



CHAPTER 36

コントロールプレーンポリシングの設定



(注) コントロールプレーンポリシングは Supervisor Engine 6-E ではサポートされません。

この章では、Control Plane Policing (CoPP; コントロールプレーンポリシング) を使用して Catalyst 4000 ファミリースイッチを保護する方法を説明します。この章の内容は Catalyst 4500 シリーズスイッチに固有であり、第 39 章「ACL によるネットワークセキュリティの設定」で説明するネットワークセキュリティ情報や手順を補足するものです。また、次のマニュアルのネットワークセキュリティ情報や手順の補足にもなります。

- 次の URL の『Cisco IOS Security Configuration Guide, Cisco IOS Release 12.4』
http://www.cisco.com/en/US/docs/ios/security/configuration/guide/12_4/sec_12_4_book.html
- 次の URL の『Cisco IOS Security Command Reference, Cisco IOS Release 12.4』
http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html



(注) この章のスイッチコマンドの構文および使用方法の詳細については、『Catalyst 4500 Series Switch Cisco IOS Command Reference』および次の URL の関連マニュアルを参照してください。

<http://www.cisco.com/en/US/products/ps6350/index.html>

この章の主な内容は、次のとおりです。

- 「制御プレーンポリシングの機能概要」(P.36-2)
- 「コントロールプレーンポリシングの注意事項」(P.36-3)
- 「CoPP のデフォルト設定」(P.36-4)
- 「CoPP の設定」(P.36-4)
- 「CoPP の設定時の注意事項および制約事項」(P.36-8)
- 「CoPP のモニタ」(P.36-8)

制御プレーン ポリシングの機能概要

CoPP 機能は、不要なトラフィックまたは DoS トラフィックから CPU を保護し、重要なコントロールプレーンおよび管理トラフィックを優先させることにより Catalyst 4000 ファミリー スイッチのセキュリティを向上させます。分類 TCAM および QoS (Quality of Service) ポリサーは、CoPP へのハードウェア サポートを提供します。CoPP は、Cisco IOS Release 12.2(31)SG がサポートするすべてのスーパーバイザ エンジンで動作します。

CPU が管理するトラフィックは、次の 3 つの機能コンポーネント (プレーン) に分割されます。

- データ プレーン
- 管理プレーン
- 制御プレーン

CoPP を使用することで、大半の CPU 行きトラフィックを保護し、ルーティングの安定性と信頼性を確保し、パケットを確実に配信することができます。特に重要なのは、CoPP を使用して CPU を DoS 攻撃 (サービス拒絶攻撃) から保護する機会が多いことです。レイヤ 2 およびレイヤ 3 コントロールプレーン パケットの選択済みセットに一致する、定義済み ACL のリストがあります。必要なポリシングパラメータをこれらのコントロールパケットに定義することはできますが、定義済み ACL の一致基準を変更することはできません。次に、定義済み ACL のリストを示します。

定義済み名前付き ACL	説明
system-cpp-dot1x	MacDA = 0180.C200.0003
system-cpp-bpdu-range	MacDA = 0180.C200.0000 - 0180.C200.000F
system-cpp-cdp	MacDA = 0100.0CCC.CCCC (UDLD/DTP/VTP/PAgP)
system-cpp-sstp	MacDA = 0100.0CCC.CCCD
system-cpp-cgmp	Mac DA = 01-00-0C-DD-DD-DD
system-cpp-ospf	IP プロトコル = OSPF、IPDA は 224.0.0.0/24 に一致
system-cpp-igmp	IP プロトコル = IGMP、IPDA は 224.0.0.0/3 に一致
system-cpp-pim	IP プロトコル = PIM、IPDA は 224.0.0.0/24 に一致
system-cpp-all-systems-on-subnet	IPDA = 224.0.0.1
system-cpp-all-routers-on-subnet	IPDA = 224.0.0.2
system-cpp-ripv2	IPDA = 224.0.0.9
system-cpp-ip-mcast-linklocal	IP DA = 224.0.0.0/24
system-cpp-dhcp-cs	IP プロトコル = UDP、L4SrcPort = 68、L4DstPort = 67
system-cpp-dhcp-sc	IP プロトコル = UDP、L4SrcPort = 67、L4DstPort = 68
system-cpp-dhcp-ss	IP プロトコル = UDP、L4SrcPort = 67、L4DstPort = 67

データプレーンおよび管理プレーントラフィックの場合、ポリシングするトラフィッククラスと一致するようにユーザの ACL を定義できます。

CoPP は MQC を使用してトラフィック分類基準を定義し、分類されたトラフィックの設定可能ポリシーアクションを指定します。MQC ではクラスマップを使用して特定のトラフィッククラスに対するパケットを定義します。トラフィックを分類したら、識別されたトラフィックにポリシーアクションを実行するための、ポリシーマップを作成できます。コントロールプレーン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と、CoPP サービスポリシーをコントロールプレーンに直接付加できます。

コントロールプレーンに付加できるポリシー マップは *system-cpp-policy* だけです。ポリシー マップの冒頭には事前に定義されたクラスマップが事前に定義された順番で含まれていることが必要です。*system-cpp-policy* ポリシー マップを作成するのに最善の方法は、グローバル マクロ *system-cpp* を使用する方法です。

system-cpp-policy には、コントロールプレーン トラフィックに対する定義済みクラス マップが含まれています。システムで定義されたすべての CoPP クラス マップの名前と、それらの一致 ACL には「*system-cpp-*」というプレフィックスが付いています。デフォルトでは、トラフィック クラスに対するアクションは指定されていません。CPU 行きデータプレーンおよび管理プレーン トラフィックに一致するクラス マップを独自に定義できます。定義したクラス マップは *system-cpp-policy* ポリシー マップに追加できます。

コントロールプレーン ポリシングの注意事項

- ポートセキュリティは、非 IP コントロール パケットに対する効果をキャンセルすることがあります。

Catalyst 4500 シリーズ スイッチでの送信元 MAC ラーニングはソフトウェアで実行されますが、コントロール パケット (IEEE BPDU/CDP/SSTP BPDU/GARP など) からの送信元 MAC アドレスのラーニングは許可されません。このような (範囲から外れた可能性のある) 高いレートのコントロール パケットを受信すると見なされるポートでポートセキュリティを設定すると、システムはパケットを転送するのではなく、(送信元アドレスが学習されるまで、ポートセキュリティがどのように実装されるか) パケットのコピーを CPU に生成します。

Catalyst 4500 スイッチング エンジンの現在のアーキテクチャでは、CPU に送信されたパケットのコピーにポリシングを適用できません。ポリシングを適用できるのは、CPU に転送されたパケットに対してだけです。したがって、パケットのコピーは、パケットが到着するレートで CPU に送信され、コントロール パケットからのラーニングが許可されないため、ポートセキュリティはトリガーされません。さらに、元のパケットではなくパケットのコピーが CPU に送信されるため、ポリシングも適用されません。

- Cisco IOS Release 12.2(31)SGA1 と同様に、GARP クラスは、CoPP の一部ではなくなりました (CSCsg08775 に伴う修正のため、*system-cpp-garp-range* エントリが CPP 設定に引き続き表示されていても、単にアイドルリングになっているだけで、その後のリリースでは削除される予定です)。これ以降、ユーザ ACL および QoS で GARP トラフィックを操作できます。GARP パケットに対して CPU を保護する場合、GARP パケットのユーザ クラスを定義したあとで、CoPP を使用して GARP パケットを下方ポリシングすることも可能です (GARP がスタティック CAM 領域の一部ではなくなったため、下方ポリシングが可能になりました)。

IOS とプラットフォーム コードの間での CPP 実装の統合が強固になったため、起動時には常にエラー メッセージが表示され、この注意事項が以前のリリースに統合されている (この修正が存在しない) バージョンから IOS ソフトウェアをダウングレードするときには、CPP が適用されません。

```
%Invalid control plane policy-map; Please unconfigure policy-map attached to control-plane, and associated class-maps, and execute config command "macro global apply system-cpp" error: failed to install policy map system-cpp-policy
```

次善策として、次の手順を実行します。

- ソフトウェアのダウングレードを実行するときは、コンフィギュレーションをバックアップします。
- コンフィギュレーションからすべての CPP エントリを手動で削除し、**macro global apply system-cpp** コマンドを再度適用します。

リリース間でアップグレードするときは、この注意事項に関連する問題は発生しないはずですが。

CoPP のデフォルト設定

CoPP はデフォルトでディセーブルです。

CoPP の設定

ここでは、次の作業について説明します。

- 「コントロールプレーン トラフィックの CoPP の設定」(P.36-4)
- 「データプレーンおよび管理プレーン トラフィックの CoPP の設定」(P.36-5)

コントロールプレーン トラフィックの CoPP の設定

コントロールプレーン トラフィックの CoPP を設定するには、次の作業を実行します。

	コマンド	目的
ステップ1	Switch# config 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	Switch(config)# qos	(任意) QoS をグローバルにイネーブルにします。
ステップ3	Switch(config)# macro global apply system-cpp	(任意) system-cpp-policy ポリシー マップを作成してコントロールプレーンに付加します。
ステップ4	Switch(config)# policy-map system-cpp-policy Switch(config-pmap)# class { system-cpp-dot1x system-cpp-bpdu-range system-cpp-cdp service system-cpp-sstp system-cpp-cgmp system-cpp-ospf system-cpp-igmp system-cpp-pim system-cpp-all-systems-on-subnet system-cpp-all-routers-on-subnet system-cpp-ripv2 system-cpp-ip-mcast-linklocal system-cpp-dhcp-cs system-cpp-dhcp-sc system-cpp-dhcp-ss } Switch(config-pmap-c)# police [aggregate name] rate burst [conform-action { drop transmit }] [{ exceed-action { drop transmit }}]	サービス ポリシー マップで 1 つまたは複数のシステム定義のコントロールプレーン トラフィックにアクションを関連付けます。必要に応じてこのステップを繰り返します。
ステップ5	Switch# show policy-map system-cpp-policy	(任意) コンフィギュレーションを確認します。

次に、CDP パケットをポリシングする例を示します。

```
Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# qos
Switch(config)# macro global apply system-cpp
Switch(config)# policy-map system-cpp-policy
Switch(config-pmap)# class system-cpp-cdp
Switch(config-pmap-c)# police 32000 1000 conform-action transmit exceed-action drop
Switch(config-pmap-c)# end
```

```

Switch# show policy-map system-cpp-policy
  Policy Map system-cpp-policy
    Class system-cpp-dot1x
    Class system-cpp-bpdu-range
  *   Class system-cpp-cdp
      police 32000 bps 1000 byte conform-action transmit exceed-action drop *
    Class system-cpp-sstp
    Class system-cpp-cgmp
    Class system-cpp-ospf
    Class system-cpp-igmp
    Class system-cpp-pim
    Class system-cpp-all-systems-on-subnet
    Class system-cpp-all-routers-on-subnet
    Class system-cpp-ripv2
    Class system-cpp-ip-mcast-linklocal
    Class system-cpp-dhcp-cs
    Class system-cpp-dhcp-sc
    Class system-cpp-dhcp-ss
Switch#

```

データプレーンおよび管理プレーン トラフィックの CoPP の設定

データプレーンおよび管理プレーン トラフィックの CoPP を設定するには、次の作業を実行します。

	コマンド	目的
ステップ1	Switch(config)# qos	(任意) QoS をグローバルにイネーブルにします。
ステップ2	Switch(config)# macro global apply system-cpp	(任意) system-cpp-policy ポリシー マップをコントロールプレーンに付加します。

	コマンド	目的
ステップ3	<pre>Switch(config)# {ip mac} access-list extended {access-list-name} For an ip access list, issue Switch(config-ext-nacl)#{permit deny} {protocol} source {source-wildcard} destination {destination-wildcard} For a mac access list, issue Switch(config-ext-macl)#{permit deny} source {source-wildcard} destination {destination-wildcard} [protocol-family] OR Switch(config)# access-list {access-list-name} {permit deny} {type-code wild-mask address mask}</pre>	<p>次のように、トラフィックと照合する ACL を定義します。</p> <ul style="list-style-type: none"> • permit : パケットが名前付き ACL をパスする条件を指定します。 • deny : パケットが名前付き ACL をパスしない条件を指定します。 <p>(注) ほとんどの場合は、重要なトラフィックまたは重要でないトラフィックを識別する ACL を設定する必要があります。</p> <ul style="list-style-type: none"> • type-code : 0x で始まる 16 進のビット数 (0x6000 など)。802 カプセル化パケットの場合は Link Service Access Point (LSAP; リンクサービスアクセスポイント) タイプコードを、SNAP カプセル化パケットの場合は SNAP タイプコードを指定します (LSAP は SAP (サービスアクセスポイント) とも呼ばれ、802 ヘッダーの DSAP (宛先サービスアクセスポイント) フィールドおよび SSAP (送信元サービスアクセスポイント) フィールドのタイプコードのことです)。 • wild-mask : 1 のビットが type-code 引数のビットに対応する 16 進数。wild-mask は、比較時に無視する type-code 引数のビットです (DSAP/SSAP のペアのマスクでは、2 つのビットが SAP コードの識別以外の目的で使用されるため、常に 0x0101 です)。 • address : 48 ビットのトークンリングアドレス。16 進の数字を 4 桁ずつドットで 3 つに区切って表します。このフィールドはベンダーコードでのフィルタリングに使用されます。 • mask : 48 ビットのトークンリングアドレス。16 進の数字を 4 桁ずつドットで 3 つに区切って表します。マスクの 1 ビットはアドレスでは無視されます。このフィールドはベンダーコードでのフィルタリングに使用されません。
ステップ4	<pre>Switch(config)# class-map {traffic-class-name} Switch(config-cmap)# match access-group {access-list-number name {access-list-name}}</pre>	<p>パケット分類基準を定義します。クラスに関連するトラフィックを識別するには、match ステートメントを使用します。</p>
ステップ5	<pre>Switch(config-cmap)# exit</pre>	<p>グローバル コンフィギュレーション モードに戻ります。</p>

	コマンド	目的
ステップ 6	<pre>Switch(config)# policy-map system-cpp-policy Switch(config-pmap)# class <class-map-name> Switch(config-pmap-c)# police [aggregate name] rate burst [conform-action {drop transmit}] [{exceed-action {drop transmit}}]</pre>	CoPP ポリシー マップにトラフィック クラスを追加します。トラフィック クラスにアクションを関連付けるには、 police 文を使用します。
ステップ 7	<pre>Switch(config)# end</pre>	特権 EXEC モードに戻ります。
ステップ 8	<pre>Switch# show policy-map system-cpp-policy</pre>	入力を確認します。

次に、信頼されるホストの送信元アドレスに 10.1.1.1 および 10.1.1.2 を設定して Telnet パケットを制約なしにコントロールプレーンに転送し、残りの Telnet パケットはすべて一定のレートでポリシングする例を示します (この例ではグローバル QoS がイネーブルであり、system-cpp-policy ポリシー マップが作成されていると仮定します)。

```
Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# qos
Switch(config)# macro global apply system-cpp

! Allow 10.1.1.1 trusted host traffic.
Switch(config)# access-list 140 deny tcp host 10.1.1.1 any eq telnet

! Allow 10.1.1.2 trusted host traffic.
Switch(config)# access-list 140 deny tcp host 10.1.1.2 any eq telnet

! Rate limit all other Telnet traffic.
Switch(config)# access-list 140 permit tcp any any eq telnet

! Define class-map "telnet-class."
Switch(config)# class-map telnet-class
Switch(config-cmap)# match access-group 140
Switch(config-cmap)# exit

! Add the class-map "telnet-class" to "system-cpp-policy" and define ! the proper action
Switch(config)# policy-map system-cpp-policy
Switch(config-pmap)# class telnet-class
Switch(config-pmap-c)# police 80000 1000 conform transmit exceed drop
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit

! Verify the above configuration steps
Switch# show policy-map system-cpp-policy
Policy Map system-cpp-policy
  Class system-cpp-dot1x
  Class system-cpp-bpdu-range
  Class system-cpp-cdp
    police 32000 bps 1000 byte conform-action transmit exceed-action drop
  Class system-cpp-sstp
  Class system-cpp-cgmp
  Class system-cpp-ospf
  Class system-cpp-igmp
  Class system-cpp-pim
  Class system-cpp-all-systems-on-subnet
  Class system-cpp-all-routers-on-subnet
  Class system-cpp-ripv2
  Class system-cpp-ip-mcast-linklocal
  Class system-cpp-dhcp-cs
```

```

Class system-cpp-dhcp-sc
Class system-cpp-dhcp-ss
* Class telnet-class
  police 8000 bps 1000 byte conform-action drop exceed-action drop

```

CoPP の設定時の注意事項および制約事項

CoPP を設定するときは、次の注意事項および制約事項に従ってください。

- 入力 CoPP だけがサポートされます。つまり、コントロールプレーンに関連する CLI では **input** キーワードだけがサポートされます。
- コントロールプレーン トラフィックをポリシングする場合はシステム定義クラス マップを使用します。
- コントロールプレーン トラフィックは、CoPP を使用する場合にだけポリシングできます。ポリシー マップをインターフェイスまたは VLAN に付加するとき、コントロールプレーン トラフィックを含むポリシー マップが受け付けられても、入力インターフェイスまたは VLAN のトラフィックはポリシングできません。
- システム定義クラス マップは、通常の QoS のポリシー マップでは使用できません。
- CPU が処理するデータプレーンおよび管理プレーン トラフィックを識別するには、ACL とクラスマップを使用します。ユーザ定義クラス マップは、CoPP の **system-cpp-policy** ポリシー マップに追加する必要があります。
- **system-cpp-policy** という名前のポリシー マップは CoPP 専用です。
- デフォルトの **system-cpp-policy** マップはシステム定義クラス マップのアクションを定義しません。つまり **no policing** です。
- **system-cpp-policy** ポリシー マップがサポートするアクションは **police** だけです。
- CoPP ポリシー ACL では **log** キーワードは使用できません。
- データプレーンおよび管理プレーン トラフィック クラスは、MAC ACL と IP ACL のどちらでも定義できます。パケットがコントロールプレーン トラフィックの事前に定義された ACL にも一致する場合は、コントロールプレーン クラスがサービス ポリシーのユーザ定義クラスの上にあるため、コントロールプレーン クラスの **police** アクション (または **no police** アクション) が実行されます。これは同じ MQC セマンティックです。
- 超過アクション **policed-dscp-transmit** は CoPP ではサポートされません。
- グローバル QoS がイネーブルで、**police** アクションが指定されないかぎり、CoPP はイネーブルになりません。

CoPP のモニタ

サイト固有ポリシーを開発して制御プレーン ポリシーの統計をモニタし、CoPP をトラブルシューティングするには、**show policy-map control-plane** コマンドを入力できます。このコマンドは、実際に適用されるポリシーのダイナミック情報を表示します。このダイナミック情報には、レート情報と、ハードウェアおよびソフトウェアに設定したポリシーに準拠または超過するバイト数 (およびパケット数) が含まれます。

show policy-map control-plane コマンドの出力は次のようになります。

```
Switch# show policy-map control-plane
```

```
Control Plane
```



```
Service-policy input: system-cpp-policy

  Class-map: system-cpp-dot1x (match-all)
    0 packets
    Match: access-group name system-cpp-dot1x

  Class-map: system-cpp-bpdu-range (match-all)
    0 packets
    Match: access-group name system-cpp-bpdu-range

*   Class-map: system-cpp-cdp (match-all)
    160 packets
    Match: access-group name system-cpp-cdp
**  police: Per-interface
    Conform: 22960 bytes Exceed: 0 bytes
*

  Class-map: system-cpp-sstp (match-all)
    0 packets
    Match: access-group name system-cpp-sstp

  Class-map: system-cpp-cgmp (match-all)
    0 packets
    Match: access-group name system-cpp-cgmp

  Class-map: system-cpp-ospf (match-all)
    0 packets
    Match: access-group name system-cpp-ospf

  Class-map: system-cpp-igmp (match-all)
    0 packets
    Match: access-group name system-cpp-igmp

  Class-map: system-cpp-pim (match-all)
    0 packets
    Match: access-group name system-cpp-pim

  Class-map: system-cpp-all-systems-on-subnet (match-all)
    0 packets
    Match: access-group name system-cpp-all-systems-on-subnet

  Class-map: system-cpp-all-routers-on-subnet (match-all)
    0 packets
    Match: access-group name system-cpp-all-routers-on-subnet

  Class-map: system-cpp-ripv2 (match-all)
    0 packets
    Match: access-group name system-cpp-ripv2

  Class-map: system-cpp-ip-mcast-linklocal (match-all)
    0 packets
    Match: access-group name system-cpp-ip-mcast-linklocal

  Class-map: system-cpp-dhcp-cs (match-all)
    83 packets
    Match: access-group name system-cpp-dhcp-cs

  Class-map: system-cpp-dhcp-sc (match-all)
    0 packets
    Match: access-group name system-cpp-dhcp-sc

  Class-map: system-cpp-dhcp-ss (match-all)
    0 packets
    Match: access-group name system-cpp-dhcp-ss
```

```
*   Class-map: telnet-class (match-all)
    0 packets
    Match: access-group 140
**  police: Per-interface
    Conform: 0 bytes Exceed: 0 bytes*
```

```
Class-map: class-default (match-any)
  0 packets
  Match: any
    0 packets
```

```
Switch#
```

コントロールプレーンのカウンタをクリアするには、**clear control-plane *** コマンドを実行します。

```
Switch# clear control-plane *
Switch#
```

すべての CoPP アクセス リスト情報を表示するには、**show access-lists** コマンドを実行します。

```
Switch# show access-lists
Extended IP access list system-cpp-all-routers-on-subnet
10 permit ip any host 224.0.0.2
Extended IP access list system-cpp-all-systems-on-subnet
10 permit ip any host 224.0.0.1
Extended IP access list system-cpp-dhcp-cs
10 permit udp any eq bootpc any eq bootps Extended IP access list
system-cpp-dhcp-sc
10 permit udp any eq bootps any eq bootpc Extended IP access list
system-cpp-dhcp-ss
10 permit udp any eq bootps any eq bootps Extended IP access list
system-cpp-igmp
10 permit igmp any 224.0.0.0 31.255.255.255 Extended IP access list
system-cpp-ip-mcast-linklocal
10 permit ip any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-ospf
10 permit ospf any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-pim
10 permit pim any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-ripv2
10 permit ip any host 224.0.0.9
Extended MAC access list system-cpp-bpdu-range
permit any 0180.c200.0000 0000.0000.000f Extended MAC access list
system-cpp-cdp
permit any host 0100.0ccc.cccc
Extended MAC access list system-cpp-cgmp
permit any host 0100.0cdd.dddd
Extended MAC access list system-cpp-dot1x
permit any host 0180.c200.0003
system-cpp-sstp
permit any host 0100.0ccc.cccd
```

CoPP アクセス リストを 1 つだけ表示するには、**show access-lists system-cpp-cdp** コマンドを実行します。

```
Switch# show access-list system-cpp-cdp
Extended MAC access list system-cpp-cdp
permit any host 0100.0ccc.cccc
Switch#
```