IPv6 ACL の設定

IPv6 ACL について

アクセスコントロールリスト（ACL）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する一連のルールです（たとえば、無線クライアントからコントローラ の管理インターフェイスに ping が実行されるのを制限する場合などに使用します）。deviceで設定した ACL は、管理インターフェイス、AP マネージャインターフェイス、任意の動的インターフェイス、またはワイヤレスクライアントとやり取りするデータトラフィックの制御用の WLAN、あるいは中央処理装置（CPU）宛のすべてのトラフィックの制御用のコントローラ CPU 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。

（注） ネットワーク内で IPv4 トラフィックだけを有効にするには、IPv6 トラフィックをブロックします。つまり、すべての IPv6 トラフィックを拒否するように IPv6 ACL を設定し、これを特定またはすべての WLAN 上で適用します。

IPv6 ACL の概要

ACL のタイプ

ユーザあたりの IPv6 ACL

ユーザあたりの 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。

ACE はコントローラ組み込みワイヤレスコントローラで設定されません。ACE は `ACCESS-Accept` 属性でdeviceに送信され、クライアント用に直接適用されます。ワイヤレス クライアントが外部deviceにローミングするときに、ACE が、AAA 属性としてモビリティ ハンドオフ メッセージで外部deviceに送信されます。ユーザあたりの ACL を使用した出力方向はサポートされていません。

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deviceで設定され、`filter-id` のみが Cisco Secure ACS で設定されます。

`filter-id` は ACCESS-Accept 属性でdeviceに送信され、deviceは ACE の filter-id をルックアップしてから、クライアントに ACE を適用します。クライアント L2 が外部deviceにローミングするときに、filter-id だけがモビリティ ハンドオフ メッセージで外部deviceに送信されます。ユーザあたりの ACL を使用した出力フィルタ ACL はサポートされていません。外部deviceは filter-id と ACE を事前に設定する必要があります。

ダウンロード可能 IPv6 ACL

ダウンロード可能 ACL（dACL）の場合、完全な ACE および `dacl` 名は Cisco Secure ACS のみで設定されます。

Cisco Secure ACS はその `ACCESS-Accept` 属性で `dacl` 名をdeviceに送信します。デバイスは `dacl` 名を取得し、ACE のために `dACL` 名を `ACCESS-request` 属性を使用して Cisco Secure ACS に送り返します。

IPv6 ACL の設定の前提条件

IP Version 6（IPv6）アクセス コントロール リスト（ACL）を作成し、それをインターフェイスに適用することによって、IPv6 トラフィックをフィルタリングできます。これは、IP Version 4（IPv4）の名前付き ACL を作成し、適用する方法と同じです。また、スイッチが Network Essentials ライセンスで稼働している場合、入力ルータ ACL を作成し、それを適用してレイヤ 3 管理トラフィックをフィルタリングすることもできます。

IPv6 ACL の設定の制約事項

IPv4 では、番号制の標準 IP ACL および拡張 IP ACL、名前付き IP ACL、および MAC ACL を設定できます。IPv6 がサポートするのは名前付き ACL だけです。IPv6 ACL は Flex 接続モードをサポートしていません。

device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。

- device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。
- deviceは再帰 ACL（**reflect** キーワード）をサポートしません。
- deviceは IPv6 フレームに MAC ベース ACL を適用しません。
- ACL を設定する場合、ACL に入力されるキーワードには、それがプラットフォームでサポートされるかどうかにかかわらず、制限事項はありません。ハードウェア転送が必要なインターフェイス（物理ポートまたは SVI）に ACL を適用する場合、deviceはインターフェイスで ACL がサポートされるかどうかを判別します。サポートされない場合、ACL の付加は拒否されます。
- インターフェイスに適用される ACL に、サポートされないキーワードを持つアクセス コントロール エントリ（ACE）を追加しようとする場合、deviceは現在インターフェイスに適用されている ACL に ACE が追加されることを許可しません。

IPv6 ACL の設定

IPv6 トラフィックをフィルタリングするには、次の手順に従います。

1. IPv6 ACL を作成し、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。
2. IPv6 ACL が、トラフィックをブロックする（deny）または通過させる（permit）よう設定します。
3. トラフィックをフィルタリングする必要があるインターフェイスに IPv6 ACL を適用します。
4. インターフェイスに IPv6 ACL を適用します。ルータ ACL では、ACL が適用されるレイヤ 3 インターフェイスにも IPv6 アドレスを設定する必要があります。

IPv6 ACL のデフォルト設定

デフォルトでは、IPv6 ACL は設定または適用されていません。

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 つのスイッチまたはスイッチ スタックに作成したり、同一インターフェイスに適用できます。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

IPv6 ACL の作成

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 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	**ipv6 access-list** *acl_name* 例： `Device# ipv6 access-list access-list-name`	名前を使用して IPv6 アクセス リストを定義し、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** 例： `{deny | permit} protocol {source-ipv6-prefix/prefix-length | any | host source-ipv6-address} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any |host destination-ipv6-address} [operator [port-number]][dscp value] [fragments][log] [log-input] [routing][sequence value] [time-range name]`	条件が一致した場合にパケットを拒否する場合は deny、許可する場合は permit を指定します。次に、条件について説明します。 • protocol には、インターネット プロトコルの名前または番号を入力します。ahp、esp、icmp、ipv6、pcp、stcp、tcp、udp、または IPv6 プロトコル番号を表す 0 ～ 255 の整数を使用できます。 • source-ipv6-prefix/prefix-length または destination-ipv6-prefix/ prefix-length は、拒否条件または許可条件を設定する送信元または宛先 IPv6 ネットワークあるいはネットワーク クラスで、コロン区切りの 16 ビット値を使用した 16 進形式で指定します（RFC 2373 を参照）。 • IPv6 プレフィックス ::/0 の短縮形として、any を入力します。 • host source-ipv6-address または destination-ipv6-address には、拒否条件または許可条件を設定する送信元または宛先 IPv6 ホスト アドレスを入力します。アドレスはコロン区切りの 16 ビット値を使用した 16 進形式で指定します。 • （任意）operator には、指定のプロトコルの送信元ポートまたは宛先ポートを比較するオペランドを指定します。オペランドには、lt（より小さい）、gt（より大きい）、eq（等しい）、neq（等しくない）、range（包含範囲）があります。

	コマンドまたはアクション	目的
		source-ipv6-prefix/prefix-length 引数のあとの operator は、送信元ポートに一致する必要があります。destination-ipv6-prefix/prefix-length 引数のあとの operator は、宛先ポートに一致する必要があります。 • （任意）port-number は、0 ～ 65535 の 10 進数または TCP あるいは UDP ポートの名前です。TCP ポート名を使用できるのは、TCP のフィルタリング時だけです。UDP ポート名を使用できるのは、UDP のフィルタリング時だけです。 • （任意）dscp value を入力して、各 IPv6 パケット ヘッダーの Traffic Class フィールド内のトラフィック クラス値と DiffServ コード ポイント値を照合します。指定できる範囲は 0 ～ 63 です。 • （任意）fragments を入力して、先頭ではないフラグメントを確認します。このキーワードが表示されるのは、プロトコルが ipv6 の場合だけです。 • （任意）log を指定すると、エントリと一致するパケットに関するログ メッセージがコンソールに送信されます。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）routing を入力して、IPv6 パケットのルーティングを指定します。 • （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 ～ 4294967295 です。

	コマンドまたはアクション	目的
		• （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** 例： `{deny | permit} tcp {source-ipv6-prefix/prefix-length | any | hostsource-ipv6-address} [operator [port-number]]{destination-ipv6-prefix/prefix-length | any |hostdestination-ipv6-address} [operator [port-number]][ack] [dscp value][established] [fin] [log][log-input] [neq {port |protocol}] [psh] [range{port | protocol}] [rst][routing] [sequence value] [syn] [time-range name][urg]`	（任意）TCP アクセス リストおよびアクセス条件を定義します。 TCP の場合は tcp を入力します。パラメータはステップ 3 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。送信元からのデータはそれ以上ありません。 • neq {port | protocol}：所定のポート番号上にないパケットだけを照合します。 • psh：プッシュ機能ビット セット • range {port | protocol}：ポート番号の範囲内のパケットだけを照合します。 • rst：リセット ビット セット • syn：同期ビット セット • urg：緊急ポインタ ビット セット
ステップ 6	**{deny|permit} udp** 例： `{deny | permit} udp {source-ipv6-prefix/prefix-length | any | hostsource-ipv6-address} [operator [port-number]]{destination-ipv6-prefix/prefix-length`	（任意）UDP アクセス リストおよびアクセス条件を定義します。 ユーザ データグラム プロトコルの場合は、udp を入力します。UDP パラメータは TCP に関して説明されているパラメータと同じです。ただし、[operator

	コマンドまたはアクション	目的
		[port]] の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でなければなりません。UDP の場合、established パラメータは無効です。
ステップ 7	**{deny|permit} icmp** 例： `{deny | permit} icmp {source-ipv6-prefix/prefix-length | any | hostsource-ipv6-address} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any | hostdestination-ipv6-address} [operator [port-number]][icmp-type [icmp-code] |icmp-message] [dscpvalue] [log] [log-input] [routing] [sequence value][time-range name]`	（任意）ICMP アクセス リストおよびアクセス条件を定義します。 インターネット制御メッセージプロトコルの場合は、icmp を入力します。ICMP パラメータはステップ 3a の IP プロトコルの説明にあるパラメータとほとんど同じですが、ICMP メッセージ タイプおよびコードパラメータが追加されています。オプションのキーワードの意味は次のとおりです。 • icmp-type：ICMP メッセージ タイプでフィルタリングする場合に入力します。指定できる値の範囲は、0 ～ 255 です。 • icmp-code：ICMP パケットを ICMP メッセージコードタイプでフィルタリングする場合に入力します。指定できる値の範囲は、0 ～ 255 です。 • icmp-message：ICMP パケットを ICMP メッセージ タイプ名または ICMP メッセージ タイプとコード名でフィルタリングする場合に入力します。ICMP メッセージのタイプ名およびコード名のリストについては、? キーを使用するか、またはこのリリースのコマンドリファレンスを参照してください。
ステップ 8	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーションモードを終了できます。
ステップ 9	**show ipv6 access-list** 例：	アクセス リストの設定を確認します。

	コマンドまたはアクション	目的
	`show ipv6 access-list`	
ステップ 10	**copy running-config startup-config** 例： `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。

WLAN IPv6 ACL の作成

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `DeviceController # configure terminal`	端末を設定します。
ステップ 2	**wireless profile policy** *profile-name* 例： `Device(config)# wireless profile policy test1`	WLAN のポリシー プロファイルを作成します。 *profile-name* はポリシー プロファイルのプロファイル名です。
ステップ 3	**ipv6 acl** *acl_name* 例： `Device(config-wireless-policy)# ipv6 acl testacl`	名前付き WLAN ACL を作成します。
ステップ 4	**ipv6 traffic-filter web** *acl_name-preauth* 例： `Device(config-wlan)# ipv6 traffic-filter web preauth1`	Web 認証の事前認証 ACL を作成します。

IPv6 ACL の確認

IPv6 ACL の表示

IPv6 ACL を表示するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 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	**show access-list** 例： `Device# show access-lists`	deviceに設定されたすべてのアクセス リストを表示します。
ステップ 4	**show ipv6 access-list** *acl_name* 例： `Device# show ipv6 access-list [access-list-name]`	設定済みのすべての IPv6 アクセス リストまたは名前付けされたアクセス リストを表示します。

IPv6 ACL の設定例

例：IPv6 ACL の作成

次に、CISCO と名前が付けられた IPv6 アクセス リストを設定する例を示します。リスト内の最初の拒否エントリは、宛先 TCP ポート番号が 5000 より大きいパケットをすべて拒否します。2 番めの拒否エントリは、送信元 UDP ポート番号が 5000 未満のパケットを拒否します。また、この2番めの拒否エントリは、すべての一致をコンソールに表示します。リスト内の最初の許可エントリは、すべての ICMP パケットを許可します。リスト内の 2 番めの許可エントリは、その他のすべてのトラフィックを許可します。暗黙の全否定の条件が各 IPv6 アクセス リストの末尾にあるため、2 番めの許可エントリは必要です。

（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(config)# ipv6 access-list CISCO
Device(config-ipv6-acl)# deny tcp any any gt 5000
Device (config-ipv6-acl)# deny ::/0 lt 5000 ::/0 log
Device(config-ipv6-acl)# permit icmp any any
Device(config-ipv6-acl)# permit any any
```

例：ワイヤレス環境でのポリシー プロファイルへの IPv6 ACL の適用

次に、ワイヤレス環境でポリシー プロファイルに IPv6 ACL を適用する例を示します。

（注） すべての IPv6 ACL をポリシー プロファイルに関連付ける必要があります。

1. IPv6 ACL を作成する。

```
Device(config)# ipv6 access-list <acl-name>
Device(config-ipv6-acl)# permit tcp 2001:DB8::/32 any
Device(config-ipv6-acl)# permit udp 2001:DB8::/32 any
```

2. ポリシー プロファイルに IPv6 ACL を適用する。

```
Device(config)# wireless profile policy <policy-profile-name>
Device(config-wireless-policy)# shutdown
Device(config-wireless-policy)# ipv6 acl <acl-name>
Device(config-wireless-policy)# no shutdown
```

例：IPv6 ACL の表示

次に、**show access-lists** 特権 EXEC コマンドの出力例を示します。出力には、スイッチまたはスイッチ スタックに設定済みのすべてのアクセス リストが表示されます。

```
Device #show access-lists
Extended IP access list hello
10 permit ip any any
IPv6 access list ipv6
permit ipv6 any any sequence 10
```

次に、show ipv6 access-lists 特権 EXEC コマンドの出力例を示します。出力には、スイッチまたはスイッチ スタックに設定済みの 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：RA スロットリングの設定

このタスクでは、省電力のワイヤレス クライアントが頻繁な非請求の定期的 RA に影響されないように、RA スロットル ポリシーを作成する方法について説明します。非請求タイプのマルチキャスト RA は、コントローラ によってスロットルされます。

始める前に

クライアント マシンで IPv6 をイネーブルにします。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ipv6 nd ra-throttler policy Mythrottle** 例： `Device (config)# ipv6 nd ra-throttler policy Mythrottle`	Mythrottle という RA スロットラ ポリシーを作成します。
ステップ 3	**throttle-period 20** 例： `Device (config-nd-ra-throttle)# throttle-period 20`	スロットリングを適用する時間間隔セグメントを特定します。
ステップ 4	**max-through 5** 例： `Device (config-nd-ra-throttle)# max-through 5`	許容する初期 RA の数を特定します。
ステップ 5	**allow at-least 3 at-most 5** 例： `Device (config-nd-ra-throttle)# allow at-least 3 at-most 5`	初期 RA が送信された後に、間隔セグメントの終了まで許容される RA の数を特定します。
ステップ 6	**switch (config)# vlan configuration 100** 例： `Device (config)# vlan configuration 100`	vlan あたりの設定を作成します。
ステップ 7	**ipv6 nd ra-th attach-policy attach-policy_name** 例： `Device (config)# ipv6 nd ra-throttle attach-policy attach-policy_name`	ルータ アドバタイズメント スロットリングをイネーブルにします。
ステップ 8	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーション モードを終了できます。