェアの制限

- ABF のサポートは、IPv4 およびイーサネット ラインカードのみが対象です。 IPv6 とその他のインターフェイスはサポートされません。
- ABF は入力ラインカードの機能であるため、出力ラインカードは ABF に対応している必要があります。

アクセスリストおよびプレフィックスリストの実装に関する情報

アクセスリストおよびプレフィックスリストを実装するには、次の概念を理解する必要があります。

アクセス リストおよびプレフィックス リスト機能のハイライト

ここでは、アクセス リストとプレフィックス リストの機能のハイライトを示します。

- Cisco IOS XR ソフトウェア では、特定のシーケンス番号を指定して、アクセス リストまたはプレフィックス リストのカウンタをクリアできます。
- Cisco IOS XR ソフトウェア では、既存のアクセス リストまたはプレフィックス リストの内容を別のアクセス リストまたはプレフィックス リストにコピーできます。
- Cisco IOS XR ソフトウェア では、permit ステートメントまたは deny ステートメントにシーケンス番号を適用して、名前付きのアクセス リストまたはプレフィックス リストでこのようなステートメントの並べ替え、追加、または削除を実行できます。

（注） 並べ替えは、IPv4 プレフィックス リストのみが対象です。

- Cisco IOS XR ソフトウェア は、標準アクセス リストと拡張アクセス リストとを区別しません。 標準アクセス リストをサポートしているのは、下位互換性を確保するためです。

IP アクセス リストの目的

アクセスリストは、パケットフィルタリングを実行して、ネットワークを介して移動するパケットとその場所を制御します。 この処理は、ネットワークトラフィックを制限したり、ユーザやデバイスによるネットワークへのアクセスを制限したりするのに役立ちます。 アクセスリストの用途は多様なので、多くのコマンドの構文でアクセス リストが参照されます。 アクセス リストを使用して、次のようなことを実行できます。

- インターフェイスでの着信パケットのフィルタリング
- インターフェイスでの発信パケットのフィルタリング
- ルーティング アップデートの内容の制限
- アドレスまたはプロトコルに基づくデバッグ出力の制限
- vty へのアクセスの制御

- 輻輳回避、輻輳管理、プライオリティ キューイング、カスタム キューイングなどの高度な機能に使用されるトラフィックの特定または分類

IP アクセス リストの機能

アクセス リストは、permit ステートメントと deny ステートメントで構成される順次リストです。これらのステートメントは、IP アドレス、場合によっては上位層 IP プロトコルに適用されます。アクセス リストには、参照に使用される名前があります。 多くのソフトウェア コマンドは、構文の一部としてアクセス リストを受け取ります。

アクセスリストを設定して名前を付けることは可能ですが、アクセスリストを受け取るコマンドによってアクセスリストが参照されるまで、有効にはなりません。複数のコマンドから同じアクセス リストを参照できます。 アクセス リストで、ルータに到達するトラフィック、またはルータ経由で送信されるトラフィックは制御できますが、ルータが送信元のトラフィックは制御できません。

IP アクセス リストのプロセスとルール

IP アクセス リストを設定するときは、次のプロセスとルールを使用してください。

- アクセスリストの条件に対してフィルタリングされる各パケットの送信元アドレスや宛先アドレス、またはプロトコルがテストされます。一度に 1 つの条件（permit ステートメントまたは deny ステートメント）がテストされます。
- パケットがアクセスリストのステートメントに一致しないと、そのパケットはリスト内の次のステートメントに対してテストされます。
- パケットとアクセスリストのステートメントが一致すると、リスト内の残りのステートメントはスキップされ、パケットは一致したステートメントに指定されたとおりに許可または拒否されます。パケットが許可されるか拒否されるかは、パケットが一致する最初のエントリによって決まります。 つまり、一致すると、それ以降のエントリは考慮されません。
- アクセス リストでアドレスまたはプロトコルが拒否されると、パケットは廃棄され、インターネット制御メッセージ プロトコル（ICMP）ホスト到達不能メッセージが返されます。ICMP は、Cisco IOS XR ソフトウェアで設定できます。
- 各アクセス リストの最後には暗黙の deny ステートメントがあるため、一致する条件がない場合は、パケットはドロップされます。つまり、各ステートメントに対してテストするときまでにパケットを許可または拒否しないと、パケットは拒否されます。
- アクセス リストには permit ステートメントを 1 つ以上含める必要があります。そうしないと、パケットはすべて拒否されます。
- 最初に一致が見つかった後は条件のテストが終了するため、条件の順序は重要です。 同じ permit ステートメントまたは deny ステートメントでも、順序が異なる場合、ある状況では通過し、別の状況では拒否されるパケットが生じる可能性があります。

- 1 つのインターフェイス、1 つのプロトコル、1 つの方向につき、許可されるアクセス リストは 1 つだけです。
- インバウンド アクセス リストは、ルータに到達するパケットを処理します。 着信パケットの処理後に、アウトバウンド インターフェイスへのルーティングが行われます。 インバウンド アクセス リストが効率的なのは、フィルタリング テストで拒否されたことでパケットが廃棄される場合、ルーティング検索のオーバーヘッドが抑えられるためです。 パケットがテストで許可されると、そのパケットに対してルーティングの処理が実施されます。インバウンドリストの場合、permit とは、インバウンドインターフェイスでパケットの受信後に処理が続行されることを示します。**deny** とは、パケットが廃棄されることを示します。
- アウトバウンド アクセス リストの場合、パケットの処理後にルータから送信されます。 着信パケットはアウトバウンドインターフェイスにルーティングされてから、アウトバウンド アクセス リストで処理されます。 アウトバウンド リストの場合、許可とは、出力バッファに対して送信されることを示し、拒否とは、パケットが廃棄されることを示します。
- アクセス リストは、使用中のアクセス グループによって適用されている場合には削除できません。 アクセス リストを削除するには、まずアクセス リストを参照しているアクセス グループを削除してから、アクセス リストを削除します。
- **ipv4 access group** コマンドを使用するには、アクセス リストが存在している必要があります。

IP アクセス リストを作成する際に役立つヒント

IP アクセス リストを作成する場合は、次の事項を考慮してください。

- アクセス リストは、インターフェイスに適用する前に作成します。
- ネットワークまたはサブネットのより具体的な参照が、より全般的な参照よりも前に出現するように、アクセス リストを構成します。
- 個々のステートメントの用途をひと目で確認および理解しやすくするために、ステートメントの前または後に役立つ注記を書き込みます。

送信元アドレスと宛先アドレス

送信元アドレスと宛先アドレスは、IP パケットの最も一般的な 2 つのフィールドで、アクセス リストの基礎となります。 送信元アドレスを指定して、特定のネットワーキング デバイスまたはホストからのパケットを制御します。宛先アドレスを指定して、特定のネットワーキング デバイスまたはホストに送信されるパケットを制御します。

ワイルドカード マスクと暗黙のワイルドカード マスク

アドレス フィルタリングでは、アクセス リスト エントリ内のアドレス ビットとアクセス リストに送信されるパケットを比較するときに、ワイルドカードマスクを使用して、対応する IP アドレ

ス ビットを確認するか無視するかを指定します。 管理者は、ワイルドカード マスクを慎重に設定することにより、許可または拒否のテストに 1 つまたは複数の IP アドレスを選択できます。

IP アドレス ビット用のワイルドカード マスクでは、数値 1 と数値 0 を使用して、対応する IP アドレス ビットをどのように扱うかを指定します。 1 と 0 は、サブネット（ネットワーク）マスクで意味する内容が逆になるため、ワイルドカード マスクは逆マスクとも呼ばれます。

- ワイルドカード マスク ビット 0 は、対応するビット値を確認することを示します。
- ワイルドカード マスクのビット 1 は、対応するビット値を無視することを意味します。

アクセス リスト ステートメントでは、送信元アドレスまたは宛先アドレスにワイルドカード マスクを指定する必要はありません。 **host** キーワードを使用した場合は、ワイルドカードマスクとして 0.0.0.0 を指定したものと見なされます。

サブネット マスクでは、ネットワークとサブネットを示す隣接ビットをマスクにする必要がありますが、それとは異なり、ワイルドカード マスクではマスクに非隣接ビットを使用できます。 IPv6 アクセス リストでは、隣接ビットのみがサポートされます。

ワイルドカード ビットの代わりに、CIDR 形式（/x）を使用することもできます。 たとえば、アドレス 1.2.3.4 0.255.255.255 は 1.2.3.4/8 と表すことができます。

トランスポート層の情報

トランスポート層の情報（パケットが TCP、UDP、ICMP、IGMP のいずれのパケットであるかなどの情報）に基づいてパケットをフィルタリングできます。

IP アクセス リスト エントリ シーケンス番号

IP アクセス リスト エントリにシーケンス番号を適用する機能によって、アクセス リストの変更が簡易になります。 この機能がない頃は、アクセスリスト内のエントリの位置を指定する方法はありませんでした。 以前は、既存のリストの途中にエントリ（ステートメント）を挿入する場合、目的の位置の後にあるすべてのエントリを削除してから、新しいエントリを追加し、削除したすべてのエントリを再入力する必要がありました。 これは手間がかかり、エラーが起こりやすい方法です。

IP アクセス リスト エントリ シーケンス番号機能を使用すると、アクセス リスト エントリにシーケンス番号を追加し、リスト内のエントリを並べ替えることができます。 新しいエントリを追加する場合、アクセスリストの目的の位置に挿入されるようにシーケンス番号を選択します。 必要に応じて、アクセス リストの現在のエントリを並べ替えて、新しいエントリを挿入できる場所を作成できます。

シーケンス番号の動作

ここでは、シーケンス番号の動作を詳しく説明します。

- シーケンス番号のないエントリを複数適用すると、最初のエントリにシーケンス番号 10 が割り当てられ、それ以降のエントリには 10 ずつ増分したシーケンス番号が割り当てられま

す。 最大シーケンス番号は 2147483646 です。 生成したシーケンス番号がこの最大値を超えると、次のメッセージが表示されます。

```
Exceeded maximum sequence number.
```

- シーケンス番号のないエントリを 1 つ指定すると、アクセス リストの最後のシーケンス番号に 10 を加えたシーケンス番号が割り当てられ、リストの末尾に配置されます。
- ACL エントリは、トラフィック フローにもハードウェアのパフォーマンスにも影響を及ぼすことなく追加できます。
- グローバル コンフィギュレーション モードで新しいアクセス リストを入力すると、そのアクセス リストのシーケンス番号が自動的に生成されます。
- ルート プロセッサ（RP）のエントリとラインカード（LC）のエントリのシーケンス番号を常に同期できるように、分散機能がサポートされています。
- この機能は、名前付きの標準および拡張 IP アクセス リストと連動します。 アクセス リストの名前を番号として指定できるため、番号も使用できます。

IP アクセス リスト ログ メッセージ

Cisco IOS XR ソフトウェア では、標準 IP アクセス リストで許可または拒否されたパケットに関するログ メッセージが表示されます。 つまり、パケットがアクセス リストに一致すると、そのパケットに関するログメッセージ情報がコンソールに送信されます。ログをコンソールに送信するメッセージのレベルは、グローバル コンフィギュレーション モードの **logging console** コマンドで制御します。

最初にパケットがアクセスリストをトリガーすると、すぐにログメッセージが生成されます。その後、5 分間隔でパケットが収集されて表示または記録されます。 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。

ただし、{ **ipv4 | ipv6** } **access-list log-update threshold** コマンドを使用すると、アクセス リストに一致したパケットを許可または拒否する際に、ログ メッセージを生成するパケットの数を設定できます。 この手順は、5 分間隔よりも短い頻度でログ メッセージを受信する場合に実行することを推奨します。

注意 *number-of-matches* 引数を 1 に設定すると、ログ メッセージはキャッシュされずにただちに送信されます。この場合、アクセス リストに一致するすべてのパケットについてログ メッセージが生成されます。 大量のログ メッセージでシステムが過負荷になる可能性があるため、1 に設定することは推奨されません。

{ ipv4 | ipv6} access-list log-update threshold コマンドを使用する場合でも、5 分タイマーは有効なままなので、各キャッシュのメッセージ数に関係なく、5 分が経過すると各キャッシュは空になります。 ログメッセージを送信するタイミングに関係なく、しきい値が指定されていない場合と同様に、ログ メッセージのキャッシュは消去され、カウントは 0 にリセットされます。

（注） ログ メッセージが多すぎて処理できない場合や、1 秒以内に 2 つ以上のログ メッセージを処理した場合には、ログ メッセージ パケットの一部がドロップされることがあります。 この動作により、ログを生成するパケットの数が多くなっても、ルータが CPU サイクルを過度に使用することはありません。 したがって、ロギング機能は課金ツールや、アクセス リストとの一致数を正確に把握するための情報源として使用しないでください。

フラグメント制御付き拡張アクセス リスト

以前のリリースでは、非フラグメント パケットと、パケットの先頭フラグメントは、IP 拡張アクセス リストで処理していました（このアクセス リストを適用した場合）が、先頭以外のフラグメントはデフォルトで許可されていました。 ただし、フラグメント制御付き IP 拡張アクセス リスト機能により、パケットの先頭以外のフラグメントもさらにきめ細かく制御できるようになりました。 この機能を使用して、IP 拡張アクセス リストを適用するときに、パケットの先頭以外の IP フラグメントを調べるかどうかを指定できます。

先頭以外のフラグメントにはレイヤ 3 情報のみが含まれているため、レイヤ 3 情報のみが含まれるアクセス リスト エントリを先頭以外のフラグメントにも適用できるようになりました。 フラグメントにはフィルタリングに必要な情報がすべて揃っており、それでアクセス リスト エントリをパケットのフラグメントに適用できるというわけです。

この機能により、オプションの **fragments** キーワードが、IP アクセス リスト コマンドの **deny**（**IPv4**）、**permit**（**IPv4**）、**deny**（**IPv6**）、**permit**（**IPv6**）に追加されています。アクセス リスト エントリに **fragments** キーワードを指定することにより、その特定のアクセス リスト エントリは、パケットの先頭以外のフラグメントにのみ適用されます。フラグメントは、指定内容に応じて許可または拒否されます。

fragments キーワードの有無に応じたアクセス リスト エントリの動作をまとめると、次のようになります。

アクセス リスト エントリの状態	結果
fragments キーワードがなく、すべてのアクセス リスト エントリ情報が一致する	アクセス リスト エントリにレイヤ 3 情報のみが含まれている場合： • エントリは、非フラグメント パケット、先頭フラグメント、先頭以外のフラグメントに適用されます。 アクセス リスト エントリにレイヤ 3 情報とレイヤ 4 情報が含まれている場合： • エントリは、非フラグメント パケットと先頭フラグメントに適用されます。 ◦ エントリが一致し、かつ permit ステートメントである場合、パケットまたはフラグメントは許可されます。 ◦ エントリが一致し、かつ deny ステートメントである場合、パケットまたはフラグメントは拒否されます。 • エントリは、次の方法で先頭以外のフラグメントにも適用されます。 先頭以外のフラグメントにはレイヤ 3 情報のみが含まれているため、アクセス リスト エントリのレイヤ 3 の部分のみが適用されます。 アクセス リスト エントリのレイヤ 3 の部分が一致し、 ◦ エントリが **permit** ステートメントである場合、先頭以外のフラグメントは許可されます。 ◦ エントリが deny ステートメントの場合は、次のアクセス リスト エントリが処理されます。 （注） 先頭以外のフラグメントと非フラグメントや先頭フラグメントとでは、deny ステートメントの処理が異なることに注意してください。

アクセス リスト エントリの状態	結果
fragments キーワードがあり、すべてのアクセス リスト エントリ情報が一致する	アクセス リスト エントリは、先頭以外のフラグメントにのみ適用されます。 （注） レイヤ 4 情報を含むアクセス リスト エントリに **fragments** キーワードは設定できません。

すべてのアクセス リスト エントリに **fragments** キーワードを追加しないでください。IP パケットの先頭フラグメントは非フラグメントと見なされ、それ以降のフラグメントとは独立して扱われるためです。 先頭フラグメントは **fragments** キーワードが含まれているアクセス リスト permit エントリまたは deny エントリとは一致しないため、パケットは次のアクセス リスト エントリと比較されます。この比較は、**fragments** キーワードが含まれていないアクセス リスト エントリによってパケットが許可または拒否されるまで続きます。 したがって、deny エントリごとに、2 つのアクセス リスト エントリが必要になる場合があります。 ペアの最初の deny エントリには **fragments** キーワードは含まれず、初期フラグメントに適用されます。 ペアの 2 番めの deny エントリには **fragments** キーワードは含まれ、以降のフラグメントに適用されます。 同じホストに複数の **deny** アクセス リスト エントリがあり、それぞれのレイヤ 4 ポートが異なる場合、そのホストに追加する必要があるのは、**fragments** キーワードを指定した deny アクセス リスト エントリ 1 つだけです。 このように、パケットのすべてのフラグメントは、アクセス リストによって同様に扱われます。

IP データグラムのパケット フラグメントは個々のパケットと見なされ、各フラグメントはアクセス リスト アカウンティングとアクセス リスト違反カウントの 1 つのパケットとして個別にカウントされます。

（注） アクセス リストおよび IP フラグメントに関するあらゆるケースを **fragments** キーワードで解決できるわけではありません。

（注） ACL 処理の範囲内では、レイヤ 3 情報は、送信元、宛先、プロトコルなど、IPv4 ヘッダー内のフィールドを参照します。 レイヤ 4 情報は、TCP または UDP の送信元ポートおよび宛先ポート、TCP のフラグ、ICMP のタイプとコードなど、IPv4 ヘッダーの後に含まれるその他のデータを参照します。

ポリシー ルーティング

ポリシー ルーティングが **match ip address** コマンドに基づくものであり、アクセス リストのエントリがレイヤ 4 ～ レイヤ 7 の情報に一致した場合、フラグメンテーションとフラグメント制御機能はポリシー ルーティングに影響を及ぼします。 先頭フラグメントがポリシー ルーティングされなかった場合でも、先頭以外のフラグメントがアクセスリストを通過し、ポリシールーティングされることがあります。その逆もまた同じです。

前に説明したようにアクセス リスト エントリに **fragments** キーワードを使用すると、先頭フラグメントと先頭以外のフラグメントに対するアクションの照合を改善できるため、ポリシー ルーティングが想定どおりに機能する可能性が高くなります。

アクセス リストのエントリに関するコメント

remark アクセス リスト コンフィギュレーション コマンドを使用すると、名前付き IP アクセス リストにエントリに関するコメント（注釈）を含めることができます。 コメントを含めると、ネットワーク管理者がアクセス リストを理解し、精査しやすくなります。 1 つのコメント行の最大長は 255 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ようにしてください。 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、混乱を招きます。 コメントに順番を付けることができます。

アクセス リストの作成後、アクセス リストをインターフェイスまたは端末回線に適用することを忘れないでください。 詳細については、アクセス リストの適用, （18 ページ）を参照してください。

アクセス コントロール リスト カウンタ

Cisco IOS XR ソフトウェアでは、ACL カウンタがハードウェアとソフトウェアの両方で維持されます。 ハードウェア カウンタは、アクセス グループをインターフェイスに適用するなど、パケット フィルタリングの用途に使用します。 ソフトウェア カウンタは、主にソフトウェア パケット処理に関するあらゆる用途に使用できます。

パケット フィルタリングでは、ACE ごとに 64 ビットのハードウェア カウンタが使用されます。同じラインカードにある所定の方向のインターフェイスに同じアクセス グループを適用した場合、ACL のハードウェア カウンタは 2 つのインターフェイス間で共有されます。

特定のアクセス グループのハードウェア カウンタを表示するには、EXEC モードで **show access-lists ipv4** [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**location** *node-id*}] コマンドを使用します。

ハードウェア カウンタをクリアするには、EXEC モードで **clear access-list ipv4** *access-list-name* [**hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**location** *node-id*}] コマンドを使用します。

わずかながらパフォーマンスが低下するため、IPv4 ACL に対するハードウェア カウントはデフォルトでは無効になっています。 ハードウェア カウントをイネーブルにするには、インターフェイス コンフィギュレーション モードで **ipv4 access-group** *access-list-name* {**ingress** | **egress**} [**hardware-count**] コマンドを使用します。 このコマンドは必要に応じて使用できるため、カウントは指定したインターフェイスに対してのみイネーブルになります。

ソフトウェア カウンタは、ソフトウェアがパケットを処理すると更新されます。たとえば、例外パケットを LC CPU にパントして処理した場合や、ルーティング プロトコルが ACL を使用した

場合などです。 維持されるソフトウェア カウンタというのは、その ACL を使用するすべてのソフトウェア アプリケーションの集合体です。 ソフトウェア専用の ACL カウンタを表示するには、EXEC モードで **show access-lists ipv4** *access-list-name* [**sequence** *number*] コマンドを使用します。

ここに挙げた情報は、ハードウェア カウントが常にイネーブルになっていることを除いて、すべて IPv6 にも当てはまります。IPv6 アクセス グループのコマンドライン インターフェイス（CLI）には **hardware-count** オプションがありません。

プレフィックス リストを使用した BGP フィルタリング

プレフィックス リストは、BGP ルート フィルタリング コマンドの多くでアクセス リストの代わりに使用できます。 プレフィックス リストを使用した場合の利点は次のとおりです。

- サイズの大きなリストをロードしてルート ルックアップを実施する場合のパフォーマンスが大幅に向上します。
- 差分更新がサポートされます。
- CLI の使い勝手が向上します。 アクセス リストを使用して BGP 更新をフィルタリングするための CLI は、パケット フィルタリング形式を使用しているため、わかりにくく使い勝手もよくありません。
- 柔軟性が高まります。

コマンドでプレフィックス リストを使用するには、あらかじめプレフィックス リストをセットアップしておく必要があります。プレフィックス リストのエントリには、シーケンス番号を割り当ててください。

プレフィックス リストでトラフィックをフィルタリングする仕組み

プレフィックス リストによるフィルタリングでは、ルートのプレフィックスが、プレフィックス リストに記載されているプレフィックスと照合されます。 一致すると、一致したルートが使用されます。 具体的には、プレフィックスを許可するか、拒否するかは次のルールに基づきます。

- 空のプレフィックス リストはすべてのプレフィックスを許可します。
- 特定のプレフィックスがプレフィックス リストのどのエントリとも一致しなかった場合、暗黙の deny が適用されます。
- プレフィックス リストの複数のエントリが特定のプレフィックスと一致したときは、最も長く、最も具体的な一致が選択されます。

シーケンス番号は自動的に生成されます。ただし、この自動生成をディセーブルにしている場合を除きます。 シーケンス番号の自動生成をディセーブルにしている場合は、IPv4 または IPv6 のプレフィックス リスト コンフィギュレーション コマンドの **permit** コマンドおよび **deny** コマンドで *sequence-number* 引数を使用して、各エントリのシーケンス番号を指定する必要があります。プレフィックス リストのエントリを削除するには、*sequence-number* 引数を指定した **permit** コマンドまたは **deny** コマンドの **no** 形式を使用してください。

show コマンドの出力には、シーケンス番号が含まれます。

ACL ベース転送の実装に関する情報

アクセスリストおよびプレフィックスリストを実装するには、次の概念を理解する必要があります。

ACL ベース転送の概要

統合ネットワークは、音声、ビデオ、およびデータを伝送します。 トラフィックによっては、ルーティング プロトコルが算出したパスを使用するのではなく、特定のパスにルーティングすることが必要になる場合があります。 これを実現するための簡単なソリューションは、ACL設定にネクストホップ アドレスを指定することです。これで、パケットベースで宛先アドレスをルックアップするのではなく、ACLに設定したネクストホップアドレスを使用して指定の宛先にパケットを転送できるようになります。 ACL設定でネクストホップを使用して転送するというこの機能は、ACL ベース転送（ABF）と呼ばれます。

ACL ベース転送を使用すると、ブロードキャスト TV over IP、IP テレフォニー、データなどを対象としたサービスを複数のプロバイダーから選択することが可能になり、カフェテリア形式でインターネットにアクセスできます。 サービス プロバイダーは、ユーザ トラフィックをさまざまなコンテンツ プロバイダーに迂回させることができます。

ABF-OT

ユーザが適切なネクストホップを柔軟に選択できるようにするため、ABFの機能が強化され、オブジェクト トラッキング（OT）と情報をやり取りできるようになりました。これは、次の機能に影響を及ぼします。

- CEF でのプレフィックスのトラッキング
- ライン ステート プロトコルのトラッキング
- IPSLA（IP サービス レベル契約）

オブジェクト トラッキングでの IPSLA のサポート

OT モジュールは、IPSLA モジュールとやり取りして到達可能性情報を取得します。 ルータは、IPSLA を使って定期的に測定を実施します。

アクセスリストおよびプレフィックスリストの実装方法

Cisco ASR 9000 SIP 700 ラインカードおよび ASR 9000 イーサネット ラインカードで IPv6 ACL をサポートするようになりました。 これに関連する基準は次のとおりです。

- ACL 対応のインターフェイス：1000（各方向 500 ずつ）、ASR 9000 イーサネット ラインカードの場合は 4000
- 一意の ACL：512（それぞれに 5 個の ACE）、ASR 9000 イーサネット ラインカードの場合は 2000
- ACL あたりの最大 ACE 数：8000（ASR 9000 イーサネット ラインカードの場合は、LC モデルに基づいて 16000、8000、4000 のいずれか）
- IPv6 ACL ログも、今後サポートする予定です。

ここでは、次の手順について説明します。

拡張アクセス リストの設定

このタスクでは、拡張 IPv4 または IPv6 アクセス リストを設定します。

手順の概要

1. **configure**
2. {**ipv4** | **ipv6**} **access-list** *name*
3. [*sequence-number*] **remark** *remark*
4. 次のいずれかを実行します。
   - [*sequence-number*]{**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
   - [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [*routing*] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
5. 必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
7. **show access-lists** {**ipv4** | **ipv6**} [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

ステップ 2	{**ipv4** | **ipv6**} **access-list** *name*
	
	例：
	
	`RP/0/RSP0/CPU0:router(config)# ipv4 access-list acl_1`
	または
	
	`RP/0/RSP0/CPU0:router(config)# ipv6 access-list acl_2`
		IPv4 または 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、名前付きアクセス リストを設定します。

ステップ 3	[*sequence-number*] **remark** *remark* 例： `RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 remark Do not allow user1 to telnet out`	（任意）名前付きのアクセス リストに **permit** ステートメントまたは **deny** ステートメントに関するコメントを書くことができます。 • 注釈は最大 255 文字まで可能で、これより長い文字は切り捨てられます。 • **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も設定できますが、どちらかの位置に統一することを推奨します。
ステップ 4	次のいずれかを実行します。 • [*sequence-number*]{**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**] • [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [*routing*] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]	IPv4 アクセス リスト acl_1 に許可または拒否の条件を 1 つ以上指定します。 • 任意の **log** キーワードを指定すると、エントリに一致したパケットに関する情報ログ メッセージがコンソールに送信されます。 • 任意の **log-input** キーワードは、ログ メッセージに入力インターフェイスも含まれることを除いて、**log** キーワードと同じように機能します。 または IPv6 アクセス リスト acl_2 に許可または拒否の条件を 1 つ以上指定します。 • IPv6 オプション ヘッダーおよび任意の上位層プロトコル タイプ情報に基づいて IPv6 トラフィックをフィルタリングする方法の詳細については、**deny**（IPv6）コマンドおよび **permit**（IPv6）コマンドを参照してください。

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit 172.16.0.0 0.0.255.255`
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 deny 192.168.34.0 0.0.0.255`
	または
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 20 permit icmp any any`
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 30 deny tcp any any gt 5000`
		（注） どの IPv6 アドレス リストにも、ネイバー アドバタイズメントおよび送信要求に使用される暗黙の permit 2 つあります。それは暗黙的ネイバー探索ネイバーアドバイタイズメント（NDNA）と暗黙的ネイバー探索ネイバー送信要求（NDNS）です。
		（注） どの IPv6 アクセス リストにも最後の一致条件として暗黙の **deny ipv6 any any** ステートメントがあります。1 つの IPv6 アクセス リストには、暗黙の **deny ipv6 any any** ステートメントを有効にするために少なくとも 1 つのエントリが含まれる必要があります。

ステップ 5	必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。	アクセス リストは変更できます。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ステップ 7	**show access-lists** {**ipv4** | **ipv6**} [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location**	（任意）現在の IPv4 または IPv6 アクセス リストの内容を表示します。

	コマンドまたはアクション	目的
	node-id} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]] 例： `RP/0/RSP0/CPU0:router# show access-lists ipv4 acl_1`	• 特定のアクセス リストの内容を表示するには、*access-list-name* 引数を使用します。 • 方向（入力または出力）とアクセスリストを指定して、それを使用するすべてのインターフェイスのハードウェアの内容とカウンタを表示するには、**hardware**、**ingress** または **egress**、および **location** または **sequence** の各キーワードを使用します。 インターフェイスのアクセス グループを設定するには、イネーブルにするアクセス リスト ハードウェア カウンタに対して **ipv4 access-group** コマンドを使用します。 • 現在の IPv4 または IPv6 アクセス リストをまとめたサマリーを表示するには、**summary** キーワードを使用します。 • インターフェイスの統計情報を表示するには、**interface** キーワードを使用します。

次の作業

アクセス リストを作成したら、回線またはインターフェイスに適用する必要があります。アクセス リストを適用する方法については、アクセス リストの適用, （18 ページ）を参照してください。

一意のアクセス リスト エントリ（ACE）の追加または削除中に、ACL コミットが失敗します。これは、割り当てられたマネージャ プロセスが存在しないために発生します。 config-ipv4-acl モードを終了してコンフィギュレーション モードに戻り、再び config-ipv4-acl モードを開始してから、最初の ACE を追加してください。

アクセス リストの適用

作成したアクセス リストを機能させるには、そのアクセス リストを参照する必要があります。アクセスリストは、発信インターフェイスまたは着信インターフェイスに適用できます。 ここでは、端末回線とネットワーク インターフェイスの両方に対してこのタスクを実行するためのガイドラインを示します。

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、すべての仮想端末回線に同じ制約を設定する必要があります。

着信アクセス リストの場合、パケットの受信後、Cisco IOS XR ソフトウェアはアクセス リストに照らしてそのパケットの送信元アドレスをチェックします。アクセスリストがアドレスを許可している場合は、パケットの処理を継続します。 アクセス リストがアドレスを拒否している場合

は、パケットを廃棄し、ICMPホスト到達不能メッセージを返します。 ICMPメッセージは設定可能です。

発信アクセス リストの場合、パケットを受信して管理下のインターフェイスに転送した後、アクセス リストに照らしてパケットの送信元アドレスをチェックします。 アクセス リストがアドレスを許可している場合は、パケットを送信します。アクセスリストがアドレスを拒否している場合は、パケットを廃棄し、ICMP ホスト到達不能メッセージを返します。

まだ定義されていないアクセスリストをインターフェイスに適用すると、アクセスリストがまだインターフェイスに適用されていないものと解釈し、すべてのパケットを容認します。 ネットワークで未定義のアクセス リストをセキュリティの手段として使用する場合は、この動作に留意してください。

インターフェイスへのアクセスの制御

このタスクでは、アクセス リストをインターフェイスに適用して、そのインターフェイスへのアクセスを制限します。

アクセス リストは、発信インターフェイスまたは着信インターフェイスに適用できます。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. 次のいずれかを実行します。
   - **ipv4 access-group** *access-list-name* {**ingress** | **egress**} [**hardware-count**] [**interface-statistics**]
   - **ipv6 access-group** *access-list-name* {**ingress** | **egress**} [**interface-statistics**]
4. 次のいずれかを実行します。
   - end
   - commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	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 2	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/2/0/2`	インターフェイスを設定し、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。 • *type* 引数には、インターフェイス タイプを指定します。 インターフェイス タイプの詳細については、疑問符（?）オンライン ヘルプ機能を使用してください。 • *instance* 引数には、物理インターフェイス インスタンスまたは仮想インスタンスを指定します。 ◦物理インターフェイス インスタンスの表記方法は *rack/slot/module/port* です。 値を区切るスラッシュ（/）は、表記の一部として必要です。 ◦仮想インターフェイス インスタンスの数値範囲は、インターフェイス タイプによって異なります。

ステップ 3	次のいずれかを実行します。
	• **ipv4 access-group** *access-list-name* {**ingress** | **egress**} [**hardware-count**] [**interface-statistics**]
	• **ipv6 access-group** *access-list-name* {**ingress** | **egress**} [**interface-statistics**]
	
	例：
	`RP/0/RSP0/CPU0:router(config-if)# ipv4 access-group p-in-filter in`
	
	`RP/0/RSP0/CPU0:router(config-if)# ipv4 access-group p-out-filter out`
		インターフェイスへのアクセスを制御します。
		• 特定の IPv4 または IPv6 アクセス リストを指定するには、*access-list-name* 引数を使用します。
		• 着信パケットをフィルタリングするには **in** キーワードを使用し、発信パケットをフィルタリングするには **out** キーワードを使用します。
		• IPv4 アクセス グループのハードウェア カウンタをイネーブルにするには、**hardware-count** キーワードを使用します。
		◦IPv6 アクセス グループのハードウェア カウンタは、自動的にイネーブルになります。
		• ハードウェアにインターフェイスごとの統計情報を指定するには、**interface-statistics** キーワードを使用します。
		
		この例では、GigabitEthernet 0/2/0/2 から発着信されるパケットにフィルタを適用します。

ステップ 4	次のいずれかを実行します。 • end • commit 例： `RP/0/RSP0/CPU0:router(config-if)# end`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:`

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config-if)# commit`	◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 ◦ **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

回線へのアクセスの制御

このタスクでは、回線にアクセス リストを適用して、その回線へのアクセスを制御します。

手順の概要

1. **configure**
2. **line** {**aux** | **console** | **default** | **template** *template-name*}
3. **access-class** *list-name*{**ingress** | **egress**}
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	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 2	**line** {**aux** | **console** | **default** | **template** *template-name*} 例： `RP/0/RSP0/CPU0:router(config)# line default`	補助、コンソール、デフォルト、またはユーザ定義の回線テンプレートを指定し、回線テンプレート コンフィギュレーション モードを開始します。 • ライン テンプレートは、物理端末回線接続（コンソール ポートおよび AUX ポート）および VTY 接続を設定して管理するために使用する属性のコレクションです。 Cisco IOS XR ソフトウェアでは、次のテンプレートを使用できます。 ◦ 補助回線テンプレート：補助回線に適用される回線テンプレート。 ◦ コンソール ライン テンプレート：コンソール回線に適用されます。 ◦ デフォルト ライン テンプレート：物理および仮想端末回線に適用されます。 ◦ ユーザ定義ライン テンプレート：仮想端末回線の範囲に適用できます。
ステップ 3	**access-class** *list-name*{**ingress** | **egress**} 例： `RP/0/RSP0/CPU0:router(config-line)# access-class acl_2 out`	IPv4 または IPv6 アクセス リストを使用して、着信接続および発信接続を制限します。 • 例では、IPv6 アクセス リスト acl_2 を使用して、デフォルトの回線テンプレートの発信接続をフィルタリングしています。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

	コマンドまたはアクション	目的
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

プレフィックス リストの設定

このタスクでは、IPv4 または IPv6 プレフィックス リストを設定します。

手順の概要

1. configure
2. {**ipv4** | **ipv6**} **prefix-list** *name*
3. [*sequence-number*] **remark** *remark*
4. [*sequence-number*] {**permit** | **deny**} *network/length* [**ge** *value*] [**le** *value*] [**eq** *value*]
5. 必要に応じてステップ 4 を繰り返します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。
6. 次のいずれかを実行します。
   - end
   - commit
7. 次のいずれかを実行します。
   - **show prefix-list ipv4** [*name*] [*sequence-number*]
   - **show prefix-list ipv6** [*name*] [*sequence-number*] [*summary*]
8. **clear** {**ipv4** | **ipv6**} **prefix-list** *name* [*sequence-number*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure 例： `RP/0/RSP0/CPU0:router# configure`	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 2	{**ipv4** | **ipv6**} **prefix-list** *name*
	
	例：
	`RP/0/RSP0/CPU0:router(config)# ipv4 prefix-list pfx_1`
	または
	`RP/0/RSP0/CPU0:router(config)# ipv6 prefix-list pfx_2`
		IPv4 または IPv6 プレフィックス リスト コンフィギュレーションモードを開始し、名前付きプレフィックス リストを設定します。
		• プレフィックス リストを作成するには、少なくとも 1 つの **permit** 句または **deny** 句を入力する必要があります。
		• プレフィックス リストのエントリをすべて削除するには、**no** {**ipv4** | **ipv6**} **prefix-list** *name* コマンドを使用します。

ステップ 3	[*sequence-number*] **remark** *remark*
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv4_pfx)# 10 remark Deny all routes with a prefix of 10/8`
	`RP/0/RSP0/CPU0:router(config-ipv4_pfx)# 20 deny 10.0.0.0/8 le 32`
		（任意）名前付きのプレフィックス リストに次の **permit** ステートメントまたは **deny** ステートメントに関するコメントを書くことができます。
		• 注釈は最大 255 文字まで可能で、これより長い文字は切り捨てられます。
		• **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も設定できますが、どちらかの位置に統一することを推奨します。

ステップ 4	[*sequence-number*] {**permit** | **deny**} *network/length* [**ge** *value*] [**le** *value*] [**eq** *value*] 例： `RP/0/RSP0/CPU0:router(config-ipv6_pfx)# 20 deny 128.0.0.0/8 eq 24`	名前付きプレフィックス リストに許可または拒否の条件を 1 つ以上指定します。 • この例では、プレフィックス リスト pfx_2 の 128.0.0.0/8 の /24 に一致するプレフィックスをすべて拒否します。
ステップ 5	必要に応じてステップ 4 を繰り返します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。	プレフィックス リストは変更できます。

ステップ 6	次のいずれかを実行します。
	• end
	• commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv6_pfx)# end`
	または
	`RP/0/RSP0/CPU0:router(config-ipv6_pfx)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:`
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレー

	コマンドまたはアクション	目的
		ション セッションが終了して、ルータが EXEC モードに戻ります。 ◦ **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ステップ 7	次のいずれかを実行します。
	• **show prefix-list ipv4** [*name*] [*sequence-number*]
	• **show prefix-list ipv6** [*name*] [*sequence-number*] [*summary*]
	例：
	`RP/0/RSP0/CPU0:router# show prefix-list ipv4 pfx_1`
	または
	`RP/0/RSP0/CPU0:router# show prefix-list ipv6 pfx_2 summary`
		（任意）現在の IPv4 または IPv6 プレフィックス リストの内容を表示します。
		• 特定のプレフィックス リストの内容を表示するには、*name* 引数を使用します。
		• プレフィックス リスト エントリのシーケンス番号を指定するには、*sequence-number* 引数を使用します。
		• プレフィックス リストの内容のサマリーを表示するには、**summary** キーワードを使用します。

ステップ 8	**clear** {**ipv4** | **ipv6**} **prefix-list** *name* [*sequence-number*] **例**： `RP/0/RSP0/CPU0:router# clear prefix-list ipv4 pfx_1 30`	（任意）IPv4 または IPv6 プレフィックス リストのヒット カウントをクリアします。 （注） ヒット カウントは、特定のプレフィックス リスト エントリに一致する数を示す値です。

標準アクセス リストの設定

このタスクでは、標準 IPv4 アクセス リストを設定します。

標準アクセス リストでは、照合操作に送信元アドレスを使用します。

手順の概要

1. **configure**
2. **ipv4 access-list** *name*
3. [*sequence-number*] **remark** *remark*
4. [*sequence-number*] {**permit** | **deny**} *source* [*source-wildcard*] [**log** | **log-input**]
5. 必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。
6. 次のいずれかを実行します。
   - end
   - commit
7. **show access-lists** [**ipv4** | **ipv6**] [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ipv4 access-list** *name* 例： `RP/0/RSP0/CPU0:router# ipv4 access-list acl_1`	IPv4 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、アクセス リスト acl_1 を設定します。
ステップ 3	[*sequence-number*] **remark** *remark* 例： `RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 remark Do not allow user1 to telnet out`	（任意）名前付きのアクセス リストに次の **permit** ステートメントまたは **deny** ステートメントに関するコメントを書くことができます。 • 注釈は最大 255 文字まで可能で、これより長い文字は切り捨てられます。 • **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も設定できますが、どちらかの位置に統一することを推奨します。

	コマンドまたはアクション	目的

ステップ 4	[*sequence-number*] {**permit** | **deny**} *source* [*source-wildcard*] [**log** | **log-input**]
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 permit 172.16.0.0 0.0.255.255`
	または
	`RRP/0/RSP0/CPU0:routerrouter(config-ipv4-acl)# 30 deny 192.168.34.0 0.0.0.255`
		パケットの通過またはドロップを決定する許可または拒否の条件を 1 つ以上指定します。
		• パケットの送信元のネットワークまたはホストの番号を指定するには、*source* 引数を使用します。
		• 送信元に適用するワイルドカード ビットを指定するには、任意の *source-wildcard* 引数を使用します。
		• 任意の **log** キーワードを指定すると、エントリに一致したパケットに関する情報ログ メッセージがコンソールに送信されます。
		• 任意の **log-input** キーワードは、ログ メッセージに入力インターフェイスも含まれることを除いて、**log** キーワードと同じように機能します。

ステップ 5	必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。エントリを削除するには、**no** *sequence-number* コマンドを使用します。	アクセス リストは変更できます。

ステップ 6	次のいずれかを実行します。
	• end
	• commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# end`
	または
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:`
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

	コマンドまたはアクション	目的
		• 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 7	**show access-lists** [**ipv4** | **ipv6**] [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]] 例： `RP/0/RSP0/CPU0:router# show access-lists ipv4 acl_1`	（任意）名前付き IPv4 アクセス リストの内容を表示します。 • IPv4 標準アクセス リストの内容は、拡張アクセス リスト形式で表示されます。

次の作業

標準アクセス リストの作成後、それを回線またはインターフェイスに適用する必要があります。アクセス リストを適用する方法については、アクセス リストの適用，（18 ページ）を参照してください。

アクセス リストのコピー

このタスクでは、IPv4 または IPv6 アクセス リストをコピーします。

手順の概要

1. **copy access-list** {**ipv4** | **ipv6**}**source-acl destination-acl**
2. **show access-lists** {**ipv4** | **ipv6**}[*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]]

手順の詳細

	コマンドまたはアクション	目的
ステップ1	**copy access-list** {**ipv4** | **ipv6**}**source-acl destination-acl** 例： `RP/0/RSP0/CPU0:router# copy ipv6 access-list list-1 list-2`	既存の IPv4 または IPv6 アクセス リストのコピーを作成します。 • コピーするアクセス リストの名前を指定するには、*source-acl* 引数を使用します。 • 送信元アクセス リストの内容のコピー先を指定するには、*destination-acl* 引数を使用します。 ◦ *destination-acl* 引数は一意の名前である必要があります。アクセス リストに *destination-acl* 引数名が存在する場合、そのアクセス リストはコピーされません。
ステップ2	**show access-lists** {**ipv4** | **ipv6**}[*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]] 例： `RP/0/RSP0/CPU0:router# show access-lists ipv4 list-2`	（任意）名前付きの IPv4 または IPv6 アクセス リストの内容を表示します。たとえば、コピー先の内容を検証して、宛先アクセス リスト list-2 に送信元アクセス リスト list-1 の情報がすべて含まれていることを確認できます。

アクセス リスト エントリの順序付けとアクセス リストの変更

このタスクでは、名前付きアクセス リストのエントリにシーケンス番号を割り当てる方法と、アクセス リストに対してエントリの追加または削除を行う方法について説明します。 アクセス リストを変更することを前提に説明します。 アクセス リストの並べ替えは任意です。

手順の概要

1. **resequence access-list** {**ipv4** | **ipv6**} *name* [*base* [*increment*]]
2. configure
3. {**ipv4** | **ipv6**} **access-list** *name*
4. 次のいずれかを実行します。
   - [*sequence-number*] {**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
   - [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [**routing**] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
5. 必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。
6. 次のいずれかを実行します。
   - end
   - commit
7. **show access-lists** [**ipv4** | **ipv6**] [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**resequence access-list** {**ipv4** | **ipv6**} *name* [*base* [*increment*]] 例： `RP/0/RSP0/CPU0:router# resequence access-list ipv4 acl_3 20 15`	（任意）開始シーケンス番号と、シーケンス番号の増分値を使用して、指定した IPv4 または IPv6 アクセス リストを並べ替えます。 • この例では、acl_3 という名前の IPv4 アクセス リストを並べ替えます。 開始シーケンス番号は 20、増分は 15 です。 増分値を選択しないと、デフォルトの増分値 10 が使用されます。
ステップ 2	configure 例： `RP/0/RSP0/CPU0:router# configure`	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	{**ipv4** | **ipv6**} **access-list** *name*
	
	例：
	`RP/0/RSP0/CPU0:router(config)# ipv4 access-list acl_1`
	または
	`RP/0/RSP0/CPU0:router(config)# ipv6 access-list acl_2`
		IPv4 または 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、名前付きアクセス リストを設定します。

ステップ 4	次のいずれかを実行します。
	• [*sequence-number*] {**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
	• [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [**routing**] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit 172.16.0.0 0.0.255.255`
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 deny 192.168.34.0 0.0.0.255`
	または
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 20 permit icmp any any`
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 30 deny tcp any any gt 5000`
		IPv4 アクセス リスト acl_1 に許可または拒否の条件を 1 つ以上指定します。
		• 任意の **log** キーワードを指定すると、エントリに一致したパケットに関する情報ログ メッセージがコンソールに送信されます。
		• 任意の **log-input** キーワードは、ログ メッセージに入力インターフェイスも含まれることを除いて、**log** キーワードと同じように機能します。
		• このアクセス リストでは **permit** ステートメントを最初に使用していますが、必要なステートメントの順序に応じて、**deny** ステートメントが最初に使用される可能性もあります。
		
		または
		IPv6 アクセス リスト acl_2 に許可または拒否の条件を 1 つ以上指定します。
		• IPv6 オプション ヘッダーと、ICMP、TCP、UDP などの上位層プロトコルに基づいて IPv6 トラフィックをフィルタリングする方法の詳細については、**permit**（IPv6）コマンドおよび **deny**（IPv6）コマンドを参照してください。
		
		（注） どの IPv6 アクセス リストにも最後の一致条件として暗黙の **deny ipv6 any any** ステートメントがあります。 1 つの IPv6 アクセス リストには、暗黙の **deny ipv6 any any** ステートメントを有効にするために少なくとも 1 つのエントリが含まれる必要があります。

ステップ 5	必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加し	アクセス リストは変更できます。

	コマンドまたはアクション	目的
	ます。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。	

ステップ 6	次のいずれかを実行します。
	• end
	• commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# end`
	または
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:`
		
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		
		• 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ステップ 7	**show access-lists** [**ipv4** | **ipv6**] [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]] 例： `RP/0/RSP0/CPU0:router# show access-lists ipv4 acl_1`	（任意）名前付きの IPv4 または IPv6 アクセス リストの内容を表示します。 • 出力をレビューして、アクセス リストに最新情報が含まれていることを確認します。

次の作業

アクセス リストがまだインターフェイスまたは回線に適用されていないか、または他の方法で参照されている場合は、アクセス リストを適用します。 アクセス リストを適用する方法については、アクセス リストの適用，（18 ページ）を参照してください。

プレフィックス リストのコピー

このタスクでは、IPv4 または IPv6 プレフィックス リストをコピーします。

手順の概要

1. **copy prefix-list** {**ipv4** | **ipv6**} *source-name destination-name*
2. 次のいずれかを実行します。
   - **show prefix-list ipv4** [*name*] [*sequence-number*]
   - **show prefix-list ipv6** [*name*] [*sequence-number*] [**summary**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**copy prefix-list** {**ipv4** | **ipv6**} *source-name destination-name* 例： `RP/0/RSP0/CPU0:router# copy prefix-list ipv6 list_1 list_2`	既存の IPv4 または IPv6 プレフィックス リストのコピーを作成します。 • コピーするプレフィックス リストの名前を指定するには *source-name* 引数を使用し、コピー元のプレフィックス リストの内容のコピー先を指定するには、*destination-name* 引数を使用します。 • *destination-name* 引数は、一意の名前である必要があります。*destination-name* 引数名がプレフィックス リストに存在する場合、そのプレフィックス リストはコピーされません。
ステップ 2	次のいずれかを実行します。 • **show prefix-list ipv4** [*name*] [*sequence-number*] • **show prefix-list ipv6** [*name*] [*sequence-number*] [**summary**]	（任意）現在の IPv4 または IPv6 プレフィックス リストの内容を表示します。 • 出力をレビューして、プレフィックス リスト list_2 に list_1 のエントリが含まれていることを確認します。

	コマンドまたはアクション	目的
	例： `RP/0/RSP0/CPU0:router# show prefix-list ipv6 list_2`	

プレフィックス リスト エントリの順序付けとプレフィックス リストの変更

このタスクでは、名前付きプレフィックス リストのエントリにシーケンス番号を割り当てる方法と、プレフィックス リストに対してエントリの追加または削除を行う方法について説明します。プレフィックス リストを変更することを前提に説明します。 プレフィックス リストの並べ替えは任意です。

はじめる前に

（注） IPv6 プレフィックス リストの並べ替えはサポートされません。

手順の概要

1. **resequence prefix-list ipv4** *name* [*base* [*increment*]]
2. configure
3. {**ipv4** | **ipv6**} **prefix-list** *name*
4. [*sequence-number*] {**permit** | **deny**} *network/length* [**ge** *value*] [**le** *value*] [**eq** *value*]
5. 必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。
6. 次のいずれかを実行します。
   - end
   - commit
7. 次のいずれかを実行します。
   - **show prefix-list ipv4** [*name*] [*sequence-number*]
   - **show prefix-list ipv6** [*name*] [*sequence-number*] [**summary**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**resequence prefix-list ipv4** *name* [*base* [*increment*]] 例： `RP/0/RSP0/CPU0:router# resequence prefix-list ipv4 pfx_1 10 15`	（任意）開始シーケンス番号と、シーケンス番号の増分値を使用して、指定したIPv4プレフィックス リストを並べ替えます。 • この例では、pfx_1 というプレフィックス リストを並べ替えます。 開始シーケンス番号は 10、増分は 15 です。
ステップ 2	configure 例： `RP/0/RSP0/CPU0:router# configure`	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	{**ipv4** | **ipv6**} **prefix-list** *name* 例： `RP/0/RSP0/CPU0:router(config)# ipv6 prefix-list pfx_2`	IPv4またはIPv6プレフィックス リスト コンフィギュレーション モードを開始し、名前付きプレフィックス リストを設定します。
ステップ 4	[*sequence-number*] {**permit** | **deny**} *network/length* [**ge** *value*] [**le** *value*] [**eq** *value*] 例： `RP/0/RSP0/CPU0:router(config-ipv6_pfx)# 15 deny 128.0.0.0/8 eq 24`	名前付きプレフィックス リストに許可または拒否の条件を 1 つ以上指定します。
ステップ 5	必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。	プレフィックス リストは変更できます。

ステップ 6	次のいずれかを実行します。
	• end
	• commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv6_pfx)# end`
	または
	`RP/0/RSP0/CPU0:router(config-ipv6_pfx)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:`
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXEC モードに戻ります。

	コマンドまたはアクション	目的
		◦ **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 ◦ **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。
ステップ 7	次のいずれかを実行します。 • **show prefix-list ipv4** [*name*] [*sequence-number*] • **show prefix-list ipv6** [*name*] [*sequence-number*] [**summary**] 例： `RP/0/RSP0/CPU0:router# show prefix-list ipv6 pfx_2`	（任意）現在の IPv4 または IPv6 プレフィックス リストの内容を表示します。 • 出力をレビューして、プレフィックス リスト pfx_2 に新しい情報がすべて含まれていることを確認します。

ACL ベース転送を実装する方法

ここでは、次の手順について説明します。

セキュリティ ACL での ACL ベース転送の設定

セキュリティ ACL で ACL ベース転送を設定するには、次のタスクを実行します。

手順の概要

1. configure
2. **ipv4 access-list** *name*
3. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**[default] nexthop1** [**ipv4** *ipv4-address1*] **nexthop2**[**ipv4** *ipv4-address2*] **nexthop3**[**ipv4** *ipv4-address3*]] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**] [[**track** *track-name*] [**ttl** *ttl* [*value1 ... value2*]]
4. 次のいずれかを実行します。
   - end
   - commit
5. **show access-list ipv4** [[*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ipv4 access-list** *name* 例： `RP/0/RSP0/CPU0:router(config)# ipv4 access-list security-abf-acl`	IPv4 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、指定したアクセス リストを設定します。

ステップ 3	[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**[default] nexthop1** [**ipv4** *ipv4-address1*] **nexthop2**[**ipv4** *ipv4-address2*] **nexthop3**[**ipv4** *ipv4-address3*]] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log**	**log-input**] [[**track** *track-name*] [**ttl** *ttl* [*value1 ... value2*]]
		
		例：
		`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 10.0.0.0 0.255.255.255 any nexthop 50.1.1.2`
		`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 15 permit ipv4 30.2.1.0 0.0.0.255 any`
		`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 permit ipv4 30.2.0.0 0.0.255.255 any nexthop 40.1.1.2`
			IPv4アクセス リストの条件を設定します。設定例では、セキュリティ ACL で ACL ベース転送を設定する方法を示しています。
			• **nexthop1**、**nexthop2**、**nexthop3** キーワードは、このエントリに指定されたネクスト ホップに転送します。
			• **default** キーワードを設定すると、ACL ベースの転送アクションが実行されるのは、パケットの宛先のPLUルックアップの結果によりデフォルト ルートを決定する場合、つまり、パケット宛先のルートを指定しない場合だけとなります。

	コマンドまたはアクション	目的
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 25 permit ipv4 any any`	

ステップ 4	次のいずれかを実行します。
	• end
	• commit
	例：
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# end`
	または
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:`
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ステップ 5	**show access-list ipv4** [[*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]] **例**： `RP/0/RSP0/CPU0:router# show access-lists ipv4 security-abf-acl`	ACL ソフトウェアに関する情報を表示します。

IPSLA-OT の実装

ここでは、次の手順について説明します。

トラック モードのイネーブル化

手順の概要

1. configure
2. **track** *track-name*
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**track** *track-name* 例： `RP/0/RSP0/CPU0:router(config)# track t1`	トラック コンフィギュレーション モードを開始します。
ステップ 3	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

トラック タイプの設定

ネクストホップ デバイスの可用性をトラッキングするメカニズムには、さまざまな種類があります。 トラッキング タイプには 4 つのタイプがあり、次のものを使用します。

- 回線プロトコル
- リスト
- ルート
- IPSLA

トラッキング タイプの設定（回線プロトコル）

回線プロトコルは、オブジェクトトラッカーコンポーネントがトラッキングできるオブジェクトタイプの 1 つです。 このオブジェクト タイプでは、インターフェイスからの状態変化通知をトラッキングするためのオプションを利用できます。 インターフェイス状態変化通知に基づいて、トラック状態を UP にするか、DOWN にするかを決定します。

手順の概要

1. configure
2. **track** *track-name*
3. **type line-protocol state interface** *type interface-path-id*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**track** *track-name* 例： `RP/0/RSP0/CPU0:router(config)# track t1`	トラック 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	**type line-protocol state interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-track)# type line-protocol state interface tengige 0/4/4/0`	状態変化通知のためにトラッキングする必要があるインターフェイスを設定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。

	コマンドまたはアクション	目的
		◦cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

トラック タイプ（リスト）の設定

リストは、ブール オブジェクト タイプです。 ブールとは、オブジェクト トラッカーでサポートされているさまざまなオブジェクト タイプの組み合わせに対して、ブール AND 演算またはブール OR 演算を実行する機能のことです。

手順の概要

1. configure
2. **track** *track-name*
3. **type list boolean and**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**track** *track-name* 例： `RP/0/RSP0/CPU0:router(config)# track t1`	トラック 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	**type list boolean and** 例： `RP/0/RSP0/CPU0:router(config-track)# type list boolean and`	ブール AND 演算またはブール OR 演算を実行できるトラック オブジェクトのリストを設定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

トラッキング タイプ（ルート）の設定

ルートは、ルート オブジェクト タイプです。 オブジェクト トラッカーは、FIB 通知をトラッキングして、ルート到達可能性およびトラック状態を判断します。

手順の概要

1. configure
2. **track** *track-name*
3. **type route reachability**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**track** *track-name* 例： `RP/0/RSP0/CPU0:router(config)# track t1`	トラック 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	**type route reachability** 例： `RP/0/RSP0/CPU0:router(config-track)# type route reachability`	到達可能性状態を動的に学習する必要があるルートを設定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション

	コマンドまたはアクション	目的
		セッションは終了せず、設定変更もコミットされません。 • 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

トラッキング タイプの設定（rtr）

IPSLA は、ipsla オブジェクト タイプです。 オブジェクト トラッカーは、ipsla 操作の戻りコードをトラッキングして、トラック状態の変化を判断します。

手順の概要

1. configure
2. **track** *track-name*
3. **type rtr** *ipsla operation id* **reachability**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**track** *track-name* 例： `RP/0/RSP0/CPU0:router(config)# track t1`	トラック 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	**type rtr** *ipsla operation id* **reachability** 例： `RP/0/RSP0/CPU0:router`**`type rtr 100 reachability`**	到達可能性のためにトラッキングする必要がある ipsla 操作 id を設定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ yes と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ no と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 設定変更を実行コンフィギュレーション ファイルに保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

IPv6 ACL 用のピュア ACL ベース転送の設定

手順の概要

1. configure
2. {**ipv6** } **access-list** *name*
3. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]] [**ttl** *ttl value* [*value1 ... value2*]][**default] nexthop1** [**vrf** *vrf-name1*][**ipv6** *ipv6-address1*] [**nexthop2** [**vrf** *vrf-name2*] [**ipv6** *ipv6-address2*] [nexthop3 [vrf vrf-name3] [**ipv6***ipv6-address3*]]]
4. 次のいずれかを実行します。
   - end
   - commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure **例**： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	{**ipv6** } **access-list** *name* **例**： `RP/0/RSP0/CPU0:router(config)# ipv6 access-list security-abf-acl`	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、指定したアクセス リストを設定します。
ステップ 3	[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log**	**log-input**]] [**ttl** *ttl value* [*value1 ... value2*]][**default] nexthop1** [**vrf** *vrf-name1*][**ipv6** *ipv6-address1*] [**nexthop2** [**vrf** *vrf-name2*] [**ipv6** *ipv6-address2*] [nexthop3 [vrf vrf-name3] [**ipv6***ipv6-address3*]]] **例**： `RP/0/RSP0/CPU0:router(config-ipv6-acl)# 10 permit ipv6 any any default nexthop1 vrf vrf_A ipv6 11::1 nexthop2 vrf vrf_B ipv6 nexthop3 vrf vrf_C ipv6 33::3`	IPv6 アクセス リストの条件を設定します。 設定例では、ACL 用にピュア ACL ベース転送を設定する方法を示しています。 • このエントリに指定されたネクストホップに転送します。
ステップ 4	次のいずれかを実行します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• end
	• commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# end`
	または
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:`
		
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		
		• 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

アクセスリストおよびプレフィックスリストの実装の設定例

ここでは、次の設定例について説明します。

アクセス リストのエントリの並べ替え：例

次に、アクセス リストを並べ替える例を示します。 並べ替え後のアクセス リストの開始値は 10 で、増分値は 20 です。 後続のエントリはユーザ指定の増分値に基づいて並べられています。範囲は 1 ～ 2147483646 です。

シーケンス番号のないエントリが入力されると、デフォルトで、アクセス リストの最後のエントリのシーケンス番号に 10 を加えたシーケンス番号が割り当てられます。

```
ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
20 permit icmp any any
30 permit tcp any host 10.3.3.3
40 permit ip host 10.4.4.4 any
60 permit ip host 172.16.2.2 host 10.3.3.12
70 permit ip host 10.3.3.3 any log
80 permit tcp host 10.3.3.3 host 10.1.2.2
100 permit ip any any

configure
 ipv4 access-list acl_1
 end
resequence ipv4 access-list acl_1 10 20

ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
30 permit icmp any any
50 permit tcp any host 10.3.3.3
70 permit ip host 10.4.4.4 any
90 permit ip host 172.16.2.2 host 10.3.3.12
110 permit ip host 10.3.3.3 any log
130 permit tcp host 10.3.3.3 host 10.1.2.2
150 permit ip any any

ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
20 permit icmp any any
30 permit tcp any host 10.3.3.3
40 permit ip host 10.4.4.4 any
60 permit ip host 172.16.2.2 host 10.3.3.12
70 permit ip host 10.3.3.3 any log
80 permit tcp host 10.3.3.3 host 10.1.2.2
100 permit ip any any

configure
ipv6 access-list acl_1
end
resequence ipv6 access-list acl_1 10 20

ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
30 permit icmp any any
50 permit tcp any host 10.3.3.3
70 permit ip host 10.4.4.4 any
90 Dynamic test permit ip any any
110 permit ip host 172.16.2.2 host 10.3.3.12
130 permit ip host 10.3.3.3 any log
150 permit tcp host 10.3.3.3 host 10.1.2.2
170 permit ip host 10.3.3.3 any
190 permit ip any any
```

シーケンス番号を指定したエントリの追加：例

次の例では、新しいエントリを IPv4 アクセス リスト acl_5 に追加しています。

```
ipv4 access-list acl_5
 2 permit ipv4 host 10.4.4.2 any
 5 permit ipv4 host 10.0.0.44 any
 10 permit ipv4 host 10.0.0.1 any
 20 permit ipv4 host 10.0.0.2 any
configure
ipv4 access-list acl_5
 15 permit 10.5.5.5 0.0.0.255
end
ipv4 access-list acl_5
```

```
 2 permit ipv4 host 10.4.4.2 any
 5 permit ipv4 host 10.0.0.44 any
10 permit ipv4 host 10.0.0.1 any
15 permit ipv4 10.5.5.5 0.0.0.255 any
20 permit ipv4 host 10.0.0.2 any
```

シーケンス番号を指定しないエントリの追加：例

次に、シーケンス番号が指定されていないエントリをアクセス リストの末尾に追加する方法を示します。 シーケンス番号のないエントリを追加すると、自動的にシーケンス番号が割り当てられ、アクセス リストの末尾に配置されます。 デフォルトの増分値は 10 であるため、エントリには、既存のアクセス リストの最後のエントリのシーケンス番号に 10 を加えたシーケンス番号が割り当てられます。

```
configure
ipv4 access-list acl_10
permit 10
.1.1.1 0.0.0.255
permit 10
.2.2.2 0.0.0.255
permit 10
.3.3.3 0.0.0.255
end

ipv4 access-list acl_10
 10 permit ip 10
.1.1.0 0.0.0.255 any
 20 permit ip 10
.2.2.0 0.0.0.255 any
 30 permit ip 10
.3.3.0 0.0.0.255 any

configure
ipv4 access-list acl_10
permit 10
.4.4.4 0.0.0.255
end

ipv4 access-list acl_10
 10 permit ip 10
.1.1.0 0.0.0.255 any
 20 permit ip 10
.2.2.0 0.0.0.255 any
 30 permit ip 10
.3.3.0 0.0.0.255 any
 40 permit ip 10
.4.4.0 0.0.0.255 any
```

クラス マップの IPv6 ACL

リリース 4.2.1 では、次の項目をサポートするために、ASR 9000 Ethernet ラインカードおよび ASR 9000 Enhanced Ethernet ラインカードの Quality of Service（QoS）機能が強化されました。

- ASR 9000 Enhanced Ethernet LC：
  - L2 および L3 インターフェイスとサブインターフェイスでのサポート
  - バンドル L2 および L3 インターフェイスとサブインターフェイスでのサポート
  - 入力と出力の両方の方向のサポート

- IPv4/IPv6 用の ICMP コードとタイプ

- ASR 9000 Ethernet LC：
  - L3 インターフェイスおよびサブインターフェイスのみでのサポート
  - L3 バンドル インターフェイスおよびサブインターフェイスでのサポート
  - 入力と出力の両方の方向のサポート
  - IPv4/IPv6 用の ICMP コードとタイプ

- IPv6 でサポートされる照合フィールド：
  - 送信元 IPv6 アドレス
  - 宛先 IPv6 アドレス
  - IPv6 プロトコル
  - 存続可能時間（TTL）またはホップ リミット
  - 送信元ポート
  - 宛先ポート
  - TCP Flags
  - IPv6 フラグ（ルーティング ヘッダー（RH）、認証ヘッダー（AH）、および宛先オプション ヘッダー（DH））

- 次の項目もサポートする IPv6 ACL を使用したクラス マップ：
  - IPv4 ACL
  - 廃棄クラス
  - QoS グループ
  - 外部 CoS
  - 内部 CoS
  - 外部 VLAN（ASR 9000 Enhanced Ethernet LC のみ）
  - 内部 VLAN（ASR 9000 Enhanced Ethernet LC のみ）
  - match-not オプション
  - タイプ オブ サービス（TOS）のサポート

- 次の項目をサポートする IPv6 ACL を使用したポリシーマップ：
  - 階層型クラス マップ

IPv6 ACL QoS の設定：例

次に、IPv4 ACL およびその他のフィールドを持つ IPv6 ACL QoS を設定する例を示します。

```
ipv6 access-list aclv6
10 permit ipv6 1111:6666::2/64 1111:7777::2/64 authen
30 permit tcp host 1111:4444::2 eq 100 host 1111:5555::2 ttl eq 10
!

ipv4 access-list aclv4
10 permit ipv4 host 10.6.10.2 host 10.7.10.2
!

class-map match-any c.aclv6
match access-group ipv6 aclv6
match access-group ipv4 aclv4
match cos 1
end-class-map
!

policy-map p.aclv6
class c.aclv6
  set precedence 3
!
class class-default
!
end-policy-map
!



show qos-ea km policy p.aclv6 vmr interface tenGigE 0/1/0/6.10 hw

================================================================================
B : type & id      E : ether type     VO : vlan outer     VI : vlan inner
Q : tos/exp/group  X : Reserved       DC : discard class  Fl : flags
F2: L2 flags       F4: L4 flags       SP/DP: L4 ports
T : IP TTL         D : DFS class#      L : leaf class#
Pl: Protocol       G : QoS Grp         M : V6 hdr ext.     C : VMR count
--------------------------------------------------------------------------------
policy name p.aclv6 and km format type 4
Total Egress TCAM entries: 5
|B    F2 VO   VI   Q  G  DC T  F4 Pl SP   DP   M  IPv4/6 SA                          IPv4/6
 DA
================================================================================
V|3019 00 0000 0000 00 00 00 00 00 00 0000 0000 80 11116666:00000000:00000000:00000000
11117777:00000000:00000000:00000000
M|0000 FF FFFF FFFF FF FF FF FF FF FF FFFF FFFF 7F 00000000:00000000:FFFFFFFF:FFFFFFFF
00000000:00000000:FFFFFFFF:FFFFFFFF
R| C=0 03080200 000000A6 F06000FF 0000FF00 0002FF00 00FF0000 FF000000 00000000
V|3019 00 0000 0000 00 00 00 0A 01 00 0064 0000 00 11114444:00000000:00000000:00000002
11115555:00000000:00000000:00000002
M|0000 FF FFFF FFFF FF FF FF 00 FE FF 0000 FFFF FF 00000000:00000000:00000000:00000000
00000000:00000000:00000000:00000000
R| C=1 03080200 000000A6 F06000FF 0000FF00 0002FF00 00FF0000 FF000000 00000000
V|3018 00 0000 0000 00 00 00 00 00 00 0000 0000 00 0A060A02 -------- -------- --------
0A070A02 -------- -------- --------
M|0000 FF FFFF FFFF FF FF FF FF FF FF FFFF FFFF FF 00000000 -------- -------- --------
00000000 -------- -------- --------
R| C=2 03080200 000000A6 F06000FF 0000FF00 0002FF00 00FF0000 FF000000 00000000
V|3018 00 2000 0000 00 00 00 00 00 00 0000 0000 00 00000000:00000000:00000000:00000000
00000000:00000000:00000000:00000000
M|0003 FF 1FFF FFFF FF FF FF FF FF FF FFFF FFFF FF FFFFFFFF:FFFFFFFF:FFFFFFFF:FFFFFFFF
FFFFFFFF:FFFFFFFF:FFFFFFFF:FFFFFFFF
R| C=3 03080200 000000A6 F06000FF 0000FF00 0002FF00 00FF0000 FF000000 00000000
V|3018 00 0000 0000 00 00 00 00 00 00 0000 0000 00 00000000:00000000:00000000:00000000
00000000:00000000:00000000:00000000
M|0003 FF FFFF FFFF FF FF FF FF FF FF FFFF FFFF FF FFFFFFFF:FFFFFFFF:FFFFFFFF:FFFFFFFF
```

```
FFFFFFFF:FFFFFFFF:FFFFFFFF:FFFFFFFF
R| C=4 03000200 00010002 FF0000FF 0000FF00 0002FF00 00FF0000 FF000000 00000000
```

次に、階層型のポリシー マップを設定する例を示します。

```
ipv6 access-list aclv6.p
10 permit ipv6 1111:1111::/8 2222:2222::/8

ipv6 access-list aclv6.c
10 permit ipv6 host 1111:1111::2 host 2222:2222::3

class-map match-any c.aclv6.c
match not access-group ipv6 aclv6.c
end-class-map
!

class-map match-any c.aclv6.p
match access-group ipv6 aclv6.p
end-class-map
!

policy-map child
class c.aclv6.c
  set precedence 7
!

policy-map parent
class c.aclv6.p
  service-policy child
  set precedence 1


(config)#do show qos-ea km policy parent vmr interface tenGigE 0/1/0/6 hw

================================================================================
B : type & id      E : ether type     VO : vlan outer     VI : vlan inner
Q : tos/exp/group  X : Reserved       DC : discard class  Fl : flags
F2: L2 flags       F4: L4 flags       SP/DP: L4 ports
T : IP TTL         D : DFS class#      L : leaf class#
Pl: Protocol       G : QoS Grp         M : V6 hdr ext.     C : VMR count

================================================================================
policy name parent and format type 4
Total Ingress TCAM entries: 3
|B    F2 VO   VI   Q  G  DC T  F4 Pl SP   DP   M  IPv4/6 SA                          IPv4/6
 DA
================================================================================
V|200D 00 0000 0000 00 00 00 00 00 00 0000 0000 00 11111111:00000000:00000000:00000002
22222222:00000000:00000000:00000003
M|0000 FF FFFF FFFF FF FF FF FF FF FF FFFF FFFF FF 00000000:00000000:00000000:00000000
00000000:00000000:00000000:00000000
R| C=0 11800200 00020000 29000000 80004100 00000000 00000000 00000000 00000000
V|200D 00 0000 0000 00 00 00 00 00 00 0000 0000 00 11000000:00000000:00000000:00000000
22000000:00000000:00000000:00000000
M|0000 FF FFFF FFFF FF FF FF FF FF FF FFFF FFFF FF 00FFFFFF:FFFFFFFF:FFFFFFFF:FFFFFFFF
00FFFFFF:FFFFFFFF:FFFFFFFF:FFFFFFFF
R| C=1 11800200 00010000 29000000 80004700 00000000 00000000 00000000 00000000
V|200C 00 0000 0000 00 00 00 00 00 00 0000 0000 00 00000000:00000000:00000000:00000000
00000000:00000000:00000000:00000000
M|0003 FF FFFF FFFF FF FF FF FF FF FF FFFF FFFF FF FFFFFFFF:FFFFFFFF:FFFFFFFF:FFFFFFFF
FFFFFFFF:FFFFFFFF:FFFFFFFF:FFFFFFFF
R| C=2 11000200 00030000 00000000 00000000 00000000 00000000 00000000 00000000
```

BVI インターフェイス上の IPv4/IPv6 ACL

リリース 4.2.1 では、IPv4/IPv6 ACL は ASR 9000 Enhanced Ethernet ラインカードの BVI インターフェイス上でイネーブルになります。

BVI インターフェイス上の ACL の場合、定義された方向は次のとおりです。

- L2 インターフェイス：入力方向
- L3 インターフェイス：出力方向

A9K-SIP-700 および ASR 9000 Ethernet ラインカードでは、BVI インターフェイス上の ACL はサポートされていません。

（注） ASR 9000 Ethernet ラインカードの場合、ACL は EFP レベルで適用できます（IPv4 L3 ACL は L2 インターフェイスで適用できます）。

BVI インターフェイスでの IPv4 ACL の設定：例

次に、BVI インターフェイスで IPv4 ACL を設定する例を示します。

```
ipv4 access-list bvi-acl
10 permit ipv4 any any ttl eq 70
20 deny ipv4 any any ttl eq 60
```

IRB/BVI インターフェイスでの ABFv4 の設定：例

次に、Integrated Routing and Bridging（IRB）/ブリッジ グループ仮想インターフェイス（BVI）インターフェイスで ABFv4 を設定する例を示します。

```
interface BVI18
 ipv4 address 192.168.18.1 255.255.255.0
 ipv4 access-group abfv4 ingress
!


l2vpn
 bridge group bg18
  bridge-domain bd18
   interface GigabitEthernet0/0/1/18
   !
   routed interface BVI18
  !
 !
!

ipv4 access-list abfv4
 10 permit ipv4 any any nexthop1 ipv4 192.168.1.20 nexthop2 ipv4 192.168.9.2 nexthop3 ipv4
 192.168.10.2
```

!

その他の関連資料

ここでは、アクセス リストおよびプレフィックス リストの実装に関連する資料を示します。

関連資料

関連項目	マニュアル タイトル
アクセス リスト コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*Access List Commands*」の章
プレフィックス リスト コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*Prefix List Commands*」の章
端末サービス コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router System Management Command Reference*』の「*Terminal Services Commands*」の章

標準

標準	タイトル
この機能でサポートされる新規の標準または変更された標準はありません。また、既存の標準のサポートは変更されていません。	—

MIB

MIB	MIB のリンク
—	MIB を特定してダウンロードするには、次の URL にある Cisco MIB Locator を使用し、[Cisco Access Products] メニューからプラットフォームを選択します。http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

説明	リンク
シスコのテクニカルサポート Web サイトでは、製品、テクノロジー、ソリューション、技術的なヒント、およびツールへのリンクなどの、数千ページに及ぶ技術情報が検索可能です。Cisco.com に登録済みのユーザは、このページから詳細情報にアクセスできます。	http://www.cisco.com/en/US/support/index.html