Cisco SM-X-16G4M2X または SM-X-40G8M2X EtherSwitch サービスモジュールの設定

初版：2019 年 6 月 25 日

最終更新：2020 年 7 月 23 日

の概要

Cisco SM-X-16G4M2X または SM-X-40G9M2X は、高密度の Small Form-Factor Pluggable（SFP）、Small Form-Factor Pluggable Plus（SFP+）、1 ギガビット 2.5 mGiGを備えたレイヤ 2 スイッチモジュールであり、Cisco 4000 シリーズ サービス統合型ルータ （ISR）に 10G 接続を提供します。また、モジュラ型ISR プラットフォームの中央転送データプレーンへの 10G 対応内部アップリンクも提供します。

SM-X-16G4M2X または SM-X-40G9M2X サービスモジュールは、すべての銅線ポートにおいて、標準規格の Power over Ethernet（PoE）、 Power over Ethernet Plus（PoE+）、 Cisco Enhanced Power over Ethernet（EPoE）、および Cisco Universal Power over Ethernet（UPoE）のサポートに対応しています。信号ペアとスペアペアの両方を活用することで、銅線ポートごとに最大 60 ワットの電力がサポートされます。

本書では、Cisco 4000シリーズサービス統合型ルータ（ISR）で SM-X-16G4M2X または SM-X-40G9M2X サービスモジュールを設定する方法について説明します。

次に、SM-X-16G4M2X または SM-X-40G8M2X サービスモジュールの機能の履歴を示します。

表 *1 : SM-X-16G4M2X* および *SM-X-40G8M2X* の機能の履歴

リリース	変更内容
Cisco IOS XE Gibraltar 16.12.1a	Cisco SM-X-16G4M2X サービスモジュールが導入されました。
Cisco IOS XE Amsterdam 17.1	Cisco SM-X-40G8M2X サービスモジュールが導入されました。

プラットフォームおよび Cisco IOS ソフトウェア イメージのサポート情報の検索

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報と注意事項については、ご使用のプラットフォームとソフトウェアリリースに対応したリリースノートを参照してください。プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

Cisco SM-X-16G4M2X または SM-X-40G9M2X サービスモジュールの設定

ここでは、Cisco SM-X-16G4M2X または SM-X-40G9M2X サービスモジュールの設定方法と、Cisco SM-X-16G4M2X または SM-X-40G9M2X サービスモジュールに関する重要な概念について説明します。

Cisco SM-X-16G4M2X サービスモジュールの前提条件

Cisco SM-X-16G4M2X を設定するには、Cisc IOS XE Bengaluru 17.5.1 リリースが必要です。

ルータで実行されている Cisco IOS ソフトウェアのバージョンを判断するには、ルータにログインし、**show version** コマンドを入力します。

```
Router> show version

Cisco IOS XE Software, Version xe.17.5.S - Standard Support Release

Cisco IOS Software, ISR Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version 15.5(2)S,
RELEASE SOFTWARE (fc3)

Technical Support: http://www.cisco.com/techsupport

Copyright (c) 1986-2015 by Cisco Systems, Inc.

Compiled Sun 22-Mar-15 02:32 by mcpre
```

- ルータ（Cisco 4000 シリーズ ISR）、Cisco IOS ソフトウェア リリースおよび機能セットを表示するには、特権 EXEC モードで **show version** コマンドを入力します。
- Cisco IOS リリース番号マッピングを表示するには、「Release Notes for the Cisco ISR 4400 Series」を参照してください。

Cisco SM-X-16G4M2X サービスモジュールの設定に関する制約事項

ここでは、Cisco SM-X-16G4M2X サービスモジュールの制約事項について説明します。

- Cisco NIM-ES2-4/NIM-ES2-8 および SM-X-16G4M2X は、機能に互換性がないため、単一のシャーシ内で共存できません。2 つのモードを切り替える場合は、システムをリロードする必要があります。

注 Cisco SM-X-16G4M2X および NIM-ES2-4/NIM-ES2-8 モジュールが同じルータに挿入されている場合、Cisco SM-X-16G4M2X サービスモジュールが優先されます。ルータがリブートし、レガシースイッチングモードではなく次世代スイッチングモードで動作します。リロード後、Cisco NIM-ES2-4/NIM-ES2-8 は非アクティブになり、Cisco SM-X-16G4M2X サービスモジュールがアクティブになります。

Power over Ethernet の設定

始める前に

SM-X-16G4M2X サービスモジュールの各銅線ポートは、次のコネクテッドデバイスのいずれか 1 つを自動検出し、電力を適切に供給します。

- IEEE 802.3af および IEEE 802.3at 準拠の受電デバイス
- Cisco EPOE および UPOE 受電デバイス

Power over Ethernet を設定するには、次のコマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** interface id 例： `Device(config)# interface gigabitethernet2/0/1`	設定する物理ポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**power inline** [**auto** | **max max-wattage**] *never* 例： `router(config-if)# power inline auto`	ポートの PoE モードを設定します。キーワードの意味は次のとおりです。 • [Auto]：受電デバイスの検出をイネーブルにします。十分な電力がある場合は、装置の検出後に PoE ポートに電力を自動的に割り当てます。これがデフォルト設定です。

	コマンドまたはアクション	目的
		• Max max-wattage：ポートで許可されている電力を制限します。PoE+ ポートの範囲は 4000 ～ 60000 mW です。Cisco UPoE ポートの範囲は 4000 ～ 60000 mW です。値を指定しない場合は、最大電力が供給されます。 • never：装置検出とポートへの電力供給をディセーブルにします。 （注） ポートにシスコの受電デバイスが接続されている場合は、power inline never コマンドでポートを設定しないでください。問題のあるリンクアップが発生し、ポートが error-disabled ステートになることがあります。
ステップ 4	**end** 例： `router(config-if)# end`	特権 EXEC モードに戻ります。

Power Over Ethernet の確認

Power Over Ethernet の設定を確認するには、次の例で示すように **show power inline** コマンドを使用します。

```
Router#show power inline
Available:500.0(w)  Used:100.3(w)  Remaining:399.8(w)

Interface Admin  Oper       Power   Device              Class Max
                            (Watts)
--------- ------ ---------- ------- ------------------- ----- ----
Gi2/0/0   auto   on         30.0    AIR-AP3802I-H-K9    4     60.0
Gi2/0/1   auto   on         10.3    IP Phone 7970       3     60.0
Gi2/0/2   auto   off        0.0     n/a                 n/a   60.0
Gi2/0/3   auto   off        0.0     n/a                 n/a   60.0
Gi2/0/4   auto   off        0.0     n/a                 n/a   60.0
Gi2/0/5   auto   off        0.0     n/a                 n/a   60.0
Gi2/0/6   auto   off        0.0     n/a                 n/a   60.0
Gi2/0/7   auto   off        0.0     n/a                 n/a   60.0
Gi2/0/8   auto   off        0.0     n/a                 n/a   60.0
Gi2/0/9   auto   off        0.0     n/a                 n/a   60.0
Gi2/0/10  auto   off        0.0     n/a                 n/a   60.0
Gi2/0/11  auto   off        0.0     n/a                 n/a   60.0
Gi2/0/12  auto   off        0.0     n/a                 n/a   60.0
Gi2/0/13  auto   off        0.0     n/a                 n/a   60.0
Gi2/0/14  auto   off        0.0     n/a                 n/a   60.0
```

```
Gi2/0/15  auto   off        0.0     n/a                 n/a   60.0
Tw2/0/16  auto   off        0.0     n/a                 n/a   60.0
Tw2/0/17  auto   on         30.0    AIR-AP3802I-H-K9    4     60.0
Tw2/0/18  auto   off        0.0     n/a                 n/a   60.0
Tw2/0/19  auto   on         30.0    AIR-AP3802I-H-K9    4     60.0
```

Universal PoE の設定

Cisco UPOE は、RJ45 ケーブルの信号ペアおよびスペアペアの両方で最大 60 W の電力を供給できます。UPOE 対応スイッチポートは、UPOE 電源デバイスとの CDP または LLDP ネゴシエーションを介し、スペアペアをイネーブルにして電力を供給できます。

エンドポイントの電源デバイスが信号ペアとスペアペアの両方で電力を消費する場合、対応する CDP/LLDP ネゴシエーションメカニズムは使用できません。次の設定を使用して、特定のポートで 4 ペアを強制的に手動設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** interface id 例： `Device(config)# interface gigabitethernet2/0/1`	設定する物理ポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**power inline four-pair forced** 例： `router(config-if)# power four-pair forced`	スイッチポートから、信号ペアおよびスペアペアの両方の電力を強制的にイネーブルにします。
ステップ 4	**end** 例： `router(config-if)# end`	特権 EXEC モードに戻ります。

ギガビット イーサネット インターフェイスの設定

速度とデュプレックス動作を設定するには、インターフェイス コンフィギュレーション モードで次の手順を実行します。

始める前に

ギガビット イーサネット インターフェイスは、10Mbps、100Mbps、または1Gbps モードとして手動で設定するか、リンクピアで適切な動作モードに自動ネゴシエートできます。

手順

	コマンドまたはアクション	目的
ステップ 1	**duplex** [**full**|**auto**] 例： `router(config-if)# duplex full`	• Auto：ピアとのデュプレックスモードを自動ネゴシエートします。 • Half：デュプレックスモードを強制的にハーフにします。ハーフモードは 10Mbps モードでのみサポートされます。 • Full：デュプレックスモードを強制的にフルにします。
ステップ 2	**speed** [**10**|**100**|**1000**|**auto**] 例： `router(config-if)# speed auto`	• 10/100/1000：速度を強制的に 10/100/1000 Mbps にします。 • Auto：ピアとの速度を自動ネゴシエートします。

2 ギガビット イーサネット インターフェイスの設定

mGig を設定するには、インターフェイス コンフィギュレーション モードで次の手順を実行します。

始める前に

mGigイーサネットインターフェイスは、100Mbps、1Gbps、または2.5Gbpsモードとして手動で設定するか、一般的に使用されている cat5e ケーブルまたはそれ以上のケーブルバリアントを介してピアリンクと自動ネゴシエートできます。

手順

	コマンドまたはアクション	目的
ステップ 1	**duplex** [**full**|**auto**] 例： `router(config-if)# duplex auto`	• Auto：ピアとのデュプレックスモードを自動ネゴシエートします。 • Full：デュプレックスモードを強制的にフルにします。
ステップ 2	**speed** [**100**|**1000**|**2500**|**auto**] 例： `router(config-if)# speed auto`	• Auto：ピアとの速度を自動ネゴシエートします。 • 100|1000|2500|：速度を 100/1000/2500 Mbps に設定します。

10 ギガビット イーサネット インターフェイスの設定

10 ギガビット イーサネット インターフェイスではデュプレックスと速度を設定できません。速度は、ポートに挿入された SFP または SPF+ のタイプによります。

フロー制御および最大伝送ユニットの設定

フロー制御により、輻輳したポートはピアノードでトラフィックを一時停止できます。出力方向で輻輳が発生したポートがある場合、他のポートにポーズフレームを使用して通知し、輻輳期間中はそのポートへのパケット転送を停止します。

（注） Cisco SM-X-16G4M2X スイッチポートは、他の Catalyst スイッチと連携する受信方向のフロー制御のみをサポートします。

すべてのインターフェイスで送受信されるフレームのデフォルト MTU サイズは、1500 バイトです。MTU サイズは、すべての外部インターフェイスでジャンボフレームをサポートするように変更できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**flowcontrol receive** [**on** | **off**] 例： `router(config-if)# flowcontrol receive on`	デフォルトの状態は off です。 • On：ピアデバイスからのポーズフレームの受信/処理をイネーブルにします。 • Off：ピアデバイスからのポーズフレームの受信/処理をディセーブルにします。
ステップ 2	**mtu** *mtu size* 例： `router(config-if)# mtu 9000`	フレームの最大伝送ユニット（MTU）サイズを設定します。範囲は 1500 ～ 9216 です。

イーサネット インターフェイス ステータスの検証

ギガビットインターフェイスのステータス情報を表示するには、**show interfaces GigabitEthernet** コマンドを使用します。

```
Router#show interfaces gigabitEthernet 2/0/14
GigabitEthernet2/0/14 is up, line protocol is up (connected)
  Hardware is SM-X-16G4M2X, address is f4db.e673.fa15 (bia f4db.e673.fa15)
  MTU 3000 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
  Full-duplex, 1000Mb/s, link type is auto, media type is 10/100/1000BaseTX
```

```
  input flow-control is on, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     258911616529 packets input, 33140686915712 bytes, 0 no buffer
     Received 0 broadcasts (0 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     0 input packets with dribble condition detected
     258846666089 packets output, 33132365295921 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out
```

mGig インターフェイスのステータス情報を表示するには、**show interfaces twoGigabitEthernet** コマンドを使用します。

```
Router# show int twoGigabitEthernet 2/0/16
TwoGigabitEthernet2/0/16 is up, line protocol is up (connected)
  Hardware is SM-X-16G4M2X, address is f4db.e673.fa17 (bia f4db.e673.fa17)
  MTU 1500 bytes, BW 2500000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
 Full-duplex, 2500Mb/s, link type is force-up, media type is 100/1000/2.5GBaseTX
input flow-control is off, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:00:05, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     172 packets input, 41736 bytes, 0 no buffer
     Received 0 broadcasts (172 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 172 multicast, 0 pause input
     0 input packets with dribble condition detected
     165 packets output, 42501 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out
```

10 ギガビットイーサネットのステータス情報を表示するには、**show interfaces tenGigabitEthernet** コマンドを使用します。

```
Router# show int tenGigabitEthernet 2/0/20
TenGigabitEthernet2/0/20 is up, line protocol is up (connected)
  Hardware is SM-X-16G4M2X, address is f4db.e673.fa1b (bia f4db.e673.fa1b)
  MTU 1500 bytes, BW 10000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
```

```
  Encapsulation ARPA, loopback not set
  Keepalive not supported
Full-duplex, 10Gb/s, link type is auto, media type is SFP-10Gbase-SR
input flow-control is off, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     2611024549517 packets input, 334211146017180 bytes, 0 no buffer
     Received 0 broadcasts (0 multicasts)
     0 runts, 28737 giants, 0 throttles
     28738 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     0 input packets with dribble condition detected
     2591035043779 packets output, 331652477689500 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out
```

MAC テーブル操作

このセクションの内容は次のとおりです。

MAC アドレス テーブルでのスタティック エントリの作成 （9 ページ）
MAC アドレスベースのトラフィック ブロッキング （10 ページ）
エージング タイマーの設定と確認 （11 ページ）

MAC アドレス テーブルでのスタティック エントリの作成

MAC アドレス テーブルにスタティック エントリを作成するには、次の作業を行います。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Router> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**mac address-table static mac-address vlan** *vlan-id* **interface** *Interface-id* 例：	MACアドレステーブルでスタティックエントリを作成します。

	コマンドまたはアクション	目的
	`Router(config)# mac address-table static 00ff.ff0d.2dc0 vlan 1 interface gigabitethernet 0/1/0`	
ステップ 4	end 例： `Router(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	show mac address-table 例： `Router# show mac address-table`	MAC アドレス テーブルを確認します。

MAC アドレスベースのトラフィック ブロッキング

特定の VLAN 内の MAC アドレスを経由するすべてのトラフィックをブロックするには、次の作業を行います。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Router> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Router#configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**mac address-table static mac-address vlan** *vlan-id* **drop** 例： `Router(config)# mac address-table static 00ff.ff0d.2dc0 vlan 1 drop`	MAC アドレステーブルで、ドロップアクションによるスタティックエントリを作成します。
ステップ 4	end 例： `Router(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	show mac address-table 例：	MAC アドレス テーブルを確認します。

	コマンドまたはアクション	目的
	`Router# show mac address-table`	

エージング タイマーの設定と確認

エージング タイマーを設定するには、次の作業を行います。

手順

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

ステップ 3	mac address-table aging-time time
	例：
	`Router(config)# mac address-table aging-time 600`
	または
	例：
	`Router(config)# mac address-table aging-time 0`
		MACアドレスのエージング タイマーを秒単位で設定します。
		• 許容値は 0 または 10 ～ 1000000 秒です。デフォルト値は300秒です。
		• スイッチ チップセットがサポートする最大エージング タイマーは634秒です。634秒以上を設定すると、MAC アドレスは 634 秒後にエージアウトします。
		• 値 0 にすると、ダイナミック MAC エントリはエージアウトしません。

ステップ 4	end 例： `Router(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	show mac address-table aging-time 例： `Router# show mac address-table aging-time`	MACアドレス テーブルを確認します。

VLAN の MAC ラーニング

指定された VLAN で MAC ラーニングをディセーブルまたはイネーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Router> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	mac address-table learning vlan *vlan-id* 例： `Router(config)#mac address-table learning vlan 10`	デフォルトでは、MAC ラーニングは各 VLAN でイネーブルです。 .
ステップ 4	end 例： `Router(config)# end`	特権 EXEC モードに戻ります。

ソフトウェア機能

Cisco SM-X-16G4M2X または SM-X-40G8M2X サービスモジュールでサポートされるソフトウェア機能は次のとおりです。

スイッチ仮想インターフェイスへの IP アドレスの割り当て

IP ルーティングを設定するには、IP アドレスをレイヤ 3 ネットワーク インターフェイスに割り当てる必要があります。これにより、IP を使用するインターフェイスでホストとの通信が可能になります。IP ルーティングはデフォルトでディセーブルであり、IP アドレスはスイッチ仮想インターフェイス（SVI）に割り当てられていません。

IP アドレスは、IP パケットの宛先を特定します。一部の IP アドレスは特殊な目的のために予約されていて、ホスト、サブネット、またはネットワーク アドレスには使用できません。RFC 1166『Internet Numbers』には、これらの IP アドレスに関する公式の説明が記載されています。

インターフェイスには、1 つのプライマリ IP アドレスを設定できます。サブネットマスクは、IP アドレスのネットワーク番号を表すビットを特定します。

IP アドレスおよびネットワークマスクを SVI に割り当てるには、特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal**	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface vlan** *vlan_id*	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、設定するレイヤ 3 VLAN を指定します。
ステップ 3	**ip address** *ip-address subnet-mask*	IP アドレスおよび IP サブネット マスクを設定します。
ステップ 4	**end**	特権 EXEC モードに戻ります。
ステップ 5	**show interfaces** [*interface-id*] **show ip interface** [*interface-id*] **show running-config interface** [*interface-id*]	入力内容を確認します。
ステップ 6	**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。

SVI でサポートされる機能

次の表に、SVI でサポートされる機能を示します。

表 *2: SVI* でサポートされる機能

テクノロジー	機能	使用例
ルーティング	Routing Protocol	SVI で設定された Routing Information Protocol（RIP）、Open Shortest Path First（OSPF）Protocol、Enhanced Interior Gateway Routing Protocol（EIGRP）などのプロトコルを使用して、レイヤ 3 ネットワークを相互接続します。 ルーティングプロトコルの詳細については、『IP ルーティング：プロトコル非依存コンフィギュレーション ガイド』を参照してください。
	ホットスタンバイ ルータ プロトコル（HSRP）	HSRP を使用して、SVI で LAN に接続されたセカンダリデバイスで冗長性と高可用性をサポートします。 HSRP の詳細については、『ファーストホップ冗長プロトコル コンフィギュレーション ガイド』を参照してください。
	DHCP	

テクノロジー	機能	使用例
		Cisco ソフトウェアを実行している Cisco デバイスには、Dynamic Host Configuration Protocol（DHCP）サーバとリレーエージェント ソフトウェアが含まれています。Cisco IOS DHCP サーバは包括的な DHCP サーバを実装し、デバイス内の指定されたアドレス プールから DHCP クライアントへの IP アドレスの割り当てと管理を行います。DHCP サーバは、ドメインネームシステム（DNS）サーバの IP アドレスやデフォルトデバイスなどの追加のパラメータを割り当てるように設定できます。 HSRP の詳細については、『IP アドレッシング：DHCP コンフィギュレーション ガイド』を参照してください。
	マルチキャスト（IPv4）	スイッチポートに接続されたクライアントにマルチキャストサポートを提供します。 HSRP の詳細については、『IP マルチキャスト：PIM コンフィギュレーション ガイド』を参照してください。
	VRF	VRF インスタンスを SVI に関連付けて、VLAN をさまざまな論理的または物理的 VPN 接続にマッピングします。 VRF プロトコルの詳細については、『IP ルーティング：プロトコル非依存コンフィギュレーション ガイド』を参照してください。

テクノロジー	機能	使用例
セキュリティ	ACL	パケットフィルタリングにより、ネットワークトラフィックを制限し、ユーザーやデバイスによるネットワークへのアクセスを制限します。 ACL プロトコルの詳細については、『セキュリティ コンフィギュレーション ガイド』の「アクセスコントロールリスト」を参照してください。
	NAT	SVI で NAT を提供します。 NAT の詳細については、『IP アドレッシング：NAT コンフィギュレーション ガイド』を参照してください。

テクノロジー	機能	使用例
QoS	標準および拡張アクセスリストによる分類	標準および拡張アクセスリストによる QoS 分類を提供します。 QoS の詳細については、『セキュリティ コンフィギュレーション ガイド』の「アクセスコントロールリスト」を参照してください。
	クラス ベース マーキング	DSCP 値とIP プレシデンス値を使用して、ユーザー定義のトラフィッククラスに基づくQoS マーキングを提供します。 QoS マーキングの詳細については、『QoS：分類コンフィギュレーション ガイド』を参照してください。
	ポリシング	SVI の入力または出力の転送速度を制限し、トラフィックが指定されたレート制限に適合または超過した場合のトラフィック処理ポリシーを指定します。 ポリシングの詳細については、『QoS: Policing and Shaping Configuration Guide』を参照してください。
ブリッジング	SVI での EVC	SVI でデフォルトのカプセル化EFPをサポートし、VLAN/BDを統合します。
	SVI での MAC ACL による EVC	EVC の詳細については、https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/cether/configuration/xe-3s/asr903/16-11-1/b-ce-layer2-xe-xe-16-11-asr900/b-ce-layer2-xe-xe-16-11-asr900_chapter_011.html を参照してください。

IEEE 802.1x プロトコル

IEEE 802.1x 規格では、一般の人がアクセス可能なポートからクライアントが LAN に接続しないように規制する（認証されている場合を除く）、クライアント/サーバ型のアクセス コントロールおよび認証プロトコルを定めています。認証サーバがポートに接続する各クライアントを認証したうえで、ルータまたは LAN が提供するサービスを利用できるようにします。

クライアントが認証されるまで、IEEE 802.1x アクセス コントロールでは、クライアントの接続先であるポートを介して、Extensible Authentication Protocol over LAN（EAPOL）、Cisco Discovery Protocol（CDP）、およびスパニングツリー プロトコル（STP）トラフィックだけが許可されます。認証後、通常のトラフィックをポート経由で送受信できます。IEEE 802.1x ポートベース認証の詳細については、『*Cisco IOS XE Gibraltar 16.10.x* セキュリティ コンフィギュレーション ガイド』[英語] の「Configuring IEEE 802.1x Port-Based Authentication」の章を参照してください。

IEEE 802.1x ポートベースの認証の設定

IEEE 802.1x ポートベースの認証は、不正なデバイス（サプリカント）によるネットワーク アクセスを防止するためにデバイスに設定されます。デバイスでは、固定構成やインストールされているモジュールに基づいて、ルータ、スイッチ、およびアクセスポイントの機能を組み合わせることができます。スイッチ機能は、組み込みスイッチ ポートまたはスイッチ ポート付きプラグインモジュールのいずれかにより提供されます。この機能は、アクセスポートとトランクポートの両方をサポートします。802.1X ポートベース認証の詳細については、『Configuring IEEE 802.1X Port-Based Authentication Guide』[英語] を参照してください。

.

VLAN 割り当てのための AAA 許可のイネーブル化

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

手順

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

	コマンドまたはアクション	目的
ステップ 3	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をイネーブルにします。
ステップ 4	**aaa authorization network radius if-authenticated** 例： `Device(config)# aaa authorization network radius if-authenticated`	ネットワーク関連のすべてのサービス要求に対して、ユーザが RADIUS 許可を受けるように device を設定します。ユーザーが承認されている場合、RADIUS 許可は成功します。
ステップ 5	**aaa authorization exec radius if-authenticated** 例： `Device(config)# aaa authorization exec radius if-authenticated`	ユーザに特権 EXEC のアクセス権限がある場合、ユーザが RADIUS 許可を受けるように device を設定します。ユーザーが承認されている場合、RADIUS 許可は成功します。
ステップ 6	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

IEEE 802.1X 認証および承認のイネーブル化

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**aaa authentication dot1x** {**default** | *listname*} *method1* [*method2...*] 例： `Device(config)# aaa authentication dot1x default group radius`	デバイスが AAA サーバと通信できるように、特権コマンド レベルにアクセスするユーザ権限の決定に使用される一連の認証方式を作成します。
ステップ 4	**dot1x system-auth-control** 例：	802.1x ポートベースの認証をグローバルにイネーブルにします。

	コマンドまたはアクション	目的
	`Device(config)# dot1x system-auth-control`	
ステップ 5	**identity profile default** 例： `Device(config)# identity profile default`	アイデンティティプロファイルを作成し、dot1x プロファイル コンフィギュレーション モードを開始します。
ステップ 6	**exit** 例： `Device(config-identity-prof)# exit`	dot1x プロファイル コンフィギュレーションモードを終了し、グローバルコンフィギュレーションモードに戻ります。
ステップ 7	**interface** *type slot/port* 例： `Device(config)# interface Gigabitethernet 1/0/1`	インターフェイス コンフィギュレーションモードを開始し、802.1X認証をイネーブルにするインターフェイスを指定します。
ステップ 8	**access-session port-control** {**auto** | **force-authorized** | **force-unauthorized**} 例： `Device(config-if)# access-session port-control auto`	インターフェイス上で 802.1x ポートベースの認証をイネーブルにします。 • **auto**：IEE 802.1x 認証をイネーブルにします。ポートは最初、無許可ステートであり、ポート経由で送受信できるのは EAPOL フレームだけです。ポートのリンク ステートがダウンからアップに変更したとき、または EAPOL-Start フレームを受信したときに、認証プロセスが開始されます。デバイスはサプリカントの識別を要求し、サプリカントと認証サーバ間で認証メッセージのリレーを開始します。デバイスはサプリカントの MACアドレスを使用して、ネットワーク アクセスを試みる各サプリカントを一意に識別します。 • **force-authorized**：802.1x 認証を無効にし、認証情報の交換を必要とせずに、ポートを許可ステートに変更します。ポートは、クライアントの IEEE 802.1x ベース認証を行わずに、通常のトラフィックを送受信します。これがデフォルト設定です。

	コマンドまたはアクション	目的
		• **force-unauthorized**：ポートが無許可ステートのままになり、サプリカントからの認証の試みをすべて無視します。デバイスは、このポートを介してサプリカントに認証サービスを提供することはできません。
ステップ 9	**dot1x pae**[**supplicant** |**authenticator** |**both**] 例： `Device(config-if)# dot1x pae authenticator`	ポートアクセスエンティティ（PAE）のタイプを設定します。 • **supplicant**：インターフェイスはサプリカントとしてだけ機能し、オーセンティケータ向けのメッセージに応答しません。 • **authenticator**：インターフェイスはオーセンティケータとしてだけ動作し、サプリカント向けのメッセージに応答しません。 • **both**：インターフェイスは、サプリカントおよびオーセンティケータとして動作するため、すべての dot1x メッセージに応答します。
ステップ 10	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。
ステップ 11	**show dot1x** 例： `Device# show dot1x`	デバイスで 802.1x 認証が設定されているかどうかを表示します。

IPv4 用 IGMP スヌーピング

IGMP スヌーピングにより、スイッチで IGMP パケットを調べたり、パケットの内容に基づいて転送先を決定したりできます。IGMP または IGMP スヌーピング クエリアからの IGMP クエリーを受信するサブネットで、IGMP スヌーピングを使用するように、スイッチを設定できます。IGMP スヌーピングは、IPv4 マルチキャスト トラフィックを受信するポートだけにそのトラフィックをダイナミックに転送するように、レイヤ 2 LAN ポートを設定することにより、レイヤ 2 で IPv4 マルチキャスト トラフィックを抑制します。

レイヤ 2 スイッチは IGMP スヌーピングを使用して、レイヤ 2 インターフェイスを動的に設定し、マルチキャスト トラフィックが IP マルチキャスト デバイスと対応付けられたインター

フェイスにのみ転送されるようにすることによって、マルチキャスト トラフィックのフラッディングを制限できます。名称が示すとおり、IGMP スヌーピングの場合は、LAN スイッチでホストとルータ間の IGMP 伝送をスヌーピングし、マルチキャスト グループとメンバ ポートを追跡する必要があります。特定のマルチキャスト グループについて、ホストから IGMP レポートを受信したスイッチは、ホストのポート番号を転送テーブル エントリに追加します。ホストから IGMP Leave Group メッセージを受信した場合は、テーブル エントリからホスト ポートを削除します。マルチキャスト クライアントから IGMP メンバーシップ レポートを受信しなかった場合にも、スイッチはエントリを定期的に削除します。この機能の詳細については、https://www.cisco.com/c/en/us/td/docs/routers/7600/ios/15S/configuration/guide/7600_15_0s_book/snooigmp.html を参照してください。

MLD スヌーピング

IP Version 4（IPv4）では、レイヤ 2 スイッチはインターネット グループ管理プロトコル（IGMP）スヌーピングを使用して、動的にレイヤ 2 インターフェイスを設定することにより、マルチキャスト トラフィックのフラッディングを抑制します。そのため、マルチキャスト トラフィックは IP マルチキャスト デバイスに対応付けられたインターフェイスにだけ転送されます。IPv6 では、MLD スヌーピングが同様の機能を実行します。MLD スヌーピングを使用すると、IPv6 マルチキャスト データは VLAN（仮想 LAN）内のすべてのポートにフラッディングされるのではなく、データを受信するポートのリストに選択的に転送されます。このリストは、IPv6 マルチキャスト制御パケットをスヌーピングすることにより構築されます。

MLD は IPv6 マルチキャスト ルータで使用されるプロトコルで、ルータに直接接続されたリンク上のマルチキャスト リスナー（IPv6 マルチキャスト パケットを受信するノード）の存在、および隣接ノードを対象とするマルチキャスト パケットを検出します。MLD は IGMP から派生しています。MLD バージョン 1（MLDv1）は IGMPv2 と、MLD バージョン 2（MLDv2）は IGMPv3 とそれぞれ同等です。MLD は Internet Control Message Protocol バージョン 6（ICMPv6）のサブプロトコルです。MLD メッセージは ICMPv6 メッセージのサブセットで、IPv6 パケット内で先頭の Next Header 値 58 により識別されます。

MLD スヌーピング設定時の注意事項

MLD スヌーピングの設定時は、次の注意事項に従ってください。

- MLD スヌーピングの特性はいつでも設定できますが、設定を有効にする場合は、 **ipv6 mld snooping** グローバル コンフィギュレーション コマンドを使用して MLD スヌーピングをグローバルにイネーブルにする必要があります。
- MLD スヌーピングと IGMP スヌーピングは相互に独立して動作します。スイッチで両方の機能を同時にイネーブルにできます。

MLD スヌーピングのデフォルト設定

表 *3: MLD* スヌーピングのデフォルト設定

機能	デフォルト設定
MLD スヌーピング（グローバル）	ディセーブル

機能	デフォルト設定
MLD スヌーピング（VLAN 単位）	イネーブルVLAN MLD スヌーピングが実行されるためには、MLD スヌーピングがグローバルにイネーブルである必要があります。
IPv6 マルチキャスト アドレス	未設定
IPv6 マルチキャスト ルータ ポート	未設定
MLD スヌーピング即時脱退	ディセーブル
MLD スヌーピングの堅牢性変数	グローバル：2、VLAN 単位：0 （注） VLAN 値はグローバル設定を上書きします。VLAN 値が 0 の場合、VLAN はグローバル数を使用します。
最後のリスナー クエリー カウント	グローバル：2、VLAN 単位：0 （注） VLAN 値はグローバル設定を上書きします。VLAN 値が 0 の場合、VLAN はグローバル数を使用します。
最後のリスナークエリーインターバル	グローバル：1000（1 秒）、VLAN：0 （注） VLAN 値はグローバル設定を上書きします。VLAN 値が 0 の場合、VLAN はグローバルのインターバルを使用します。
TCN クエリー送信請求	ディセーブル
TCN クエリー カウント	2
MLD リスナー抑制	

VLAN に対する MLD スヌーピングのイネーブル化またはディセーブル化

VLAN で MLD スヌーピングをイネーブルにするには、次の手順を実行します。

手順

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

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 mld snooping** 例： `Device(config)# ipv6 mld snooping`	スイッチで MLD スヌーピングをイネーブルにします。
ステップ 4	**ipv6 mld snooping vlan** *vlan-id* 例： `Device(config)# ipv6 mld snooping vlan 1`	VLAN で MLD スヌーピングをイネーブルにします。指定できる VLAN ID の範囲は 1 ～ 1001 および 1006 ～ 4094 です。 （注） VLAN スヌーピングをイネーブルにするには、MLD スヌーピングがグローバルにイネーブルである必要があります。
ステップ 5	**end** 例： `Device(config)# ipv6 mld snooping vlan 1`	特権 EXEC モードに戻ります。

単方向リンク検出の設定

UniDirectional Link Detection（UDLD）は、光ファイバまたはツイストペア イーサネット ケーブルを通して接続されたデバイスからケーブルの物理設定をモニタリングしたり、単一方向リンクの存在を検出できるようにするためのレイヤ2プロトコルです。このプロトコルが単一方向リンクを正常に識別してディセーブルにするには、接続されたすべてのデバイスで UDLD プロトコルがサポートされている必要があります。UDLD は単一方向リンクを検出すると、影響を受けるポートをディセーブルにして警報を発信します。単一方向リンクは、スパニングツリー トポロジ ループをはじめ、さまざまな問題を引き起こす可能性があります。

UDLD のグローバルなイネーブル化

アグレッシブモードまたは通常モードで UDLD をイネーブルにし、デバイス上のすべての光ファイバポートに設定可能なメッセージタイマーを設定するには、次の手順に従います。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**udld** {**aggressive** | **enable** | **message time** *message-timer-interval*} 例： `Device(config)# udld enable message time 10`	UDLD モードの動作を指定します。 • **aggressive**：すべての光ファイバ ポートにおいて、アグレッシブ モードでUDLDをイネーブルにします。 • **enable**：上のすべての光ファイバ ポート上で、UDLDを通常モードでイネーブルにします。UDLD はデフォルトでディセーブルです。 個々のインターフェイスの設定は、**udld enable** グローバル コンフィギュレーション コマンドの設定を上書きします。 • **message time** *message-timer-interval*：アドバタイズメント フェーズにあり、双方向リンクが検出されたポートでの UDLD プローブ メッセージの時間間隔を設定します。有効な範囲は 1 ～ 90 秒です。デフォルト値は 15 です。 （注） このコマンドが作用するのは、光ファイバ ポートだけです。他のポートタイプで UDLD をイネーブルにする場合は、**udld** インターフェイス コンフィギュレーション コマンドを使用します。 UDLDをディセーブルにするには、このコマンドの **no** 形式を使用します。
ステップ 3	**end** 例：	特権 EXEC モードに戻ります。

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

インターフェイスでの UDLD のイネーブル化

アグレッシブ モードまたは通常モードをイネーブルにする、またはポート上で UDLD をディセーブルにするには、次の手順に従います。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet`	UDLD用にイネーブルにするポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**udld port** [**aggressive**] 例： `Device(config-if)# udld port aggressive`	UDLD はデフォルトでディセーブルです。 • **udld port**：指定されたポート上で、UDLDを通常モードでイネーブルにします。 • **udld port aggressive**：（任意）指定されたインターフェイスにおいて、アグレッシブ モードで UDLD をイネーブルにします。 （注） 特定の光ファイバポート上で UDLD をディセーブルにする場合は、 **no udld port** インターフェイス コンフィギュレーション コマンドを使用します。
ステップ 4	**end** 例：	特権 EXEC モードに戻ります。

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

スイッチド ポート アナライザ（SPAN）の設定

ここでは、SM-X-16G4M2X または SM-X-40G8M2X サービスモジュールで、スイッチドポートアナライザ（SPAN）セッションを設定する方法について説明します。SM-X-16G4M2X または SM-X-40G8M2X サービスモジュールには、次の制約事項が適用されます。

- モジュール内のローカル SPAN のみサポートされます。モジュール間の SPAN はサポートされません。
- 各 SM-X-16G4M2X または SM-X-40G8M2X サービスモジュールでは、すべてのポートで 66 の SPAN セッションをサポート可能です。ただし、ローカル SPAN セッションとリモート SPAN 送信元セッションを含む送信元セッションとして使用できるのは、そのうちの 8 つだけです。残りのセッションは、リモート SPAN 接続先セッションとして使用できます。
- セッション ID の範囲は 1 ～ 66 です。

（注） 送信、受信、または送受信の監視がサポートされています。

SPAN および RSPAN

ポートまたは VLAN を通過するネットワークトラフィックを解析するには、SPAN または RSPAN を使用して、そのデバイス上、またはネットワークアナライザやその他のモニタデバイス、あるいはセキュリティデバイスに接続されている別のデバイス上のポートにトラフィックのコピーを送信します。SPAN は送信元ポート上または送信元 VLAN 上で受信、送信、または送受信されたトラフィックを宛先ポートにコピー（ミラーリング）して、解析します。SPAN は送信元ポートまたは VLAN 上のネットワーク トラフィックのスイッチングには影響しません。宛先ポートは SPAN 専用にする必要があります。デフォルトでは、宛先ポートはトラフィックを受信も転送もしません。宛先ポートで入力転送が有効になっている場合は、トラフィックを受信または転送できます。

SPAN を使用してモニタできるのは、送信元ポートを出入りするトラフィックまたは送信元 VLAN に出入りするトラフィックだけです。送信元 VLAN にルーティングされたトラフィックはモニタできません。たとえば、着信トラフィックをモニタしている場合、別の VLAN から送信元 VLAN にルーティングされているトラフィックはモニタできません。ただし、送信元 VLAN で受信し、別の VLAN にルーティングされるトラフィックは、モニタできます。

ネットワーク セキュリティ デバイスからトラフィックを注入する場合、SPAN または RSPAN 宛先ポートを使用できます。たとえば、Cisco 侵入検知システム（IDS）センサー装置を宛先ポートに接続した場合、IDS デバイスは TCP リセット パケットを送信して、疑わしい攻撃者の TCP セッションを停止させることができます。

ローカル SPAN セッションの作成

SPAN セッションを作成し、送信元（監視対象）ポートまたは VLAN、および宛先（監視側）ポートを指定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例： `Device(config)# no monitor session all`	セッションに対する既存の SPAN 設定を削除します。 • *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。 • **local**：すべてのローカル セッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 4	**monitor session** *session_number* **source** { **interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] 例： `Device(config)# monitor session 1 source interface gigabitethernet1/0/1`	SPAN セッションおよび送信元ポート/VLAN（モニタ対象ポート）を指定します。 • *session_number* の範囲は、1 ～ 66 です。 • *interface-id* には、モニタリングする送信元ポートを指定します。有効なインターフェイスには、物理インターフェイスおよびポートチャネル論理インターフェイス（**port-channel** *port-channel-number*）があります。有効なポートチャネル番号は 1 ～ 32 です。

	コマンドまたはアクション	目的
		• *vlan-id* には、監視する送信元 VLAN を指定します。指定できる範囲は 1 ～ 4094 です（RSPAN VLAN は除く）。 （注） 1 つのセッションに、一連のコマンドで定義された複数の送信元（ポートまたは VLAN）を含めることができます。ただし、1 つのセッション内では送信元ポートと送信元 VLAN を併用できません。 • （任意）**[,|-]** には、一連のインターフェイスまたはインターフェイスの範囲を指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。 • （任意）**both | rx | tx**：監視するトラフィックの方向を指定します。トラフィックの方向を指定しなかった場合、送信元インターフェイスは送信トラフィックと受信トラフィックの両方を送信します。 • **both**：受信トラフィックと送信トラフィックの両方を監視します。 • **rx**：受信トラフィックをモニタします。 • **tx**：送信トラフィックをモニタします。 （注） **monitor session** *session_number***source** コマンドを複数回使用すると、複数の送信元ポートを設定できます。

	コマンドまたはアクション	目的
ステップ 5	**monitor session** *session_number* **destination** { **interface** *interface-id* [**,** | **-**] [**encapsulation {replicate | dot1q}**]} 例： `Device(config)# monitor session 1 destination interface gigabitethernet1/0/2 encapsulation replicate`	（注） ローカル SPAN の場合は、送信元および宛先インターフェイスに同じセッション番号を使用する必要があります。 • *session_number* には、ステップ 4 で入力したセッション番号を指定します。 • （任意）**[,|-]** には、一連のインターフェイスまたはインターフェイスの範囲を指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。 （任意）**encapsulation replicate** には、宛先インターフェイスが送信元インターフェイスのカプセル化方式を複製することを指定します。選択しない場合のデフォルトは、ネイティブ形式（タグなし）でのパケットの送信です。 （任意）**encapsulation dot1q** は宛先インターフェイスが IEEE 802.1Q カプセル化の送信元インターフェイスの着信パケットを受け入れるように指定します。 （注） **monitor session** *session_number* **destination** コマンドを複数回使用すると、複数の送信元ポートを設定できます。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例：	（任意）コンフィギュレーション ファイルに設定を保存します。

	コマンドまたはアクション	目的
	`Device# copy running-config startup-config`	

宛先で許可される着信トラフィックを伴うローカル SPAN の作成

SPAN セッションを作成し、さらに送信元ポートまたは VLAN および宛先ポートを指定した後、宛先ポートでネットワーク セキュリティ デバイス（Cisco IDS センサー装置等）用に着信トラフィックをイネーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例： `Device(config)# no monitor session all`	セッションに対する既存の SPAN 設定を削除します。 • *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。 • **local**：すべてのローカル セッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 4	**monitor session** *session_number* **source** { **interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] 例： `Device(config)# monitor session 2 source gigabitethernet1/0/1 rx`	SPAN セッションおよび送信元ポート（モニタ対象ポート）を指定します。

	コマンドまたはアクション	目的
ステップ 5	**monitor session** *session_number* **destination** { **interface** *interface-id* [**,** | **-**] [**encapsulation replicate**] [**ingress** { **dot1q vlan** *vlan-id* | **untagged vlan** *vlan-id* | **vlan** *vlan-id*}]} 例： `Device(config)# monitor session 2 destination interface gigabitethernet1/0/2 encapsulation replicate ingress dot1q vlan 6`	SPAN セッション、宛先ポート、パケットカプセル化、および入力 VLAN とカプセル化を指定します。 • *session_number* には、ステップ 4 で入力したセッション番号を指定します。 • *interface-id* には、宛先ポートを指定します。宛先インターフェイスには物理ポートまたはポートチャネルを指定する必要があります。EtherChannel や VLAN は指定できません。 •（任意）[**,**|**-**]：一連のインターフェイスまたはインターフェイスの範囲を指定します。カンマまたはハイフンの前後にスペースを 1 つずつ入力します。 •（任意）**encapsulation replicate** には、宛先インターフェイスが送信元インターフェイスのカプセル化方式を複製することを指定します。選択しない場合のデフォルトは、ネイティブ形式（タグなし）でのパケットの送信です。 •（任意）**encapsulation dot1q** は宛先インターフェイスが IEEE 802.1Q カプセル化の送信元インターフェイスの着信パケットを受け入れるように指定します。 • **ingress** 宛先ポートでの着信トラフィックの転送をイネーブルにして、カプセル化タイプを指定します。 • **dot1q vlan** *vlan-id*：デフォルトの VLAN として指定した VLAN で、IEEE 802.1Q でカプセル化された着信パケットを受け入れます。

	コマンドまたはアクション	目的
		• **untagged vlan** *vlan-id* または **vlan** *vlan vlan-id*：デフォルトの VLAN として指定した VLAN で、タグなしでカプセル化された着信パケットを受け入れます。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

フィルタリングする VLAN の指定

SPAN 送信元トラフィックを特定の VLAN に制限するには、次の手順を実行します。

手順

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

	コマンドまたはアクション	目的
ステップ 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例： `Device(config)# no monitor session all`	セッションに対する既存の SPAN 設定を削除します。 • *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。 • **local**：すべてのローカル セッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 4	**monitor session** *session_number* **source interface** *interface-id* 例： `Device(config)# monitor session 2 source interface gigabitethernet1/0/2 rx`	送信元ポート（モニタ対象ポート）と SPAN セッションの特性を指定します。 • *session_number* の範囲は、1 ～ 66 です。 • *interface-id* には、モニタリングする送信元ポートを指定します。指定したインターフェイスは、あらかじめトランク ポートとして設定しておく必要があります。
ステップ 5	**monitor session** *session_number* **filter vlan** *vlan-id* [**,** | **-**] 例： `Device(config)# monitor session 2 filter vlan 1 - 5 , 9`	SPAN 送信元トラフィックを特定の VLAN に制限します。 • *session_number* には、ステップ 4 で指定したセッション番号を入力します。 • *vlan-id* に指定できる範囲は 1 ～ 4094 です。 • （任意）カンマ（**,**）を使用して一連の VLAN を指定するか、ハイフン（**-**）を使用して VLAN 範囲を指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。
ステップ 6	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate** |**encapsulation dot1q**]}	SPAN セッションおよび宛先ポート（監視側ポート）を指定します。

	コマンドまたはアクション	目的
	例： `Device(config)# monitor session 2 destination interface gigabitethernet1/0/1`	• *session_number* には、ステップ 4 で入力したセッション番号を指定します。 • *interface-id* には、宛先ポートを指定します。宛先インターフェイスには物理ポートまたはポートチャネルを指定する必要があります。EtherChannel や VLAN は指定できません。 •（任意）**[,|-]** には、一連のインターフェイスまたはインターフェイスの範囲を指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。 •（任意）**encapsulation replicate** には、宛先インターフェイスが送信元インターフェイスのカプセル化方式を複製することを指定します。選択しない場合のデフォルトは、ネイティブ形式（タグなし）でのパケットの送信です。 •（任意） **encapsulation dot1q** IEEE 802.1Q は、複数のスイッチとルータを相互接続し、VLAN トポロジを定義するための標準プロトコルです。サブインターフェイスに VLAN ID を割り当てます。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 9	**copy running-config startup-config** 例：	（任意）コンフィギュレーション ファイルに設定を保存します。

	コマンドまたはアクション	目的
	`Device# copy running-config startup-config`	

SPAN セッションの確認

SPAN セッションで設定された送信元と宛先を確認するには、**show monitor session** コマンドを使用します。

```
Router#show monitor session 1

Session 1
---------
Session 1
---------
Type : Local Session
Source Ports :
Both : Gi0/1/0
Destination Ports : Gi0/1/1
```

SPAN セッションの削除

SPAN セッションから送信元または宛先を削除するには、次の例に示すように、グローバル コンフィギュレーション モードで **no monitor session** *session* コマンドを使用します。

Router(config)#**no monitor session** *1*

RSPAN VLAN としての VLAN の設定

新しい VLAN を作成し、RSPAN セッション用の RSPAN VLAN になるように設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**vlan** *vlan-id* 例：	VLAN ID を入力して VLAN を作成するか、または既存の VLAN の VLAN ID を入力して、VLAN コンフィギュレーショ

	コマンドまたはアクション	目的
	`Device(config)# vlan 100`	ン モードを開始します。指定できる範囲は 2 ～ 1001 または 1006 ～ 4094 です。 RSPAN VLAN を VLAN 1（デフォルト VLAN）または VLAN ID 1002 ～ 1005（トークンリングおよびFDDI VLAN 専用）にすることはできません。
ステップ 4	**remote-span** 例： `Device(config-vlan)# remote-span`	VLAN を RSPAN VLAN として設定します。
ステップ 5	**end** 例： `Device(config-vlan)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

RSPAN に参加するすべてのデバイスに RSPAN VLAN を作成する必要があります。RSPAN VLAN ID が標準範囲（1005 未満）であり、VTP がネットワーク内でイネーブルである場合は、1 つのデバイスに RSPAN VLAN を作成し、VTP がこの RSPAN VLAN を VTP ドメイン内の他のデバイスに伝播するように設定できます。拡張範囲 VLAN（1005 を超える ID）の場合、送信元と宛先の両方のデバイス、および中間デバイスに RSPAN VLAN を設定する必要があります。

VTP プルーニングを使用して、RSPAN トラフィックが効率的に流れるようにするか、または RSPAN トラフィックの伝送が不要なすべてのトランクから、RSPAN VLAN を手動で削除します。

VLAN からリモート SPAN 特性を削除して、標準 VLAN に戻すように変換するには、**no remote-span** VLAN コンフィギュレーション コマンドを使用します。

SPAN セッションから送信元ポートまたは VLAN を削除するには、**no monitor session***session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} グローバル コンフィギュレーション コマンドを使用します。セッションから RSPAN VLAN を削除するには、**no monitor session** *session_number* {**Source|destination** }**remote vlan***vlan-id* コマンドを使用します。

RSPAN 送信元セッションの作成

RSPAN 送信元セッションを作成および開始し、モニタ対象の送信元および宛先 RSPAN VLAN を指定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例： `Device(config)# no monitor session 1`	セッションに対する既存の SPAN 設定を削除します。 • *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。 • **local**：すべてのローカル セッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 4	**monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] 例： `Device(config)# monitor session 1 source interface gigabitethernet1/0/1 tx`	RSPAN セッションおよび送信元ポート（モニタ対象ポート）を指定します。 • *session_number* の範囲は、1 ～ 66 です。 • RSPAN セッションの送信元ポートまたは送信元VLANを入力します。 • *interface-id* には、モニタリングする送信元ポートを指定しま

	コマンドまたはアクション	目的
		す。有効なインターフェイスには、物理インターフェイスおよびポートチャネル論理インターフェイス（**port-channel** *port-channel-number*）があります。有効なポートチャネル番号は 1 ～ 32 です。 • *vlan-id* には、モニタする送信元 VLAN を指定します。指定できる範囲は 1 ～ 4094 です（RSPAN VLAN は除く）。 1 つのセッションに、一連のコマンドで定義された複数の送信元（ポートまたは VLAN）を含めることができます。ただし、1 つのセッション内で送信元ポートと送信元 VLAN を併用することはできません。 • （任意）**[,|-]**：一連のインターフェイスまたはインターフェイスの範囲を指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。 • （任意）**both | rx | tx**：監視するトラフィックの方向を指定します。トラフィックの方向を指定しなかった場合、送信元インターフェイスは送信トラフィックと受信トラフィックの両方を送信します。 • **both**：受信トラフィックと送信トラフィックの両方を監視します。 • **rx**：受信トラフィックをモニタします。 • **tx**：送信トラフィックをモニタします。

	コマンドまたはアクション	目的
ステップ 5	**monitor session** *session_number* **destination remote vlan** *vlan-id* 例： `Device(config)# monitor session 1 destination remote vlan 100`	RSPAN セッション、宛先 RSPAN VLAN、および宛先ポート グループを指定します。 • *session_number* には、ステップ 4 で指定した番号を入力します。 • *vlan-id* には、宛先セッションへミラートラフィックを転送する RSPAN VLAN を、送信元セッションで指定します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

RSPAN 送信元セッションでフィルタリングする VLAN の指定

RSPAN 送信元トラフィックを特定の VLAN に制限するように RSPAN 送信元セッションを設定するには、次の手順を実行します。

手順

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

	コマンドまたはアクション	目的
	`Device# configure terminal`	
ステップ 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例： `Device(config)# no monitor session 2`	セッションに対する既存の SPAN 設定を削除します。 • *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。 • **local**：すべてのローカル セッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 4	**monitor session** *session_number* **source interface** *interface-id* 例： `Device(config)# monitor session 2 source interface gigabitethernet1/0/2 rx`	送信元ポート（モニタ対象ポート）と SPANセッションの特性を指定します。 • *session_number* の範囲は、1 ～ 66 です。 • *interface-id* には、モニタリングする送信元ポートを指定します。指定したインターフェイスは、あらかじめトランク ポートとして設定しておく必要があります。
ステップ 5	**monitor session** *session_number* **filter vlan** *vlan-id* [**,** | **-**] 例： `Device(config)# monitor session 2 filter vlan 1 - 5 , 9`	SPAN 送信元トラフィックを特定の VLAN に制限します。 • *session_number* には、ステップ 4 で指定したセッション番号を入力します。 • *vlan-id* に指定できる範囲は 1 ～ 4094 です。 • （任意）**,**| **-** カンマ（**,**）を使用して一連の VLAN を指定するか、ハイフン（**-**）を使用して VLAN 範囲を指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。

	コマンドまたはアクション	目的
ステップ 6	**monitor session** *session_number* **destination remote vlan** *vlan-id* 例： `Device(config)# monitor session 2 destination remote vlan 902`	RSPAN セッションおよび宛先リモート VLAN（RSPAN VLAN）を指定します。 • *session_number* には、ステップ 4 で指定したセッション番号を入力します。 • *vlan-id* には、宛先ポートにモニタ対象トラフィックを伝送する RSPAN VLAN を指定します。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

RSPAN 宛先セッションの作成および着信トラフィックの設定

RSPAN 宛先セッションを作成し、送信元 RSPAN VLAN および宛先ポートを指定し、宛先ポートでネットワーク セキュリティ デバイス（Cisco IDS センサー装置等）用に着信トラフィックをイネーブルにするには、次の手順を実行します。

手順

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

	コマンドまたはアクション	目的
	`Device# configure terminal`	
ステップ 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例： `Device(config)# no monitor session 2`	セッションに対する既存の SPAN 設定を削除します。 • *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。 • **local**：すべてのローカル セッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 4	**monitor session** *session_number* **source remote vlan** *vlan-id* 例： `Device(config)# monitor session 2 source remote vlan 901`	RSPAN セッションと送信元 RSPAN VLAN を指定します。 • *session_number* の範囲は、1 ～ 66 です。 • *vlan-id* には、宛先セッションで RSPAN VLAN を指定します。これは送信元セッションからミラートラフィックを受信します。
ステップ 5	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**ingress** { **dot1q vlan** *vlan-id* | **untagged vlan** *vlan-id* **| vlan** *vlan-id*}]} 例： `Device(config)# monitor session 2 destination interface gigabitethernet1/0/2 ingress vlan 6`	SPAN セッション、宛先ポート、パケット カプセル化、および着信 VLAN とカプセル化を指定します。 • *session_number* には、ステップ 5 で指定した番号を入力します。 RSPAN 宛先セッションでは、送信元 RSPAN VLAN および宛先ポートに同じセッション番号を使用する必要があります。 • *interface-id* には、宛先インターフェイスを指定します。宛先インターフェイスは物理インターフェイスでなければなりません。 • **encapsulation replicate** はコマンドラインのヘルプストリングに表示さ

	コマンドまたはアクション	目的
		れますが、RSPAN ではサポートされていません。元の VLAN ID は RSPAN VLAN ID によって上書きされ、宛先ポート上のすべてのパケットはタグなしになります。 • （任意）[,|-] には、一連のインターフェイスまたはインターフェイスの範囲を指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。 • 宛先ポートでの着信トラフィックの転送をイネーブルにして、カプセル化タイプを指定するには、**ingress** を追加のキーワードと一緒に入力します。 • **dot1q vlan** *vlan-id*：デフォルトの VLAN として指定した VLAN で、IEEE 802.1Q でカプセル化された着信パケットを転送します。 • **untagged vlan** *vlan-id* または **vlan** *vlan-id*：デフォルトの VLAN として指定した VLAN で、タグなしでカプセル化された着信パケットを転送します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

	コマンドまたはアクション	目的
	startup-config	

VLANs

VLANは、ユーザの物理的な位置に関係なく、機能またはアプリケーションなどで論理的に分割された、スイッチによるネットワークです。VLANには、物理LANと同じ属性があります。ただし、同じ LAN セグメントに物理的に配置されていないエンドステーションもグループ化できます。どのようなデバイスポートでもVLANに属することができ、ユニキャスト、ブロードキャスト、マルチキャストのパケットは、その VLAN 内のエンドステーションだけに転送またはフラッディングされます。各 VLAN は 1 つの論理ネットワークと見なされ、VLAN に属さないステーション宛のパケットは、ルータまたはフォールバックブリッジングをサポートするデバイスを経由して伝送しなければなりません。デバイススタックでは、スタック全体にまたがる複数のポートでVLANを形成できます。VLANはそれぞれが独立した論理ネットワークと見なされるので、VLANごとに独自のブリッジ管理情報ベース（MIB）情報があり、スパニングツリーの独自の実装をサポートできます。

VLAN は通常、IP サブネットワークに対応付けられます。たとえば、特定の IP サブネットに含まれるエンド ステーションはすべて同じ VLAN に属します。デバイス上のインターフェイスの VLAN メンバーシップは、インターフェイスごとに手動で割り当てます。この方法でデバイスインターフェイスを VLAN に割り当てた場合、これをインターフェイスベース（またはスタティック）VLAN メンバーシップと呼びます。

デバイスは、デバイス仮想インターフェイス（SVI）を使用して、VLAN 間でトラフィックをルーティングできます。VLAN 間でトラフィックをルーティングするには、SVI を明示的に設定して IP アドレスを割り当てる必要があります。

アクセス ポート

アクセス ポートは（音声 VLAN ポートとして設定されている場合を除き）1 つの VLAN だけに所属し、その VLAN のトラフィックだけを伝送します。トラフィックは、VLAN タグが付いていないネイティブ形式で送受信されます。アクセス ポートに着信したトラフィックは、ポートに割り当てられている VLAN に所属すると見なされます。アクセスポートがタグ付きパケット（タグ付き IEEE 802.1Q）を受信した場合、そのパケットは廃棄され、送信元アドレスは学習されません。

トランク ポート

トランク ポートは複数の VLAN のトラフィックを伝送し、デフォルトで VLAN データベース内のすべての VLAN のメンバとなります。次のトランク ポート タイプはサポートされています。

- IEEE 802.1Q トランク ポートは、タグ付きとタグなしの両方のトラフィックを同時にサポートします。IEEE 802.1Q トランク ポートは、デフォルトのポート VLAN ID（PVID）に割り当てられ、すべてのタグなしトラフィックはポートのデフォルト PVID 上を流れます。NULL VLAN ID を備えたすべてのタグなしおよびタグ付きトラフィックは、ポートのデフォルト PVID に所属するものと見なされます。発信ポートのデフォルト PVID と等し

いVLAN IDを持つパケットは、タグなしで送信されます。残りのトラフィックはすべて、VLAN タグ付きで送信されます。

デフォルトでは、トランク ポートは、VTPに認識されているすべてのVLANのメンバですが、トランク ポートごとに VLAN の許可リストを設定して、VLAN メンバーシップを制限できます。許可 VLAN のリストは、その他のポートには影響を与えませんが、対応トランク ポートには影響を与えます。デフォルトでは、使用可能なすべての VLAN（VLAN ID 1 ～ 4094）が許可リストに含まれます。トランクポートは、VTP が VLAN を認識し、VLAN が有効な状態にある場合に限り、VLANのメンバーになることができます。VTPが新しい有効になっているVLAN を認識し、その VLAN がトランクポートの許可リストに登録されている場合、トランクポートは自動的にその VLAN のメンバになり、トラフィックはその VLAN のトランク ポート間で転送されます。VTPが、VLANのトランクポートの許可リストに登録されていない、新しい有効な VLAN を認識した場合、ポートはその VLAN のメンバーにはならず、その VLAN のトラフィックはそのポート間で転送されません。

VLAN の詳細については、https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9200/software/release/16-10/configuration_guide/vlan/b_1610_vlan_9200_cg/configuring_vlans.html を参照してください。

VLAN の作成

始める前に

VTP バージョン 1 および 2 でデバイスが VTP トランスペアレントモードの場合は、1006 を超える VLAN ID を割り当てることができますが、それらを VLAN データベースに追加できません。

VLAN を設定するには、次の手順を実行します。VLAN は、アクセスモードまたはトランクモードで設定できます。手順は両方のモードで同じです。

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	**vlan** *vlan-id* 例： `(config)# vlan 20`	VLAN ID を入力して、VLAN コンフィギュレーション モードを開始します。新規の VLAN ID を入力して VLAN を作成するか、または既存の VLAN ID を入力してその VLAN を変更します。 （注） このコマンドで指定できる VLAN ID 範囲は 1 ～ 4094 です。

	コマンドまたはアクション	目的
ステップ 3	**name** *vlan-name* 例： `(config-vlan)# name test20`	（任意）VLAN の名前を入力します。VLAN名を指定しなかった場合には、デフォルトとして、VLANという語の後ろに先行ゼロを含めた *vlan-id* 値が付加されます。たとえば、VLAN4 のデフォルトの VLAN 名は VLAN0004 になります。
ステップ 4	**exit** 例： `(config-vlan)# exit`	コンフィギュレーション モードに戻ります。
ステップ 5	**interface interface-id** 例： `router(config)# interface gigabitethernet1/0/1`	設定する物理ポートを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 6	**switchport mode access** 例： `router(config-if)# switchport mode access`	インターフェイスをVLANアクセスポートとして設定します。
ステップ 7	**switchport access vlan** *vlan id* 例： `router(config-if)# switchport access vlan 20`	このアクセス ポートでトラフィックを伝送する VLAN を指定します。このコマンドを入力しないと、アクセスポートはVLAN1 だけのトラフィックを伝送します。このコマンドを使用して、アクセスポートがトラフィックを伝送する VLAN を変更できます。
ステップ 8	**end** 例： `router(config-if)# end`	コンフィギュレーション モードに戻ります。

レイヤ 2 スイッチング用 LAN ポートの設定

ここでは、Cisco 4000 シリーズ ルータでレイヤ 2 スイッチング用に 3 種類すべてのイーサネット LAN ポートを設定する方法について説明します。このセクションの設定タスクは LAN スイッチング モジュール上の LAN ポートに適用されます。

レイヤ 2 LAN ポート モード

次の表に、レイヤ 2 LAN ポートモードを示し、LAN ポートにおける各モードの機能について説明します。

表 *4:* レイヤ *2 LAN* ポート モード

モード	機能
switchport mode access	LAN ポートは永続的な非トランキング モードになり、リンクを非トランク リンクに変換するようにネゴシエーションを行います。ネイバー LAN ポートが変更に同意しなくても、LAN ポートは非トランク ポートになります。
switchport mode dynamic desirable	リンクからトランク リンクへの変換を LAN ポートにアクティブに試行させます。ネイバー LAN ポートが **trunk**、**desirable** または **auto** モードに設定されていれば、LAN ポートはトランクポートになります。このモードは、すべての LAN ポートのデフォルト モードです。
switchport mode dynamic auto	LAN ポートにリンクからトランク リンクへの変換を試行させます。ネイバー LAN ポートが **trunk** または **desirable** モードに設定されていれば、LAN ポートはトランクポートになります。
switchport mode trunk	LAN ポートは永続的なトランキング モードになり、リンクをトランク リンクに変換するようにネゴシエーションを行います。ネイバー ポートが変更に同意しなくても、LAN ポートはトランク ポートになります。
switchport nonegotiate	LAN ポートを永続的なトランキング モードにしますが、LAN ポートが DTP フレームを生成するのを防ぎます。トランク リンクを確立するには、ネイバー ポートを手動でトランク ポートとして設定する必要があります。

（注） DTP はポイントツーポイント プロトコルです。ただし、インターネットワーキング デバイスによっては、DTP フレームが正しく転送されないことがあります。この問題を避けるために、これらのリンク上でトランキングを行わない場合は、DTP をサポートしないデバイスに接続されている LAN ポートが、**access** キーワードを使用して設定されていることを確認してください。DTP をサポートしないデバイスへのトランキングをイネーブルにするには、**nonegotiate** キーワードを使用して、LAN ポートをトランクにし、DTP フレームが生成されないようにします。

レイヤ 2 LAN インターフェイスのデフォルト設定

次の表に、レイヤ 2 LAN ポートのデフォルト設定を示します。

表 *5:* レイヤ *2 LAN* インターフェイスのデフォルト設定

機能	デフォルト
インターフェイス モード：	
• **switchport** コマンドの入力前	
• **switchport** コマンドの入力後	**switchport mode dynamic desirable**
デフォルト アクセス VLAN	VLAN 1
ネイティブ VLAN（802.1Q トランク用）	VLAN 1

レイヤ 2 スイッチング用の LAN インターフェイスの設定

ここでは、Cisco 4000 シリーズ ルータにおけるレイヤ 2 スイッチングの設定手順について説明します。

（注） **default interface** {**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/subslot/port* コマンドを使用して、インターフェイスをデフォルトの設定に戻します。

スパニングツリー プロトコルの概要

スパニングツリープロトコル（STP）は、ネットワーク内のループを回避しながらパスを冗長化するためのレイヤ2リンク管理プロトコルです。レイヤ2イーサネット ネットワークが正常に動作するには、任意の2つのステーション間で存在できるアクティブ パスは1つだけです。エンドステーション間に複数のアクティブパスがあると、ネットワークにループが生じます。このループがネットワークに発生すると、エンドステーションにメッセージが重複して到着する可能性があります。デバイスは、複数のレイヤ 2 インターフェイスのエンドステーション MAC アドレスを学習する可能性もあります。このような状況によって、ネットワークが不安定になります。スパニングツリーの動作は透過的であり、エンドステーション側で、単一LAN セグメントに接続されているのか、複数セグメントからなるスイッチド LAN に接続されているのかを検出することはできません。

STP は、スパニングツリーアルゴリズムを使用し、スパニングツリーのルートとして冗長接続ネットワーク内のデバイスを1つ選択します。アルゴリズムは、次に基づき、各ポートに役割を割り当て、スイッチド レイヤ 2 ネットワークを介して最良のループフリー パスを算出します。 アクティブ トポロジでのポートの役割：

- ルート：スパニングツリー トポロジに対して選定される転送ポート
- 指定：各スイッチド LAN セグメントに対して選定される転送ポート
- 代替：スパニングツリーのルート ブリッジへの代替パスとなるブロック ポート
- バックアップ：ループバック コンフィギュレーションのブロック ポート

すべてのポートに役割が指定されているデバイス、またはバックアップの役割が指定されているデバイスはルートデバイスです。少なくとも *1* つのポートに役割が指定されているデバイスは、指定デバイスを意味します。

冗長データ パスはスパニングツリーによって、強制的にスタンバイ（ブロックされた）ステートにされます。スパニングツリーのネットワーク セグメントでエラーが発生したときに冗長パスが存在する場合は、スパニングツリー アルゴリズムがスパニングツリー トポロジを再計算し、スタンバイ パスをアクティブにします。デバイスは、スパニングツリーフレーム（ブリッジ プロトコル データ ユニット（BPDU）と呼ばれる）を定期間隔で送受信します。デバイスはこれらのフレームを転送せずに、ループのないパスを構成するために使用します。BPDU には、送信側デバイスおよびそのポートについて、デバイスおよび MAC アドレス、デバイスプライオリティ、ポートプライオリティ、パスコストなどの情報が含まれます。スパニングツリーはこの情報を使用して、スイッチドネットワーク用のルートデバイスおよびルートポートを選定し、さらに、各スイッチドセグメントのルートポートおよび指定ポートを選定します。

デバイスの 2 つのポートがループの一部である場合、 spanning-tree および、パス コスト設定は、どのポートがフォワーディング ステートになるか、およびどのポートがブロッキング ステートになるかを制御します。スパニングツリー ポート プライオリティ値は、ネットワーク トポロジにおけるポートの位置とともに、トラフィック転送におけるポートの位置がどれだけ適切であるかを表します。The コスト値は、メディア速度を表します。

（注） デフォルトでは、Small Form-Factor Pluggable（SFP）モジュールを備えていないインターフェイスにだけ、デバイスが（接続が稼働していることを確認するために）キープアライブメッセージを送信します。[**no**] **keepalive** インターフェイス コンフィギュレーション コマンドをキーワードなしで入力すると、インターフェイスのデフォルトを変更できます。

Cisco SM-X-16G4M2X レイヤ 2 ギガビット EtherSwitch サービスモジュールは、すべての VLAN で STP（IEEE 802.1D ブリッジプロトコル）を使用します。デフォルトでは、（STP を手動でディセーブルにしない限り）設定されている VLAN ごとに 1 つの STP インスタンスが動作します。STP は、VLAN 単位でイネーブルおよびディセーブルにすることができます。

STP の詳細については、https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9200/software/release/16-10/configuration_guide/lyr2/b_1610_lyr2_9200_cg/configuring_spanning___tree_protocol.html を参照してください。

STP のデフォルト設定

次の表に、STP のデフォルト設定を記載します。

表 *6 : STP* のデフォルト設定

機能	デフォルト値
ディセーブル状態	すべての VLAN でディセーブル化された STP
ブリッジ プライオリティ	32768

機能	デフォルト値
SPT ポートプライオリティ（ポート単位で設定可能：レイヤ 2 アクセスポートとして設定された LAN ポートで使用される）	128
SPT ポートコスト（ポート単位で設定可能：レイヤ 2 アクセスポートとして設定された LAN ポートで使用される）	ギガビット イーサネット：4
STP VLAN ポートプライオリティ（VLAN 単位で設定可能。レイヤ 2 トランクポートとして設定された LAN ポートで使用される）	128
STP VLAN ポートコスト（VLAN 単位で設定可能。レイヤ 2 トランクポートとして設定された LAN ポートで使用される）	ギガビット イーサネット：1000000000
hello タイム	2 秒
転送遅延時間	15 秒
最大エージング タイム	20 秒
モード	PVST

STP のイネーブル化

（注） デフォルトではすべての VLAN で STP がディセーブルです。

STP は、VLAN 単位でイネーブルにできます。Cisco SM-X-16G4M2X または SM-X-40G8M2X レイヤ 2 Gigabit EtherSwitch サービスモジュールは、VLAN ごとに個別の STP インスタンスを維持します（STP をディセーブルに設定した VLAN を除きます）。

デフォルト モード以外のモードをイネーブルにする場合、この手順は必須です。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**spanning-tree mode** {**pvst** | **mst** | **rapid-pvst**}	スパニングツリーモードを設定します。

	コマンドまたはアクション	目的
		すべてのスタック メンバーは、同じバージョンのスパニング ツリーを実行します。 • PVST+ をイネーブルにするには、**pvst** を選択します。 • MSTP をイネーブルにするには、**mst** を選択します。 • rapid PVST+ をイネーブルにするには、**rapid-pvst** を選択します。
ステップ 3	**interface** *interface-id*	設定するインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。有効なインターフェイスとしては、物理ポート、VLAN、ポート チャネルなどがあります。VLAN ID の範囲は 1 ～ 4094 です。指定できるポートチャネルの範囲は 1 ～ 48 です。
ステップ 4	**spanning-tree link-type point-to-point** 例： `Device(config-if)# spanning-tree link-type point-to-point`	このポートのリンク タイプがポイントツーポイントであることを指定します。 このポート（ローカル ポート）をポイントツーポイント リンクでリモート ポートと接続し、ローカル ポートが指定ポートになると、はリモート ポートとネゴシエーションし、ローカル ポートをフォワーディング ステートにすばやく変更します。
ステップ 5	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 6	**clear spanning-tree detected-protocols** 例： `Device# clear spanning-tree detected-protocols`	デバイス上のいずれかのポートがレガシー IEEE 802.1D デバイス上のポートに接続されている場合は、このコマンドによりデバイス全体のプロトコル移行プロセスを再開します。 このステップは、このデバイスで Rapid PVST+ が稼働していることを指定デバイスが検出する場合のオプションです。

	コマンドまたはアクション	目的
ステップ 7	Device# **show spanning-tree vlan** *vlan_ID*	STP がイネーブルになっていることを確認します。

次のタスク

注意 VLAN のすべてのスイッチおよびブリッジでスパニングツリーがディセーブルになっていない場合は、VLAN でスパニングツリーをディセーブルにしないでください。スパニングツリーは、VLAN の一部のスイッチおよびブリッジでディセーブルにしておきながら、VLAN のその他のスイッチおよびブリッジでイネーブルにしておくことはできません。スパニングツリーをイネーブルにしたスイッチとブリッジに、ネットワークの物理トポロジに関する不完全な情報が含まれることになるので、この処理によって予想外の結果となることがあります。

注意 物理的なループの存在しないトポロジーであっても、スパニングツリーをディセーブルにすることは推奨しません。スパニングツリーは、設定の誤りおよび配線の誤りに対する保護手段として動作します。VLAN に物理ループが存在しないことを確認せずに、VLAN でスパニングツリーをディセーブルにしないでください。

次に、VLAN 200 で STP をイネーブルにする例を示します。

```
Device# configure terminal
Device(config)# spanning-tree vlan 200

Device(config)# end

Device#
```

（注） STP はデフォルトでは無効に設定されています。

次に、設定を確認する例を示します。

```
Device# show spanning-tree vlan 200

G0:VLAN0200
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
             Address     00d0.00b8.14c8
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32768
             Address     00d0.00b8.14c8
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300
Interface        Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
```

```
Gi1/4            Desg FWD 200000    128.196  P2p
Gi1/5            Back BLK 200000    128.197  P2p
Device#
```

（注） VLAN 200 スパニングツリーを作成するには、VLAN 200 にアクティブなインターフェイスが少なくとも 1 つ必要です。この例では、VLAN 200 内の 2 つのインターフェイスがアクティブです。

オプションの STP 機能の設定

ここでは、次のオプションの STP 機能の設定方法について説明します。

PortFast のイネーブル化

注意 PortFast は、単一のエンドステーションをレイヤ 2 アクセスポートに接続する場合に限って使用してください。そうしない場合、ネットワーク ループが発生する可能性があります。

レイヤ 2 アクセスポート上で PortFast をイネーブルにするには、次の作業を行います。

手順

	コマンドまたはアクション	目的
ステップ 1	Router(config)# **interface** {*type* [1] *slot/port* }	設定するポートを選択します。
ステップ 2	Router(config-if)#**spanning-tree portfast**	単一のワークステーションまたはサーバに接続されたレイヤ 2 アクセス ポート上で PortFast をイネーブルにします。
ステップ 3	Router(config-if)#**spanning-tree portfast default**	PortFast をイネーブルにします。
ステップ 4	Router(config-if)# **end**	設定モードを終了します。
ステップ 5	Router# **show running interface** {*type* [2]*slot/port* }	設定を確認します。

PortFast BPDU フィルタリングの設定

ここでは、PortFast BPDU フィルタリングを設定する手順について説明します。

PortFast BPDU フィルタリングをグローバルにイネーブルにするには、次の作業を行います。

手順

	コマンドまたはアクション	目的
ステップ 1	Router(config)# **spanning-tree portfast bpdufilter default**	ルータ上で BPDU フィルタリングをグローバルにイネーブルにします。
ステップ 2	Router# **show spanning-tree summary totals**	設定を確認します。

PortFast BPDU フィルタリングのイネーブル化

各ポート上で、BPDU フィルタリングはデフォルトに設定されています。次に、ポート上で PortFast BPDU フィルタリングをイネーブルにして、PVST+ モードで設定を確認する例を示します。

```
Router(config)# spanning-tree portfast bpdufilter default

Router(config)# ^Z
Router# show spanning-tree summary totals

Switch is in pvst mode
Root bridge for: G0:VLAN0013, G0:VLAN0020, G1:VLAN0020
EtherChannel misconfig guard is enabled
Extended system ID           is enabled
Portfast Default             is disabled
PortFast BPDU Guard Default  is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default            is disabled
UplinkFast                   is disabled
BackboneFast                 is disabled
Pathcost method used         is short
Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
3 vlans                       0         0        0          3          3
```

非トランキング ポート上で PortFast BPDU フィルタリングをイネーブルにするには、次の作業を行います。

手順

	コマンドまたはアクション	目的
ステップ 1	Router(config)# **interface fastEthernet 4/4**	設定するインターフェイスを選択します。
ステップ 2	Router(config-if)#**spanning-tree bpdufilter enable**	BPDUフィルタリングをイネーブルにします。
ステップ 3	Router# **show spanning-tree interface fastEthernet 4/4**	設定を確認します。

次のタスク

次に、非トランキング ポート上で PortFast BPDU フィルタリングをイネーブルにする例を示します。

```
Router(config)# interface fastEthernet 4/4
Router(config-if)# spanning-tree bpdufilter enable

Router(config-if)# ^Z
Router# show spanning-tree interface fastEthernet 4/4
Vlan             Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
VLAN0010         Desg FWD 1000      160.196  Edge P2p
Router# show spanning-tree interface fastEthernet 4/4 detail

 Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
   Port path cost 1000, Port priority 160, Port Identifier 160.196.
   Designated root has priority 32768, address 00d0.00b8.140a
   Designated bridge has priority 32768, address 00d0.00b8.140a
   Designated port id is 160.196, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   The port is in the portfast mode by portfast trunk configuration
   Link type is point-to-point by default
   Bpdu filter is enabled
   BPDU:sent 0, received 0
Router#
```

BPDU ガードのイネーブル化

BPDU ガードをグローバルにイネーブルにするには、次の作業を行います。

手順

	コマンドまたはアクション	目的
ステップ 1	Router(config)# **spanning-tree portfast bpduguard default** 例： `Router(config)# no spanning-tree portfast bpduguard default`	BPDU ガードをグローバルにイネーブルにします。 BPDU ガードをグローバルにディセーブルにします。
ステップ 2	Router(config)#**end**	設定モードを終了します。
ステップ 3	Router# **show spanning-tree summary totals**	設定を確認します。

次のタスク

次に、BPDU ガードをイネーブルにする例を示します。

```
Router# configure terminal
Router(config)# spanning-tree portfast bpduguard
Router(config)# end
Router#
```

次に、設定を確認する例を示します。

```
Router# show spanning-tree summary totals
 default
Root bridge for:VLAN0010
EtherChannel misconfiguration guard is enabled
Extended system ID   is disabled
Portfast             is enabled by default
PortFast BPDU Guard  is disabled by default
Portfast BPDU Filter is enabled by default
Loopguard            is disabled by default
UplinkFast           is disabled
BackboneFast         is disabled
Pathcost method used is long
Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
2 vlans                   0        0         0        3          3
Router#
```

UplinkFast の有効化

UplinkFast を使用すると、ブリッジプライオリティが 49152 に増えるとともに、デバイス上のすべてのレイヤ 2 LAN インターフェイスの STP ポート コストに 3000 が加算されます。その結果、ルータがルートブリッジになる可能性が低くなります。*max_update_rate* 値は、1 秒間に送信されるマルチキャスト パケット数を表します（デフォルトは 150 パケット/秒です）。ブリッジプライオリティを設定している VLAN 上では、UplinkFast をイネーブルにすることはできません。ブリッジ プライオリティを設定している VLAN 上で UplinkFast をイネーブルにするには、グローバル コンフィギュレーション モードで **no spanning-tree vlan** *vlan_ID* **priority** コマンドを入力して、VLAN のブリッジプライオリティをデフォルトに戻します。

（注） UplinkFast をイネーブルにすると、デバイス上のすべての VLAN に影響します。個々の VLAN について UplinkFast を設定することはできません。

UplinkFast をイネーブルにするには、次の作業を行います。

手順

	コマンドまたはアクション	目的
ステップ 1	Router(config)# **spanning-tree uplinkfast** [**max-update-rate** *max_update_rate*]	UplinkFast をイネーブルにします。
ステップ 2	Router(config)# **no spanning-tree uplinkfast max-update-rate**	デフォルトのレートに戻します。
ステップ 3	Router(config)# **no spanning-tree uplinkfast**	UplinkFast をディセーブルにします。
ステップ 4	Router(config)#**end**	設定モードを終了します。
ステップ 5	Router# **show spanning-tree vlan** *vlan_ID*	UplinkFast がイネーブルになっていることを確認します。

次のタスク

次に、UplinkFast をイネーブルにして、アップデート速度を 400 パケット/秒に設定する例を示します。

```
Router# configure terminal

Router(config)# spanning-tree uplinkfast max-update-rate 400

Router(config)# exit

Router#
```

次に、UplinkFast がイネーブルになっていることを確認する例を示します。

```
Router# show spanning-tree uplinkfast

UplinkFast is enabled
Router#
```

BackboneFast のイネーブル化

（注） BackboneFast が適切に動作するのは、ネットワーク内のすべてのネットワーク デバイス上でイネーブルになっている場合だけです。BackboneFast は、トークンリング VLAN ではサポートされません。この機能は、サードパーティ製のネットワーク デバイスと組み合わせて使用することができます。

BackboneFast をイネーブルにするには、次の作業を行います。

手順

	コマンドまたはアクション	目的
ステップ 1	Router(config)# **spanning-tree backbonefast**	BackboneFast をイネーブルにします。
ステップ 2	Router(config)# **no spanning-tree backbonefast**	BackboneFast をディセーブルにします。
ステップ 3	Router(config)#**end**	設定モードを終了します。
ステップ 4	Router# **show spanning-tree vlan** *vlan_ID*	UplinkFast がイネーブルになっていることを確認します。

次のタスク

次に、BackboneFast をイネーブルにする例を示します。

```
Router# configure terminal

Router(config)# spanning-tree backbonefast
```

```
Router(config)# end

Router#
```

次に、BackboneFast がイネーブルになっていることを確認する例を示します。

```
Router# show spanning-tree backbonefast

BackboneFast is enabled
BackboneFast statistics
-----------------------
Number of transition via backboneFast (all VLANs) : 0
Number of inferior BPDUs received (all VLANs)     : 0
Number of RLQ request PDUs received (all VLANs)   : 0
Number of RLQ response PDUs received (all VLANs)  : 0
Number of RLQ request PDUs sent (all VLANs)       : 0
Number of RLQ response PDUs sent (all VLANs)      : 0

Router#
```

EtherChannel の概要

EtherChannel は、スイッチ、ルータ、およびサーバ間にフォールトトレラントな高速リンクを提供します。EtherChannel を使用して、ワイヤリング クローゼットとデータセンター間の帯域幅を増やすことができます。さらに、ボトルネックが発生しやすいネットワーク上のあらゆる場所に EtherChannel を配置できます。EtherChannel は、他のリンクに負荷を再分散させることによって、リンク切断から自動的に回復します。リンク障害が発生した場合、EtherChannel は自動的に障害リンクからチャネル内の他のリンクにトラフィックをリダイレクトします。

EtherChannel は、単一の論理リンクにバンドルする個別のイーサネットリンクで構成されます。

EtherChannel は、スイッチ間またはスイッチとホスト間に、最大 4 Gbps（ギガビット EtherChannel）の全二重帯域幅を提供します。

各 EtherChannel は、互換性のある設定のイーサネットポートを 4 つまで使用して構成できます。

チャネル グループおよびポートチャネル インターフェイス

EtherChannel は、チャネル グループとポートチャネル インターフェイスから構成されます。チャネル グループはポートチャネル インターフェイスに物理ポートをバインドします。ポートチャネル インターフェイスに適用した設定変更は、チャネル グループにまとめてバインドされるすべての物理ポートに適用されます。channel-group コマンドは、物理ポートおよびポートチャネル インターフェイスをまとめてバインドします。各 EtherChannel には 1 ～ 32 番のポートチャネル論理インターフェイスがあります。ポートチャネル インターフェイス番号は、channel-group インターフェイス コンフィギュレーション コマンドで指定した番号に対応しています。

Port Aggregation Protocol; ポート集約プロトコル

ポート集約プロトコル（PAgP）はシスコ独自のプロトコルで、Cisco デバイスおよび PAgP をサポートするベンダーによってライセンス供与されたデバイスでのみ稼働します。PAgP を使

用すると、イーサネット ポート間で PAgP パケットを交換することにより、EtherChannel を自動的に作成できます。

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

Link Aggregation Control Protocol

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

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

Auto-LAG

Auto-LAG 機能は、スイッチに接続されたポートで EtherChannel を自動的に作成できる機能です。デフォルトでは、Auto-LAG がグローバルに無効にされ、すべてのポート インターフェイスで有効になっています。Auto-LAG は、グローバルに有効になっている場合にのみ、スイッチに適用されます。

Auto-LAG をグローバルに有効にすると、次のシナリオが可能になります。

- パートナー ポート インターフェイス上に EtherChannel が設定されている場合、すべてのポートインターフェイスが自動 EtherChannel の作成に参加します。詳細については、次の表「アクターとパートナーデバイス間でサポートされる Auto-LAG 設定」を参照してください。
- すでに手動 EtherChannel の一部であるポートは、自動 EtherChannel の作成に参加することはできません。
- Auto-LAG がすでに自動で作成された EtherChannel の一部であるポート インターフェイスで無効になっている場合、ポートインターフェイスは自動 EtherChannel からバンドル解除されます。

- 次の表に、アクターとパートナー デバイス間でサポートされる Auto-LAG 設定を示します。

表 *7:* アクターとパートナー デバイス間でサポートされる *Auto-LAG* 設定

アクター/パートナー	アクティブ	パッシブ	自動
アクティブ	対応	対応	対応
パッシブ	対応	不可	対応
自動	対応	対応	対応

Auto-LAG をグローバルに無効にすると、自動で作成されたすべての Etherchannel が手動 EtherChannel になります。

既存の自動で作成された EtherChannel で設定を追加することはできません。追加するには、最初に **port-channel**<*channel-number*>**persistent** を実行して、手動 EtherChannel に変換する必要があります。

レイヤ 2 EtherChannel の設定

レイヤ 2 EtherChannel を設定するには、インターフェイス コンフィギュレーション モードで **channel-group** コマンドを使用して、チャネルグループにポートを割り当てます。このコマンドにより、ポートチャネル論理インターフェイスが自動的に作成されます。

Cisco SM-X-16G4M2X EtherChannel を表示するには、**show etherchannel swport xxx** コマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	物理ポートを指定し、インターフェイス コンフィギュレーション モードを開始します。 指定できるインターフェイスは、物理ポートです。

	コマンドまたはアクション	目的
		PAgP EtherChannel の場合、同じタイプおよび速度のポートを 4 つまで同じグループに設定できます。 LACP EtherChannel の場合、同じタイプのイーサネット ポートを 8 まで設定できます。最大 8 つのポートを active モードに、最大 8 つのポートを standby モードにできます。
ステップ 4	**switchport mode** {**access** | **trunk**} 例： `Device(config-if)# switchport mode access`	すべてのポートをスタティックアクセス ポートとして同じ VLAN に割り当てるか、またはトランクとして設定します。 ポートをスタティックアクセス ポートとして設定する場合は、ポートを 1 つの VLAN にのみ割り当ててください。指定できる範囲は 1 ～ 4094 です。
ステップ 5	**switchport access vlan** *vlan-id* 例： `Device(config-if)# switchport access vlan 22`	ポートをスタティックアクセス ポートとして設定する場合は、ポートを 1 つの VLAN にのみ割り当ててください。指定できる範囲は 1 ～ 4094 です。
ステップ 6	**channel-group** *channel-group-number* **mode** {**auto** [**non-silent**] | **desirable** [**non-silent**] | **on** } | { **active** | **passive**} 例： `Device(config-if)# channel-group 5 mode auto`	チャネル グループにポートを割り当て、PAgP モードまたは LACP モードを指定します。 **mode** には、次のキーワードのいずれか 1 つを選択します。 • **auto** –PAgP 装置が検出された場合に限り、PAgP をイネーブルにします。ポートをパッシブ ネゴシエーション ステートにします。この場合、ポートは受信する PAgP パケットに応答しますが、PAgP パケット ネゴシエーションを開始することはありません。 • **desirable** –無条件に PAgP をイネーブルにします。ポートをアクティブ ネゴシエーション ステートにします。この場合、ポートは PAgP パケットを送信することによって、相手ポートとのネゴシエーションを開始します。

	コマンドまたはアクション	目的
		• **on** – : PAgP または LACP を使用せずにポートが強制的にチャネル化されます。**on** モードでは、使用可能な EtherChannel が存在するのは、**on** モードのポートグループが、**on** モードの別のポートグループに接続する場合だけです。 • **non-silent** –（任意）デバイスが PAgP 対応のパートナーに接続されている場合、ポートが **auto** または **desirable** モードになると非サイレント動作を行うようにデバイスポートを設定します。**non-silent** を指定しなかった場合は、サイレントが指定されたものと見なされます。サイレント設定は、ファイル サーバまたはパケット アナライザとの接続に適しています。サイレントを設定すると、PAgP が動作してチャネル グループにポートを結合し、このポートが伝送に使用されます。 • **active**：LACP 装置が検出された場合に限り、LACP をイネーブルにします。ポートをアクティブ ネゴシエーション ステートにします。この場合、ポートは LACP パケットを送信することによって、相手ポートとのネゴシエーションを開始します。 • **passive** – : ポート上で LACP をイネーブルにして、ポートをパッシブ ネゴシエーション ステートにします。この場合、ポートは受信する LACP パケットに応答しますが、LACP パケット ネゴシエーションを開始することはありません。
ステップ 7	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

EtherChannel ロード バランシングの設定

複数の異なる転送方式の 1 つを使用するように EtherChannel ロードバランシングを設定できます。

このタスクはオプションです。

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	**port-channel swport load-balance** { **dst-ip** | **dst-mac** | **dst-mixed-ip-port** | **dst-port** | **extended** [**dst-ip** | **dst-mac** | **dst-port** | **ipv6-label** | **l3-proto** | **src-ip** | **src-mac** | **src-port**] | **src-dst-ip** | **src-dst-mac** **src-dst-mixed-ip-port src-dst-portsrc-ip** | **src-mac** | **src-mixed-ip-port** | **src-port** } 例： `Device(config)# port-channel swport load-balance src-mac`	EtherChannel のロードバランシング方式を設定します。 次のいずれかの負荷分散方式を選択します。 • **dst-ip**：宛先ホストの IP アドレスを指定します。 • **dst-mac**：着信パケットの宛先ホストの MAC アドレスを指定します。 • **dst-mixed-ip-port**：ホストの IP アドレスおよび TCP/UDP ポートを指定します。 • **dst-port**：宛先 TCP/UDP ポートを指定します。 • **extended**：標準コマンドで使用可能なもの以外に、送信元および宛先の方式を組み合わせた、拡張ロードバランシング方式を指定します。 • **ipv6-label**：IPv6 フロー ラベルを指定します。 • **l3-proto**：レイヤ 3 プロトコルを指定します。 • **src-dst-ip**：送信元および宛先ホストの IP アドレスを指定します。 • **src-dst-mac**：送信元および宛先ホストの MAC アドレスを指定します。

	コマンドまたはアクション	目的
		• **src-dst-mixed-ip-port**：送信先および宛先ホストの IP アドレスおよび TCP/UDP ポートを指定します。 • **src-dst-port**：送信元および宛先 TCP/UDP ポートを指定します。 • **src-ip**：送信元ホストの IP アドレスを指定します。 • **src-mac**：着信パケットの送信元 MAC アドレスを指定します。 • **src-mixed-ip-port**：送信元ホストの IP アドレスおよび TCP/UDP ポートを指定します。 • **src-port**：送信元 TCP/UDP ポートを指定します。
ステップ 3	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

PAgP 学習方式およびプライオリティの設定

このタスクはオプションです。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/2`	伝送ポートを指定し、インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**pagp learn-method physical-port** 例： `Device(config-if)# pagp learn-method physical port`	PAgP 学習方式を選択します。 デフォルトでは、**aggregation-port learning** が選択されています。つまり、EtherChannel 内のポートのいずれかを使用して、デバイスがパケットを送信元に送信します。集約ポート ラーナーの場合、どの物理ポートにパケットが届くかは重要ではありません。 is物理ポートラーナーである別のデバイスに接続する**physical-port**を選択します。 **port-channel load-balance** グローバル コンフィギュレーション コマンドを **src-mac** に設定してください。 学習方式はリンクの両端で同じ方式に設定する必要があります。
ステップ 5	**pagp port-priority** *priority* 例： `Device(config-if)# pagp port-priority 200`	選択したポートがパケット伝送用として選択されるように、プライオリティを割り当てます。 *priority* に指定できる範囲は 0 ～ 255 です。デフォルト値は 128 です。プライオリティが高いほど、ポートが PAgP 伝送に使用される可能性が高くなります。
ステップ 6	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

LACP ポートチャネルの最小リンク機能の設定

リンク アップ状態で、リンク アップ ステートに移行するポート チャネル インターフェイスの EtherChannel でバンドルする必要のあるアクティブ ポートの最小数を指定できます。EtherChannel の最小リンクを使用して、低帯域幅 LACP EtherChannel がアクティブになることを防止できます。また、LACP EtherChannel にアクティブメンバーポートが少なすぎて、必要な最低帯域幅を提供できない場合、この機能により LACP EtherChannel が非アクティブになります。

ポート チャネルに必要なリンクの最小数を設定する。次の作業を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface port-channel** *channel-number* 例： `Device(config)# interface port-channel 2`	ポートチャネルのインターフェイス コンフィギュレーション モードを開始します。 *channel-number* の範囲は 1 ～ 63 です。
ステップ 4	**port-channel min-links** *min-links-number* 例： `Device(config-if)# port-channel min-links 3`	リンク アップ状態で、リンク アップ ステートに移行するポート チャネル インターフェイスの EtherChannel でバンドルする必要のあるメンバ ポートの最小数を指定できます。 *min-links-number* の範囲は 2 ～ 8 です。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

LACP 高速レート タイマーの設定

LACP タイマー レートを変更することにより、LACP タイムアウトの時間を変更することができます。**lacp rate** コマンドを使用し、LACP がサポートされているインターフェイスで受信される LACP 制御パケットのレートを設定します。タイムアウト レートは、デフォルトのレート（30 秒）から高速レート（1 秒）に変更することができます。このコマンドは、LACP がイネーブルになっているインターフェイスでのみサポートされます。

手順

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

	コマンドまたはアクション	目的
	`Device# configure terminal`	
ステップ 3	**interface** {**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* 例： `Device(config)# interface gigabitEthernet 2/1`	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。
ステップ 4	**lacp rate** {**normal** | **fast**} 例： `Device(config-if)# lacp rate fast`	LACP がサポートされているインターフェイスで受信される LACP 制御パケットのレートを設定します。 タイムアウトレートをデフォルトにリセットするには、**no lacp rate** コマンドを使用します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

ステップ 6	**show lacp internal**
	例：
	`Device# show lacp internal`
	`Device# show lacp counters`
		設定を確認します。

グローバルな Auto-LAG の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	[**no**] **port-channel swport auto** 例： `Device(config)# port-channel swport auto`	スイッチ上の Auto-LAG 機能をグローバルで有効にします。スイッチ上の Auto-LAG 機能をグローバルで無効にするには、このコマンドの no 形式を使用します。

	コマンドまたはアクション	目的
		（注） デフォルトでは、auto-LAG機能は各ポート上でイネーブルになっています。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show etherchannel swport auto** 例： `Device# show etherchannel swport auto`	EtherChannel が自動的に作成されたことが表示されます。

モジュラ QoS コマンドライン インターフェイス

MQC（モジュラ QoS コマンドライン インターフェイス（CLI））では、QoS グループ値に基づいてパケット分類とマーキングを設定できます。デバイスでは、QoS 機能はモジュラ QoS コマンドラインインターフェイス（MQC）を使用してイネーブルにできます。MQCはコマンドラインインターフェイス（CLI）構造を採用しています。これを使用すると、トラフィックポリシーを作成し、作成したポリシーをインターフェイスにアタッチできます。1 つのトラフィック ポリシーには、1 つのトラフィック クラスと 1 つ以上の QoS 機能が含まれます。トラフィック クラスがトラフィックを分類するために使用されるのに対して、トラフィック ポリシーの QoS 機能は分類されたトラフィックの処理方法を決定します。MQC の主な目的の 1 つは、プラットフォームに依存しないインターフェイスを提供することにより、シスコプラットフォーム全体の QoS を設定することです。モジュラ QoS の詳細については、『Quality of Service Configuration Guide, Cisco IOS XE Fuji 16.9.x』を参照してください。

トラフィック クラスの作成

一致基準が含まれるトラフィッククラスを作成するには、**class-map** コマンドを使用してトラフィッククラス名を指定し、必要に応じて、次の **match** コマンドをクラスマップ コンフィギュレーション モードで使用します。

始める前に

この設定作業で指定するすべての match コマンドの使用は任意ですが、1 つのクラスに少なくとも 1 つの一致基準を設定する必要があります。

手順

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

	コマンドまたはアクション	目的
	`Device# configure terminal`	

ステップ 2	**class-map** *class-map name* {**match-any** }
	例：
	`Device(config)# class-map type ngsw-qos test_1000`
	`Device(config-cmap)#`
		クラス マップ コンフィギュレーション モードを開始します。
		• 名前を指定したクラスとパケットとの照合に使用されるクラス マップを作成します。
		• **match-any**：トラフィック クラスで受信したトラフィックがその一部と分類されるには、一致基準のいずれかを満たす必要があります。

ステップ 3	**match access-group** {*index number* | *name*}
	例：
	`Device(config-cmap)# match access-group 100`
	`Device(config-cmap)#`
		このコマンドでは次のパラメータを使用できます。
		• access-group
		• cos
		• dscp
		• group-object
		• ip
		• mpls
		• precedence
		• protocol
		• qos-group
		• vlan
		• wlan
		（任意）この例では、アクセス グループ ID を入力します。
		• アクセス リスト インデックス（1 ～ 2799 の値）
		• 名前付きアクセス リスト

ステップ 4	**match cos** *CoS* 値 例： `Device(config-cmap)# match cos 2 3 4`	（任意）IEEE 802.1Q または ISL サービス クラス（ユーザ）プライオリティ値に一致します。

	コマンドまたはアクション	目的
	5 `Device(config-cmap)#`	• 最大 4 つの CoS 値（0 ～ 7）をスペースで区切って入力します。

ステップ 5	**match dscp** *DSCP 値*
	例：
	`Device(config-cmap)# match dscp af11 af12`
	`Device(config-cmap)#`
		（任意）IPv4 および IPv6 パケットの DSCP 値に一致します。

ステップ 6	**match ip** {**dscp** *dscp value* | **precedence** *precedence value* }
	例：
	`Device(config-cmap)# match ip dscp af11 af12`
	`Device(config-cmap)#`
		（任意）次を含む IP 値に一致します。
		• **dscp**：IP DSCP（DiffServ コードポイント）に一致します。
		• **precedence**：IP precedence（0 ～ 7）に一致します。

ステップ 7	**match qos-group** *QoS グループ値*
	例：
	`Device(config-cmap)# match qos-group 10`
	`Device(config-cmap)#`
		（任意）QoS グループ値（0 ～ 31）に一致します。

ステップ 8	**match vlan** *vlan value*
	例：
	`Device(config-cmap)# match vlan 210`
	`Device(config-cmap)#`
		（任意）VLAN ID（1 ～ 4095）に一致します。

ステップ 9	**end** 例： `Device(config-cmap)# end`	設定の変更内容を保存します。

次のタスク

ポリシー マップを設定します。

トラフィック ポリシーの作成

トラフィックポリシーを作成するには、**policy-map** グローバル コンフィギュレーション コマンドを使用して、トラフィックポリシーの名前を指定します。

トラフィッククラスは、**class**コマンドを使用したときにトラフィックポリシーと関連付けられます。**class** コマンドは、ポリシー マップ コンフィギュレーション モードを開始した後に実行しなければなりません。**class** コマンドを入力すると、デバイスが自動的にポリシー マップ クラス コンフィギュレーション モードを開始します。ここでトラフィックポリシーの QoS ポリシーを定義します。

次のポリシー マップ クラスのアクションがサポートされます。

- bandwidth：帯域幅設定オプション。
- exit：QoS クラス アクション コンフィギュレーション モードを終了します。
- no：コマンドのデフォルト値を無効にするか、設定します。
- police：ポリシング機能の設定オプション。
- priority：このクラスの完全スケジューリング プライオリティの設定オプション。
- queue-buffers：キューのバッファ設定オプション。
- queue-limit：重み付けテール ドロップ（WTD）設定オプションのキューの最大しきい値。
- service-policy：QoS サービス ポリシーを設定します。
- set：次のオプションを使用して QoS 値を設定します。
  - CoS 値
  - DSCP 値
  - precedence 値
  - QoS グループ値
- shape：トラフィック シェーピング設定オプション。

始める前に

最初にクラス マップを作成する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**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 2	**policy-map type***policy-map name* 例： `Device(config)# policy-map type ngsw-qos test_1000`	ポリシーマップ コンフィギュレーション モードを開始します。 1 つ以上のインターフェイスに対応付けることができるポリシーマップを作成または修正し、サービス ポリシーを指定します。
ステップ 3	**class** {*class-name* | **class-default**} 例： `Device(config-pmap)# class test_1000`	ポリシーを作成または変更するクラスの名前を指定します。 未分類のパケットのシステムデフォルト クラスも作成できます。
ステップ 4	**bandwidth** {**kb/s** *kb/s value* | **percent** *percentage* | **remaining** {*percent* | *ratio*}} 例： `Device(config-pmap-c)# bandwidth 50`	（任意）次のいずれかを使用して帯域幅を設定します。 • **kb/s**：kpbs に 20000 ～ 10000000 の値を入力します。 • **percent**：このポリシーマップに使用される総帯域幅の割合を入力します。 • **remaining**：残りの帯域幅の割合を入力します。
ステップ 5	**exit** 例： `Device(config-pmap-c)# exit`	（任意）QoS クラス アクション コンフィギュレーション モードを終了します。
ステップ 6	**no** 例： `Device(config-pmap-c)# no`	（任意）コマンドを無効にします。
ステップ 7	**police** {*target_bit_rate* | **cir** | **rate**} 例： `Device(config-pmap-c)# police 100000`	（任意）ポリサーを設定します。 • *target_bit_rate*：ビット レート/秒を入力します。8000 ～ 10000000000 の値を入力します。 • **cir**：認定情報レート。 • **rate**：ポリシング レート、階層型ポリシーの PCR、またはシングル

	コマンドまたはアクション	目的
		レベルの ATM 4.0 ポリサー ポリシーの SCR を指定します。
ステップ 8	例： `Device(config-pmap-c)#`	（任意）このクラスに完全スケジューリングプライオリティを設定します。コマンド オプションは次のとおりです。 • **level**：マルチレベル プライオリティ キューを確立します。値を入力します（1 または 2）。
ステップ 9	**queue-buffers ratio***ratio limit* 例： `Device(config-pmap-c)# queue-buffers ratio 10`	（任意）クラスのキューバッファを設定します。キューバッファの割合制限（0 ～ 100）を入力します。
ステップ 10	**queue-limit** {*packets* | **cos** | **dscp** | **percent**} 例： `Device(config-pmap-c)# queue-limit cos 7 percent 50`	（任意）テール ドロップに対してキューの最大しきい値を指定します。 • *packets*：デフォルトのパケット数。1 ～ 2000000 の間の値を入力します。 • **cos**：各 CoS 値のパラメータを入力します。 • **dscp**：各 DSCP 値のパラメータを入力します。 • **percent**：しきい値の割合を入力します。
ステップ 11	**service-policy** *policy-map name* 例： `Device(config-pmap-c)# service-policy test_2000`	（任意）QoS サービス ポリシーを設定します。
ステップ 12	**set** {**cos** | **dscp** | **ip** | **precedence** | **qos-group** | **wlan**} 例： `Device(config-pmap-c)# set cos 7`	（任意）QoS 値を設定します。使用可能な QoS 設定値は次のとおりです。 • **cos**：IEEE 802.1Q/ISL サービス クラスまたはユーザプライオリティを設定します。

	コマンドまたはアクション	目的
		• **dscp**：IP（v4）およびIPv6パケットのDSCPを設定します。 • **ip**：IP固有の値を設定します。 • **precedence**：IP（v4）およびIPv6パケットのprecedenceを設定します。 • **qos-group**：QoSグループを設定します。
ステップ13	**shape average** {*target _bit_rate* | **percent**} 例： `Device(config-pmap-c) #shape average percent 50`	（任意）トラフィックシェーピングを設定します。コマンドパラメータは次のとおりです。 • *target_bit_rate*：ターゲットビットレート。 • **percent**：認定情報レートのインターフェイス帯域幅の割合。
ステップ14	**end** 例： `Device(config-pmap-c) #end`	設定の変更内容を保存します。

次のタスク

インターフェイスを設定します。

クラスベース パケット マーキングの設定

この手順は、クラスベース パケット マーキング機能をデバイスで設定する方法を説明している重要な手順です。

- CoS 値
- DSCP 値
- IP 値
- precedence 値
- QoS グループ値
- WLAN 値

始める前に

この手順を開始する前にクラス マップとポリシー マップを作成する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**policy-maptype***policy name*
	例：
	`Device(config)# policy-map type ngsw-qos policy1`
	`Device(config-pmap)#`
		ポリシーマップ コンフィギュレーション モードを開始します。
		1 つ以上のインターフェイスに対応付けることができるポリシーマップを作成または修正し、サービスポリシーを指定します。

ステップ 3	**class** *class name* 例： `Device(config-pmap)# class class1`	ポリシー クラス マップ コンフィギュレーションモードを開始します。ポリシーを作成または変更するクラスの名前を指定します。 ポリシー クラス マップ コンフィギュレーションモードには、次のコマンドオプションが含まれます。 • **bandwidth**：帯域幅設定オプション。 • **exit**：QoS クラス アクション コンフィギュレーションモードを終了します。 • **no**：コマンドのデフォルト値を無効にするか、設定します。 • **police**：ポリシング機能の設定オプション。 • **priority**：このクラスの完全スケジューリング プライオリティの設定オプション。 • **queue-buffers**：キューのバッファ設定オプション。

	コマンドまたはアクション	目的
		• **queue-limit**：重み付けテール ドロップ（WTD）設定オプションのキューの最大しきい値。 • **service-policy**：QoS サービス ポリシーを設定します。 • **set**：次のオプションを使用して QoS 値を設定します。 • CoS 値 • DSCP 値 • precedence 値 • QoS グループ値 • WLAN 値 • **shape**：トラフィック シェーピング設定オプション。 （注） この手順では、**set** コマンドオプションを使用して、使用可能な設定について説明します。その他のコマンドオプション（**bandwidth**）についてはこのマニュアルの他の項で説明します。このタスクでは、使用可能なすべての **set** コマンドが表示されますが、クラス単位でサポートされるのは 1 つの **set** コマンドだけです。
ステップ 4	**set cos** {*cos value* | **cos table** *table-map name* | **dscp table** *table-map name* | **precedence table** *table-map name* | **qos-group table** *table-map name* | **wlan user-priority table** *table-map name*} 例： `Device(config-pmap)# set cos 5`	（任意）発信パケットの固有の IEEE 802.1Q レイヤ 2 CoS 値を設定します。値は 0 ～ 7 です。 **set cos** コマンドを使用して次の値を設定することもできます。 • **cos table**：CoS 値をテーブル マップに基づいて設定します。

	コマンドまたはアクション	目的
		• **dscp table**：コード ポイント値をテーブルマップに基づいて設定します。 • **precedence table**：コードポイント値をテーブルマップに基づいて設定します。 • **qos-group table**：テーブル マップに基づいて QoS グループから CoS 値を設定します。 • **wlan user-priority table**：テーブルマップに基づいて WLAN ユーザ プライオリティから CoS 値を設定します。
ステップ 5	**set dscp** {*dscp value* | **default** | **dscp table** *table-map name* | **ef** | **precedence table** *table-map name* | **qos-group table** *table-map name* | **wlan user-priority table** *table-map name*} 例： `Device(config-pmap)# set dscp af11`	（任意）DSCP 値を設定します。 特定の DSCP 値の設定に加えて、**set dscp** コマンドを使用して次を設定できます。 • **default**：パケットをデフォルト DSCP 値（000000）と一致させます。 • **dscp table**：テーブルマップに基づいて DSCP からパケットの DSCP 値を設定します。 • **ef**：パケットを EF DSCP 値（101110）と一致させます。 • **precedence table**：テーブルマップに基づいて優先順位からパケットの DSCP 値を設定します。 • **qos-group table**：テーブル マップに基づいて QoS グループからパケットの DSCP 値を設定します。 • **wlan user-priority table**：パケットの DSCP 値を、テーブル マップに基づいた WLAN ユーザ プライオリティに基づいて設定します。

	コマンドまたはアクション	目的
ステップ 6	**set ip** {**dscp** | **precedence**} 例： `Device(config-pmap)# set ip dscp c3`	（任意）IP 固有の値を設定します。これらの値は、IP DSCP 値または IP precedence 値です。 **set ip dscp** コマンドを使用して、次の値を設定することができます。 • *dscp value*：特定の DSCP の値を設定します。 • **default**：パケットをデフォルト DSCP 値（000000）と一致させます。 • **dscp table**：テーブル マップに基づいて DSCP からパケットの DSCP 値を設定します。 • **ef**：パケットを EF DSCP 値（101110）と一致させます。 • **precedence table**：テーブル マップに基づいて優先順位からパケットの DSCP 値を設定します。 • **qos-group table**：テーブル マップに基づいて QoS グループからパケットの DSCP 値を設定します。 • **wlan user-priority table**：パケットの DSCP 値を、テーブル マップに基づいた WLAN ユーザ プライオリティに基づいて設定します。 **set ip precedence** コマンドを使用して、次の値を設定することができます。 • *precedence value*：precedence 値を設定します（0 ～ 7）。 • **cos table**：テーブル マップに基づいてレイヤ 2 CoS からパケットの precedence 値を設定します。 • **dscp table**：テーブル マップに基づいて DSCP 値からパケットの precedence 値を設定します。

	コマンドまたはアクション	目的
		• **precedence table**：テーブル マップに基づいて優先順位から precedence 値を設定します。 • **qos-group table**：テーブル マップに基づいて QoS グループから precedence 値を設定します。
ステップ 7	**set precedence** {*precedence value* | **cos table** *table-map name* | **dscp table** *table-map name* | **precedence table** *table-map name* | **qos-group table** *table-map name*} 例： `Device(config-pmap)# set precedence 5`	（任意）IPv4 と IPv6 パケットの precedence 値を設定します。 **set precedence** コマンドを使用して、次の値を設定することができます。 • *precedence value*：precedence 値を設定します（0 ～ 7）。 • **cos table**：レイヤ 2 CoS からのパケットの precedence 値をテーブル マップに基づいて設定します。 • **dscp table**：テーブルマップに基づいて DSCP 値からパケットの precedence 値を設定します。 • **precedence table**：テーブルマップに基づいて優先順位から precedence 値を設定します。 • **qos-group table**：テーブル マップに基づいて QoS グループから precedence 値を設定します。
ステップ 8	**set qos-group** {*qos-group value* | **dscp table** *table-map name* | **precedence table** *table-map name*} 例： `Device(config-pmap)# set qos-group 10`	（任意）QoS グループ値を設定します。このコマンドを使用して次の値を設定できます。 • *qos-group value*：1 から 31 までの数。 • **dscp table**：テーブルマップに基づいて DSCP からコードポイント値を設定します。 • **precedence table**：テーブルマップに基づいて優先順位からコードポイント値を設定します。

	コマンドまたはアクション	目的
ステップ 9	**set wlan user-priority** {*wlan user-priority value* | **cos table** *table-map name* | **dscp table** *table-map name* | **qos-group table** *table-map name* | **wlan table** *table-map name*} 例： `Device(config-pmap)# set wlan user-priority 1`	（任意）WLANユーザプライオリティ値を設定します。このコマンドを使用して次の値を設定できます。 • *wlan user-priority value*：0 ～ 7 の範囲の値。 • **cos table**：テーブル マップに基づいて Cos から WLAN ユーザ プライオリティ値を設定します。 • **dscp table**：テーブルマップに基づいて DSCP から WLAN ユーザ プライオリティ値を設定します。 • **qos-group table**：テーブル マップに基づいて QoS グループからWLANユーザプライオリティ値を設定します。 • **wlan table**：テーブル マップに基づいて WLAN ユーザ プライオリティから WLAN ユーザ プライオリティ値を設定します。
ステップ 10	**end** 例： `Device(config-pmap)# end`	設定変更を保存します。
ステップ 11	**show policy-map** 例： `Device# show policy-map`	（任意）すべてのサービス ポリシーに設定されたすべてのクラスに関するポリシー設定情報を表示します。

次のタスク

service-policy コマンドを使用して、インターフェイスにトラフィック ポリシーを付加します。

トラフィック ポリシーのインターフェイスへの適用

トラフィッククラスとトラフィックポリシーの作成後、**service-policy** インターフェイス コンフィギュレーション コマンドを使用して、トラフィックポリシーをインターフェイスに付加し、ポリシーを適用する方向を指定します（インターフェイスに着信するパケットまたはインターフェイスから送信されるパケット）。

始める前に

インターフェイスにトラフィック ポリシーを付加する前に、トラフィック クラスとトラフィック ポリシーを作成する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type* 例：	
ステップ 3	**service-policy** { **input** *policy-map* | **output** *policy-map*} 例： `Device(config-if)# service-policy output policy_map_01`	ポリシー マップを入力または出力インターフェイスに適用します。このポリシー マップは、そのインターフェイスのサービス ポリシーとして使用されます。 この例では、トラフィック ポリシーでそのインターフェイスから送信されるすべてのトラフィックを評価します。
ステップ 4	**end** 例： `Device(config-if)# end`	設定変更を保存します。
ステップ 5	**show policy map** 例： `Device# show policy map`	（任意）指定されたインターフェイスのポリシーの統計情報を表示します。

次のタスク

他のトラフィック ポリシーをインターフェイスに付加し、ポリシーを適用する方向を指定します。

ポリシー マップによる物理ポートのトラフィックの分類、ポリシング、およびマーキング

実行対象となるトラフィック クラスを指定する非階層型ポリシー マップを、物理ポート上に設定できます。サポートされるアクションは再マーキングとポリシングです。

始める前に

この手順を開始する前に、ネットワーク トラフィックの分類、ポリシング、およびマーキングについて、あらかじめポリシー マップによって決定しておく必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**class-map** {*class-map name* | **match-any**}
	例：
	`Device(config)# class-map ipclass1`
	`Device(config-cmap)# exit`
		クラスマップ コンフィギュレーション モードを開始します。
		• 名前を指定したクラスとパケットとの照合に使用されるクラスマップを作成します。
		• **match-any** を指定すると、トラフィック クラスで受信したトラフィックの場合、一致基準の 1 つに必ず一致し、そのトラフィック クラスの一部と分類されます。これはデフォルトです。

ステップ 3	**match access-group** { *access list index* | *access list name* }
	例：
	`Device(config-cmap)# match access-group 1000`
	`Device(config-cmap)# exit`
		このコマンドでは次のパラメータを使用できます。
		• access-group
		• cos
		• dscp
		• group-object
		• ip
		• mpls
		• precedence
		• protocol
		• qos-group
		• vlan
		• wlan

	コマンドまたはアクション	目的
		（任意）この例では、アクセス グループ ID を入力します。 • アクセス リスト インデックス（1 ～ 2799 の値） • 名前付きアクセス リスト
ステップ 4	**policy-map** *policy-map-name* 例： `Device(config)# olicy-map type ngsw-qos flowit`	ポリシー マップ名を入力することによってポリシーマップを作成し、ポリシーマップ コンフィギュレーション モードを開始します。 デフォルトでは、ポリシーマップは定義されていません。
ステップ 5	**class** {*class-map-name* | **class-default**} 例： `Device(config-pmap)# class ipclass1`	トラフィックの分類を定義し、ポリシーマップ クラス コンフィギュレーション モードを開始します。 デフォルトでは、ポリシーマップ クラス マップは定義されていません。 すでに **class-map** グローバル コンフィギュレーション コマンドを使用してトラフィック クラスが定義されている場合は、このコマンドで *class-map-name* にその名前を指定します。 **class-default** トラフィック クラスは定義済みで、どのポリシーにも追加できます。このトラフィック クラスは、常にポリシー マップの最後に配置されます。暗黙の **match any** が **class-default** クラスに含まれている場合、他のトラフィック クラスと一致しないパケットはすべて **class-default** と一致します。
ステップ 6	**set** {**cos** | **dscp** | **ip** | **precedence** | **qos-group** | **wlan user-priority**} 例： `Device(config-pmap-c)# set dscp 45`	（任意）QoS 値を設定します。使用可能な QoS 設定値は次のとおりです。 • **cos**：IEEE 802.1Q/ISL サービス クラスまたはユーザプライオリティを設定します。 • **dscp**：IP（v4）および IPv6 パケットの DSCP を設定します。 • **ip**：IP 固有の値を設定します。

	コマンドまたはアクション	目的
		• **precedence**：IP（v4）および IPv6 パケットの precedence を設定します。 • **qos-group**：QoS グループを設定します。 • **wlan user-priority**：WLAN ユーザ プライオリティを設定します。 この例では、 **set dscp** コマンドが、パケットでの新しい DSCP 値を設定して IP トラフィックを分類します。
ステップ 7	**police** {*target_bit_rate* | **cir** | **rate** } 例： `Device(config-pmap-c)# police 100000 conform-action transmit exceed-action drop`	（任意）ポリサーを設定します。 • *target_bit_rate*：ビット レート/秒を指定し、8000 ～ 10000000000 の値を入力します。 • **cir**：認定情報レート。 • **rate**：階層型ポリシーのポリシング レート PCR を指定します。 この例では、**police** コマンドが 100000 セットのターゲットビットレートを超えるトラフィックがドロップされるクラスにポリサーを追加します。
ステップ 8	**exit** 例： `Device(config-pmap-c)# exit`	ポリシーマップ コンフィギュレーション モードに戻ります。
ステップ 9	**exit** 例： `Device(config-pmap)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 10	**interface** *interface-id* 例： `Device(config)# interface HundredGigabitEthernet 1/0/2`	ポリシーマップを適用するポートを指定し、インターフェイス コンフィギュレーション モードを開始します。 有効なインターフェイスには、物理ポートが含まれます。

	コマンドまたはアクション	目的
ステップ 11	**service-policy input** *policy-map-name* 例： `Device(config-if)# service-policy input flowit`	ポリシーマップ名を指定し、入力ポートに適用します。サポートされるポリシー マップは、入力ポートに 1 つだけです。
ステップ 12	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 13	**show policy-map** [*policy-map-name* [**class** *class-map-name*]] 例： `Device# show policy-map`	（任意）入力を確認します。
ステップ 14	**copy running-config startup-config** 例： `Device# copy-running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

次のタスク

必要に応じて QoS 設定は、ポリシー マップを使用して、SVI のトラフィックの分類、ポリシング、およびマーキングを設定します。

MACsec の暗号化

ここでは、Cisco SM-X-16G4M2X または SM-X-40G8M2X で MACsec 暗号化を設定する方法について説明します。

MACsec 暗号化の前提条件

- Cisco Identity Services Engine（ISE）リリース 2.0 が設定されていることを確認します。
- 802.1x 認証と AAA がデバイスに設定されていることを確認します。

MACsec 暗号化の制約事項

- MACsec 設定は、EtherChannel ポートではサポートされません。
- MACsec 暗号化を設定するには、HSEC ライセンスが必要です。
- MKA 事前共有キーアプローチのみがスイッチ間 MACsec でサポートされます。CTS/SAP（NDAC）および証明書ベースの MKA はサポートされません。
- Extended Packet Numbering（XPN）はサポートされません。
- クリアされた VLAN タグはサポートされません。

MACsec 暗号化について

MACsec 暗号化の推奨事項

ここでは、MACsec 暗号化の設定に関する推奨事項を示します。

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

MACsec 暗号化の概要

MACsec は 2 台の MACsec 対応デバイス間のパケットの認証と暗号化の IEEE 802.1AE 規格です。Cisco SM-X-16G4M2X または SM-X-40G8M2X は、スイッチとホストデバイス間の暗号化に、スイッチからホストへのリンクでの MACsec Key Agreement（MKA）による 802.1AE 暗号化をサポートします。また、スイッチは、MKA ベースのキー交換プロトコルを使用して、スイッチ間（ネットワーク間デバイス）セキュリティの MACsec 暗号化をサポートします。

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

表 *8 :* スイッチ ポートの *MACsec* サポート

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

MKA は、スイッチからホストへのリンクでサポートされます。ホスト側のリンクは、IEEE 802.1x の有無にかかわらず異種デバイスを扱うために、一般に柔軟な認証順序を使用し、オプションで MKA ベースの MACsec 暗号化を使用できます。

Media Access Control Security と MACsec Key Agreement

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

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

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

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

EAPOL プロトコル データ ユニット（PDU）のパケット本体は、MACsec Key Agreement PDU（MKPDU）と呼ばれます。MKA セッションと参加者は、MKA ライフタイム（6 秒間）が経過しても参加者から MKPDU を受信していない場合に削除されます。たとえば、MKA ピアが接続を解除した場合、デバイス上の参加者は MKA ピアから最後の MKPDU を受信した後、6 秒間が経過するまで MKA の動作を継続します。

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

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

MKA ポリシー

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

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

ポリシーマップアクションの定義

ここでは、ポリシーマップアクションとその定義について説明します。

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

残りのアクションについては説明の必要はなく、認証に関連したものです。

仮想ポート

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

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

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

MKA 統計情報

一部の MKA カウンタはグローバルに集約され、その他のカウンタはグローバルとセッション単位の両方で更新されます。

キー ライフタイムおよびヒットレス キー ロールオーバー

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

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

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

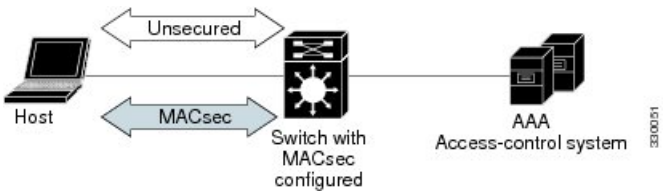
MACsec、MKA、および 802.1x ホスト モード

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

シングルホスト モード

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

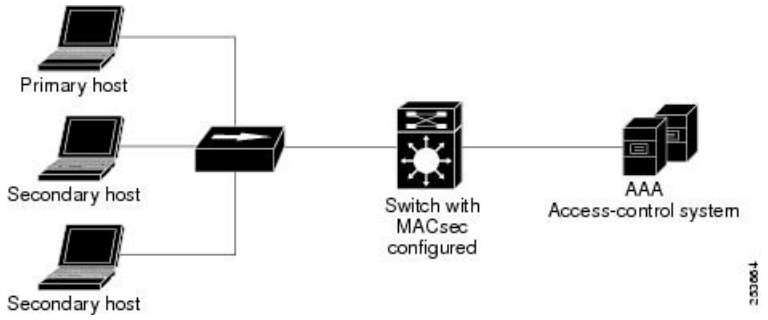
図 *1:* セキュアなデータ セッションでのシングルホスト モードの *MACsec*



マルチ ホスト モード

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

図 *2:* マルチホスト モードの *MACsec*：非セキュア



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

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

マルチドメイン モード

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

ポートチャネルの MKA/MACsec

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

（注） ポートチャネルの一部として形成される EtherChannel リンクは、合同または異種のいずれかです。つまり、リンクは MACsec セキュアまたは非 MACsec セキュアのいずれかになります。ポートチャネルの一方のポートメンバが MACsec に設定されていない場合でも、ポートメンバ間の MKA セッションが確立されます。

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

MACsec 暗号化の設定方法

MKA および MACsec の設定

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

MKA ポリシーの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**mka policy** *policy name* 例： `Device(config)# mka policy mka_policy`	MKA ポリシーを指定して、MKA ポリシー コンフィギュレーション モードを開始します。ポリシー名の長さは最大で 16 文字です。 （注） MKA ポリシー内のデフォルトの MACsec 暗号スイートは常に GCM-AES-128 です。
ステップ 4	**key-server** *priority* 例： `Device(config-mka-policy)# key-server priority 200`	MKA キーサーバオプションを設定し、優先順位を設定します（0 ～ 255 の値）。 （注） キーサーバプライオリティの値を 255 に設定した場合、ピアはキー サーバになることはできません。

	コマンドまたはアクション	目的
ステップ 5	**include-icv-indicator** 例： `Device(config-mka-policy)# include-icv-indicator`	MKPDU の ICV インジケータを有効にします。ICVインジケータを無効にするには、このコマンドの **no** 形式を使用します（**no include-icv-indicator**）。
ステップ 6	**macsec-cipher-suite** *gcm-aes-128* 例： `Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128`	128 ビット暗号により SAK を取得するための暗号スイートを設定します。
ステップ 7	**confidentiality-offset** オフセット値 例： `Device(config-mka-policy)# confidentiality-offset 0`	各物理インターフェイスに機密性（暗号化）オフセットを設定します。 （注） オフセット値は、0、30、または 50 を指定できます。クライアントで Anyconnect を使用している場合は、オフセット 0 を使用することをお勧めします。
ステップ 8	**end** 例： `Device(config-mka-policy)# end`	MKA ポリシーコンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。
ステップ 9	**show mka policy** 例： `Device# show mka policy`	MKA ポリシー設定情報を表示します。

例

次に、MKA ポリシーを設定する例を示します。

```
Switch(config)# mka policy mka_policy
Switch(config-mka-policy)# key-server priority 200
Switch(config-mka-policy)# macsec-cipher-suite gcm-aes-128
Switch(config-mka-policy)# confidentiality-offset 30
Switch(config-mka-policy)# end
```

PSK を使用した MACsec MKA の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**key chain** *key-chain-name* **macsec** 例： `Device(config)# key chain keychain1 macsec`	キー チェーンを設定して、キー チェーン コンフィギュレーション モードを開始します。
ステップ 4	**key** *hex-string* 例： `Device(config-key-chain)# key 1000`	キーチェーン内の各キーの固有識別子を設定し、キーチェーンのキー コンフィギュレーション モードを開始します。 （注） 128 ビット暗号化の場合は、1 ～ 32 文字の 16 進数キー文字列を使用します。256 ビット暗号の場合は、64 文字の 16 進数キー文字列を使用します。
ステップ 5	**key-string** *{ [0|6|7] pwd-string | pwd-string}* 例： `Device(config-key-chain)# key-string 12345678901234567890123456789012`	キー文字列のパスワードを設定します。16 進数の文字のみを入力する必要があります。
ステップ 6	**lifetime local** [*start timestamp {hh::mm::ss | day | month | year}*] [**duration** *seconds | end timestamp {hh::mm::ss | day | month | year}*] 例： `Device(config-key-chain)# lifetime local 12:12:00 July 28 2016 12:19:00 July 28 2016`	事前共有キーの有効期間を設定します。
ステップ 7	**end** 例： `Device(config-key-chain)# end`	キー チェーン コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。

PSK を使用した、インターフェイスでの MACsec MKA の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config-if)# interface GigabitEthernet 1/0/0`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**macsec network-link** 例： `Device(config-if)# macsec network-link`	インターフェイス上で MACsec をイネーブルにします。
ステップ 5	**mka policy** *policy-name* 例： `Device(config-if)# mka policy mka_policy`	MKA ポリシーを設定します。
ステップ 6	**mka pre-shared-key key-chain** *key-chain name* 例： `Device(config-if)# mka pre-shared-key key-chain key-chain-name`	MKA 事前共有キーのキーチェーン名を設定します。 （注） MKA 事前共有キーは、物理インターフェイスまたはサブインターフェイスで設定できますが、両方で設定することはできません。
ステップ 7	**macsec replay-protection window-size** *frame number* 例： `Device(config-if)# macsec replay-protection window-size 10`	リプレイ保護の MACsec ウィンドウ サイズを設定します。
ステップ 8	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

次のタスク

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

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

スイッチとホスト間モードでの MKA MACsec の設定

スイッチとホスト間モードで MKA MACsec を設定するには、次の手順を実行します。

- アイデンティティ制御ポリシーを含む SANet で dot1x を設定します。
- （任意）linksec ポリシーを使用してアイデンティティ制御ポリシーを設定します。
- （任意）MKA ポリシーを設定します。
- インターフェイスに MACsec を適用します。
- （任意）設定された MKA ポリシーをインターフェイスに適用します。
- 設定したアイデンティティ制御ポリシーをインターフェイスに適用します。

802.1x 認証の有効化と AAA の設定

手順

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

	コマンドまたはアクション	目的
ステップ 4	**aaa authentication dot1x default group** *group-name* 例： `Device(config)# aaa authentication dot1x default group macsec-ise`	IEEE 802.1x 用にデフォルトの認証サーバ グループを設定します。
ステップ 5	**aaa authorization network default group** *group-name* 例： `Device(config)# aaa authentication dot1x default group macsec-ise`	ネットワーク認証のデフォルト グループを設定します。
ステップ 6	**dot1x system-auth-control** 例： `Device(config)# dot1x system-auth-control`	デバイス上で 802.1X を有効にします。
ステップ 7	**aaa group server** {**radius** | **tacacs+***group-name* 例： `Device(config)# aaa group server radius macsec-ise`	RADIUS サーバの設定の名前を Protected Access Credential（PAC）のプロビジョニング用に指定し、RADIUS サーバ設定モードを開始します。
ステップ 8	**server** *name* 例： `Device(config)# server name macsec`	サーバの設定の名前を Protected Access Credential（PAC）のプロビジョニング用に指定し、RADIUS サーバ設定モードを開始します。
ステップ 9	**address** *ip-address* **auth-port** *port-number* **acct-port** *port-number* 例： `Device(config)# address ipv4 <ise.ip> auth-port 1812 acct-port 1813`	RADIUS サーバのアカウンティングおよび認証パラメータの IPv4 アドレスを設定します。
ステップ 10	**key** *string* 例： `Device(config)# key cisco123`	デバイスと RADIUS サーバとの間におけるすべての RADIUS 通信用の認証および暗号キーを指定します。
ステップ 11	**policy-map type control subscriber** *control-policy-name* 例： `Device(config)# policy-map type control subscriber cisco-subscriber`	加入者セッションに対して制御ポリシーを定義し、制御ポリシーマップイベントの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 12	**event** *event name* ［**match-all** | **match-first**］ 例： `Device(config-event-control-policymap)# event session-started match-all`	条件が満たされた場合に制御ポリシーのアクションをトリガーするイベントのタイプを指定します。 • **match-all** デフォルトの動作です。 • 使用可能なイベントタイプを表示するには、（**?**）のオンライン ヘルプ機能を使用します。イベントタイプの完全な説明については、**event** コマンドについて参照してください。
ステップ 13	*priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**] 例： `Device(config-class-control-policymap)# 10 class always do-until-failure`	アクションの 1 つが失敗し、制御ポリシーマップアクションのコンフィギュレーションモードが開始されるまで、制御クラスが制御ポリシー内のアクションを指定された順序で実行するように指定します。
ステップ 14	*action-number* **authenticate using** {**dot1x** | **mab** | **webauth**} [**aaa** {**authc-list** *authc-list-name* | **authz-list** *authz-list-name*]} [**merge**] [**parameter-map** *map-name*] [**priority** *priority-number*] [**replace** | **replace-all**] [**retries** *number* {**retry-time** *seconds*}] 例： `Device(config-action-control-policymap)# 10 authenticate using dot1x priority 10`	（任意）指定されたメソッドを使用して加入者セッションの認証を開始します。
ステップ 15	**exit**	グローバル コンフィギュレーション モードに戻ります。
ステップ 16	**interface** {*type* | *slot* | *port*} 例： `Device(config)# interface 1/10`	設定するインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 17	**switchport mode access vlan** *vlan id* 例： `Device(config-if)# switchport access vlan 17`	このアクセスポートでトラフィックを伝送する VLAN を指定します。このコマンドを入力しないと、アクセスポートは VLAN1 だけのトラフィックを伝送します。このコマンドを使用して、アクセスポートがトラフィックを伝送する VLAN を変更できます。

	コマンドまたはアクション	目的
ステップ 18	**switchport mode** {**access** | **trunk**} 例： `Device(config-if)# switchport mode access`	トランキングなし、タグなしの単一 VLAN イーサネット インターフェイスとして、インターフェイスを設定します。アクセス ポートは、1 つの VLAN のトラフィックだけを伝送できます。アクセス ポートは、デフォルトで、VLAN 1 のトラフィックを送受信します。
ステップ 19	**access-session closed** 例： `Device(config-if)# access-session closed`	ポートへのアクセスをクローズすると、クライアントまたはデバイスは、認証が実行される前にネットワーク アクセスを取得できます。
ステップ 20	**access-session port-control** {**auto** | **force-authorized** | **force-unauthorized**} 例： `Device(config-if)# access-session port-control auto`	インターフェイスでのポートベース認証をイネーブルにします。
ステップ 21	**dot1x pae** [**supplicant** | **authenticator**] 例： `Device(config-if)# dot1x pae authenticator`	インターフェイスでのポートベース認証をイネーブルにします。 • **サプリカント**：インターフェイスはサプリカントとしてだけ機能し、オーセンティケータ向けのメッセージに応答しません。 • **オーセンティケータ**：インターフェイスはオーセンティケータとしてだけ動作し、サプリカント向けのメッセージに応答しません。 • **両方**：インターフェイスは、サプリカントおよびオーセンティケータとして動作するため、すべての dot1x メッセージに応答します。
ステップ 22	**policy-map type control subscriber** *control-policy-name* 例： `Device(config)# policy-map type control subscriber cisco-subscriber`	加入者セッションに対して制御ポリシーを定義し、制御ポリシーマップイベントのコンフィギュレーション モードを開始します。
ステップ 23	**exit**	グローバル コンフィギュレーション モードに戻ります。

linksec ポリシーでのアイデンティティ制御ポリシーの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**service-template** *template-name* 例： `Device(config)# service-template dot1x-macsec-policy`	加入者セッションに適用する一連のサービスポリシー属性が含まれるテンプレートを定義して、サービステンプレート コンフィギュレーション モードを開始します。
ステップ 4	**linksec policy** {**must-not-secure** | **must-secure** | **should-secure**} 例： `Device(config-service-template)# linksec policy must-secure`	リンクセキュリティポリシーを must-secure として設定します。 • セキュアMACポリシーは、セキュア MACsec セッションが確立された場合にのみ、eEdgeデバイスポートを許可します。
ステップ 5	**exit** 例： `Device(config-service-template)# exit`	サービス テンプレート コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 6	**policy-map type control subscriber** *control-policy-name* 例： `Device(config)# policy-map type control subscriber cisco-subscriber`	加入者セッションに対して制御ポリシーを定義し、制御ポリシーマップ イベントのコンフィギュレーション モードを開始します。
ステップ 7	**event authentication-success** ［**match-all** | **match-any**］ 例： `Device(config-event-control-policymap)# event authentication-success match-all`	すべての認証イベントがmatchであり、制御ポリシーマップ クラスのコンフィギュレーション モードが開始された場合に、制御ポリシーのアクションをトリガーするイベントのタイプを指定します。

	コマンドまたはアクション	目的
ステップ 8	*priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**] 例： `Device(config-class-control-policymap)# 10 class always do-until-failure`	アクションの 1 つが失敗し、制御ポリシーマップアクションのコンフィギュレーションモードが開始されるまで、制御クラスが制御ポリシー内のアクションを指定された順序で実行するように指定します。
ステップ 9	*action-number* **activate** {**policy type control subscriber** *control-policy-name* | **service-template** *template-name* [**aaa-list** *list-name*] [**precedence** [**replace-all**]} 例： `Device(config-action-control-policymap)# 10 activate service-template dot1x-macsec-policy`	加入者セッションのコントロールポリシーをアクティブ化します。
ステップ 10	**end** 例： `Device(config-action-control-policymap)# end`	制御ポリシーマップアクションのコンフィギュレーションモードを終了し、特権 EXEC モードを開始します。

スイッチ間モードでの MACsec の設定

スイッチ間モードで MACsec を設定するには、次の作業を実行します。

- MACsec 事前共有キーを設定します。
- （任意）MKA ポリシーを設定します。
- インターフェイスに MACsec を適用します。
- （任意）設定された MKA ポリシーをインターフェイスに適用します。
- 設定された MACsec 事前共有キーをインターフェイスに適用します。

MKA 事前共有キーの設定

MACsec Key Agreement（MKA）事前共有キーを設定するには、次のタスクを実行します。

手順

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

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**key chain** *key-chain-name* [**macsec**] 例： `Device(config)# Key chain keychain1 macsec`	キー チェーンを設定して、キー チェーン コンフィギュレーション モードを開始します。
ステップ 4	**key** *hex-string* 例： `Device(config-keychain)# key 9ABCD`	キーを設定して、キー チェーン コンフィギュレーション モードを開始します。 （注） Cisco IOS XE Everest リリース 16.6.1 以降では、接続アソシエーション キー名（CKN）は、このキーの 16 進文字列として設定されている文字列とまったく同じ文字列を使用します。この動作の変更の詳細については、このタスクの後の「MKA-PSK：CKN 動作の変更」セクションを参照してください。
ステップ 5	**cryptographic-algorithm** {**gcm-aes-128** | **gcm-aes-256**} 例： `Device(config-keychain-key)# cryptographic-algorithm gcm-aes-128`	暗号化認証アルゴリズムを設定します。
ステップ 6	**key-string** {[**0** | **6**] *pwd-string* | **7** | *pwd-string*} 例： `Device(config-keychain-key)# key-string 0 pwd`	キー文字列のパスワードを設定します。
ステップ 7	**end** 例： `Device(config-keychain-key)# end`	特権 EXEC モードに戻ります。

MKA の設定

MACsec Key Agreement（MKA）は、キー管理パラメータの設定と制御を可能にします。MKA を設定するには、次のタスクを実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**mka policy** *policy-name* 例： `Device(config)# mka policy MKAPolicy`	MKA ポリシーを設定します。
ステップ 4	**key-server priority** *key-server-priority* 例： `Device(config-mka-policy)# key-server priority 200`	（任意）MKA キー サーバの優先度を設定します。
ステップ 5	**macsec-cipher-suite** {**gcm-aes-128** | **gcm-aes-256** | **gcm-aes-xpn-128** | **gcm-aes-xpn-256**} 例： `Device(config-mka-policy)# macsec-cipher-suite gcm-aes-128 gcm-aes-256`	（任意）セキュア アソシエーション キー（SAK）導出のための暗号スイートを設定します。各暗号スイートの各オプションは1回だけ繰り返すことができますが、任意の順序で使用できます。
ステップ 6	**confidentiality-offset 30** 例： `Device(config-mka-policy)# confidentiality-offset 30`	（任意）MACsec 操作の機密性オフセットを設定します。
ステップ 7	**end** 例：	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config-mka-policy)# end`	（注） MKA ポリシーは、XPN 暗号の機密性オフセットを処理しません。したがって、XPN および非XPN暗号の両方が機密性オフセットとともに MKA ポリシーで設定されている場合、機密性オフセットは XPN 暗号では無視されます。そのため、XPN または非XPN暗号を使用して MKA ポリシーを設定する際は、慎重に判断してください。

インターフェイスでの MACsec および MKA の設定

インターフェイスで MACsec と MKA を設定するには、次のタスクを実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface TenGigabitEthernet 1/0/0`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**switchport mode { access | trunk }** 例： `Device(config-if)# switchport mode trunk }`	トランクに switchport mode を設定します。
ステップ 5	**macsec network-link** 例： `Device(config-if)# mka pre-shared-key key-chain key-chain-name`	ネットワークリンクで MKA MACsec をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 6	**mka policy** *policy-name* 例： `Device(config)# mka policy MKAPolicy`	MKA ポリシーを設定します。
ステップ 7	**mka pre-shared-key key-chain** *key-chain-name* 例： `Device(config)# mka pre-shared-key key-chain k10`	MKA 事前共有キーに keychain10 を設定します。
ステップ 8	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

PSK を使用したポートチャネルの MKA/MACsec の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config-if)# interface gigabitethernet 1/0/3`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**macsec network-link** 例： `Device(config-if)# macsec network-link`	インターフェイス上で MACsec をイネーブルにします。レイヤ 2 およびレイヤ 3 ポートチャネルをサポートします。
ステップ 5	**mka policy** *policy-name* 例： `Device(config-if)# mka policy mka_policy`	MKA ポリシーを設定します。

	コマンドまたはアクション	目的
ステップ 6	**mka pre-shared-key key-chain** *key-chain-name* 例： `Device(config-if)# mka pre-shared-key key-chain key-chain-name`	MKA 事前共有キーのキーチェーン名を設定します。 （注） MKA 事前共有キーは、物理インターフェイスまたはサブインターフェイスで設定できますが、両方で設定することはできません。
ステップ 7	**macsec replay-protection window-size** *frame number* 例： `Device(config-if)# macsec replay-protection window-size 0`	リプレイ保護の MACsec ウィンドウ サイズを設定します。
ステップ 8	**channel-group** *channel-group-number* **mode** {**auto** | **desirable**} | {**active** | **passive**} | {**on**} 例： `Device(config-if)# channel-group 3 mode auto active on`	チャネル グループ内にポートを設定し、モードを設定します。 （注） インターフェイスで MACsec を設定しないと、チャネルグループのポートを設定できません。このステップの前に、ステップ 3、4、5、および 6 のコマンドを設定する必要があります。 channel-number の指定できる範囲は 1 ～ 4096 です。ポートチャネルがない場合は、このチャネルグループに関連付けられたポートチャネルが自動的に作成されます。モードは、以下のキーワードのいずれかを選択します。 • **auto** ：PAgP デバイスが検出された場合に限り、PAgP を有効にします。ポートをパッシブ ネゴシエーション ステートにします。この場合、ポートは受信する PAgP パケットに応答しますが、PAgP パケット ネゴシエーションを開始することはありません。

	コマンドまたはアクション	目的
		（注） EtherChannel メンバが、スイッチスタックにある異なるスイッチのメンバである場合、**auto** キーワードはサポートされません。 • **desirable** ：無条件に PAgP を有効にします。ポートをアクティブ ネゴシエーション ステートにします。この場合、ポートは PAgP パケットを送信することによって、相手ポートとのネゴシエーションを開始します。 （注） EtherChannel メンバが、スイッチスタックにある異なるスイッチのメンバである場合、**desirable** キーワードはサポートされません。 • **on** ：PAgP または LACP を使用せずにポートが強制的にチャネル化されます。on モードでは、EtherChannel が存在するのは、**on** モードのポートグループが、**on**モードの別のポートグループに接続する場合だけです。 • **active** ：LACP デバイスが検出された場合に限り、LACP を有効にします。ポートをアクティブ ネゴシエーション ステートにします。この場合、ポートは LACP パケットを送信することによって、相手ポートとのネゴシエーションを開始します。 • **passive** ：ポート上で LACP を有効にして、ポートをパッシブ ネゴシエーション ステートにします。この場合、ポートは受信する LACP パケットに応答しますが、LACP パケットネゴシエーションを開始することはありません。

	コマンドまたはアクション	目的
ステップ 9	**end** 例： `Device(config-if)# cend`	インターフェイスコンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface port-channel** *channel-group-number* 例： `Device(config)# interface port-channel 1`	ポート チャネル インターフェイスを作成します。 （注） ポート チャネル インターフェイスを削除するには、このコマンドの **no** 形式を使用します。
ステップ 4	**switchport** 例： `Device(config-if)# switchport`	レイヤ 3 モードになっているインターフェイスを、レイヤ 2 設定のレイヤ 2 モードに切り替えます。
ステップ 5	**switchport mode** {**access** | **trunk**} 例： `Device(config-if)# switchport mode access`	すべてのポートをスタティックアクセス ポートとして同じ VLAN に割り当てるか、またはトランクとして設定します。
ステップ 6	**end** 例： `Device(config-if)# end`	インターフェイスコンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

MACsec 暗号化の設定例

例：MKA および MACsec の設定

次に、MKA ポリシーを作成する例を示します。

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

次に、インターフェイスにダウンリンク MACsec を設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# switchport access vlan 17
Device(config-if)# switchport mode access
Device(config-if)# macsec
Device(config-if)# access-session host-mode single-host
Device(config-if)# access-session closed
Device(config-if)# access-session port-control auto
Device(config-if)#mka policy mka_policy
Device(config-if)# dot1x pae authenticator
Device(config-if)#service-policy type control subscriber POLICY_SHOULDSECURE
Device(config-if)#end
```

例：PSK を使用した MACsec MKA の設定

次に、PSK を使用して、MKA MACsec を設定する例を示します。

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

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

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

例：PSK を使用したポートチャネルの MACsec MKA の設定

Etherchannel モード - Static/On

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

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

レイヤ 2 EtherChannel 設定

デバイス 1

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

デバイス 2

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

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

```
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
```

```
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

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

        A - formed by Auto LAG


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

 Group  Port-channel  Protocol    Ports

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

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

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

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

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

        A - formed by Auto LAG


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

 Group  Port-channel  Protocol    Ports

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

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

EtherChannel モード - LACP

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

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

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

レイヤ 2 EtherChannel 設定

デバイス 1

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

デバイス 2

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

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

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

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

        A - formed by Auto LAG
```

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


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

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

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

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

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

        A - formed by Auto LAG


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

Group  Port-channel  Protocol    Ports

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

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

EtherChannel モード - PAgP

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

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

```
Device(config-if)# exit
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# channel-group 2 mode desirable
Device(config-if)# macsec network-link
Device(config-if)# mka policy POLICY
Device(config-if)# mka pre-shared-key key-chain KC
Device(config-if)# end
```

レイヤ 2 EtherChannel 設定

デバイス 1

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

デバイス 2

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

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

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

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

        A - formed by Auto LAG


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


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

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

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

```
 Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
```

```
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

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

        A - formed by Auto LAG


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

 Group  Port-channel  Protocol    Ports

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

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

アクティブな MKA セッションの表示

次に、すべてのアクティブな MKA セッションを表示します。

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

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

例：MKA 情報の表示

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

```
Device# show mka sessions


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

=======================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited
Key-Server
Port-ID        Peer-RxSCI          MACsec-Peers     Status
CKN
```

```
==========================================================================
Gi1/0/1         204c.9e85.ede4/002b p2                NO
YES
43              c800.8459.e764/002a 1                 Secured
01000000000000000000000000000000000000000000000000000000000000000000
```

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

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

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

==========================================================================
Interface       Local-TxSCI          Policy-Name       Inherited
Key-Server
Port-ID         Peer-RxSCI           MACsec-Peers      Status
CKN
==========================================================================
Gi1/0/1         204c.9e85.ede4/002b p2                NO
YES
43              c800.8459.e764/002a 1                 Secured
01000000000000000000000000000000000000000000000000000000000000000000
```

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

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


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

Local Tx-SCI.............. 204c.9e85.ede4/002b
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier...... 43
Interface Name........... GigabitEthernet1/0/1
Audit Session ID.........
CAK Name (CKN)...........
01000000000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)...... 89567
EAP Role................. NA
Key Server............... YES
MKA Cipher Suite......... AES-128-CMAC

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

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

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

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

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

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

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

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

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

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

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

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

```
Device# show mka sessions details

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

Local Tx-SCI............. 204c.9e85.ede4/002b
Interface MAC Address.... 204c.9e85.ede4
MKA Port Identifier...... 43
Interface Name........... GigabitEthernet1/0/1
Audit Session ID.........
CAK Name (CKN)...........
0100000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)... D46CBEC05D5D67594543CEAE
Message Number (MN)...... 89572
```

```
EAP Role.................. NA
Key Server................ YES
MKA Cipher Suite......... AES-128-CMAC

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

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

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

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

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

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

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

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

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

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

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

次に、**show mka policy** コマンドの出力例を示します。

```
Device# show mka policy


MKA Policy Summary...
```

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

p1                1        FALSE   TRUE    0      0      GCM-AES-128

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

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

```
Device# show mka policy p2


MKA Policy Summary...

Policy            KS       Delay   Replay  Window Conf   Cipher
   Interfaces
Name              Priority Protect Protect Size   Offset Suite(s)
   Applied
================================================================================
p2                2        FALSE   TRUE    0      0      GCM-AES-128
   Gi1/0/1
```

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

```
Device# show mka policy p2 detail

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

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

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

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


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

CA Statistics
   Pairwise CAKs Derived... 0
   Pairwise CAK Rekeys..... 0
```

```
   Group CAKs Generated.... 0
   Group CAKs Received..... 0

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

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

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

```
Device# show mka summary

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

================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited
Key-Server
Port-ID        Peer-RxSCI          MACsec-Peers     Status
CKN
================================================================================
Gi1/0/1        204c.9e85.ede4/002b p2               NO
YES
43             c800.8459.e764/002a 1                Secured
01000000000000000000000000000000000000000000000000000000000000000000000000000000



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

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

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

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

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

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

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

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

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

IPv6 ファースト ホップ セキュリティの概要

IPv6 のファースト ホップ セキュリティ（FHS IPv6）は、ポリシーを物理インターフェイス、EtherChannel インターフェイス、または VLAN に適用できる一連の IPv6 セキュリティ機能です。IPv6 ソフトウェア ポリシー データベース サービスは、これらのポリシーを保存しアクセスします。ポリシーを設定または変更すると、ポリシー属性はソフトウェア ポリシー データベースで保存または更新され、その後指定したとおりに適用されます。次の IPv6 ポリシーが現在サポートされています。

- 手動 IPv6 バインディング：セキュアネットワークのスタティック IPv6 バインディングを作成します。
- IPv6 アドレス収集/検査/ガード：NDP および DHCPv6 収集による動的バインディングテーブルの作成を許可します。また、不正なホストによる不正なメッセージを防止するために制御パケットを検査し、不正な RA および DHCP サーバメッセージを保護します。
- IPv6 デバイストラッキング：IPv6 デバイストラッキングを使用して、ネットワーク内のエンドノードの存在、場所、および移動を追跡できます。SISF は、スイッチポートが受信したトラフィックをスヌーピングし、デバイスアイデンティティ（MAC と IP アドレス）を抽出して、バインディングテーブルに保存します。Cisco TrustSec、IEEE 802.1X、LISP、web 認証などの多くの機能は、この情報の正確性に依存して正常に動作します。
- IPv6 FHS バインディングの回復：IPv6 バインディングアドレスの回復により、ルータの完全な障害からバインディングテーブルを回復できます。バインディングテーブルにない不明な送信元からトラフィックを受信した場合、IPv6 FHS バインディング回復機能は、NDP または DHCPv6 リカバリによる IPv6 アドレス収集に基づいたバインディングテーブルの再構築に役立ちます。
- IPv6 ソース ガード：IPv4 ソース ガードと同様、IPv6 ソース ガードは送信元アドレス スプーフィングを防ぐために、送信元アドレスまたはプレフィックスを検証します。

  ソース ガードでは、送信元または宛先アドレスに基づいてトラフィックを許可または拒否するようにハードウェアをプログラムします。ここでは、データ パケットのトラフィックのみを処理します。

  IPv6 ソース ガード機能は、ハードウェア TCAM テーブルにエントリを格納し、ホストが無効な IPv6 送信元アドレスでパケットを送信しないようにします。

  ソースガードパケットをデバッグするには、**debug device-tracking source-guard** 特権 EXEC コマンドを使用します。

> **注** IPv6 ソースガード機能は、入力方向でのみサポートされています。つまり、出力方向ではサポートされていません。IPv6 プレフィックスガードはサポートされていません。

- IPv6 DHCP ガード：IPv6 DHCP ガード機能は、承認されない DHCPv6 サーバおよびリレーエージェントからの返信およびアドバタイズメント メッセージをブロックします。IPv6 DHCP ガードは、偽造されたメッセージがバインディング テーブルに入るのを防ぎ、DHCPv6 サーバまたは DHCP リレーからデータを受信することが明示的に設定されていないポートで受信された DHCPv6 サーバ メッセージをブロックできます。この機能を使用するには、ポリシーを設定してインターフェイスまたは VLAN にアタッチします。
- IPv6 ルータ アドバタイズメント ガード：IPv6 ルータアドバタイズメント（RA）ガード機能を使用すると、ネットワーク管理者は、ネットワーク デバイス プラットフォームに到着した不要または不正な RA ガードメッセージをブロックまたは拒否できます。RA は、リンクで自身をアナウンスするためにデバイスによって使用されます。RA ガード機能は、これらの RA を分析して、未承認のデバイスによって送信された偽の RA をフィルタリングして除外します。ホスト モードでは、ポートではルータ アドバタイズメントとルータ リダイレクト メッセージはすべて許可されません。RA ガード機能は、レイヤ 2 デバイスの設定情報を、受信した RA フレームで検出された情報と比較します。レイヤ 2 デバイスは、RA フレームとルータ リダイレクト フレームの内容を設定と照らし合わせて検証した後で、RA をユニキャストまたはマルチキャストの宛先に転送します。RA フレームの内容が検証されない場合は、RA はドロップされます。

手動 IPv6 バインディングの設定

IPv6 バインディング テーブル コンテンツを設定するには、特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**device-tracking binding vlan** *vlan-id* {*ipv6-address* **interface** interface {*mac_address*} [**tracking**{ [default | disable] [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] | [**enable** [**reachable-lifetimevalue** [*seconds* | **default** | **infinite**] } 例： `Device(config)# decive-tracking binding`	バインディング テーブル データベースにスタティック エントリを追加します。

	コマンドまたはアクション	目的
ステップ 4	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 5	**show device-tracking binding** 例： `Device# show device-tracking binding`	バインディング テーブルの内容を表示します。

IPv6 バインディングリカバリの設定

IPv6 バインディングリカバリを設定するには、特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**device-tracking policy** *policy-name* 例： `Device(config)# device-tracking policy example_policy`	デバイス トラッキング ポリシーを作成し、IPv6デバイストラッキングポリシーコンフィギュレーション モードを開始します。
ステップ 4	**data-glean recovery {dhcp | ndp [dhcp] }** 例： `Device(config-device-tracking)# data-glean recovery dhcp`	データ アドレス グリーニングをイネーブルにし、さまざまな条件に対してメッセージを検証し、メッセージのセキュリティ レベルを指定します。
ステップ 5	**data-glean log-only** 例： `Device(config-device-tracking)# data-glean log-only`	ソース（または「データ」）アドレス グリーニングを使用して、IPv6 ファーストホップ セキュリティ バインディング テーブルのリカバリをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 6	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

IPv6 ネイバー探索インスペクションポリシーの設定

特権 EXEC モードから、IPv6 ND インスペクション ポリシーを設定するには、次の手順に従ってください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**device-tracking policy** *policy-name* 例： `Device(config)# device-tracking policy example_policy`	ポリシーを作成し、デバイストラッキング 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	**security-level inspect** 例： `Device(config-device-tracking)# security-level inspect`	この機能によって適用されるセキュリティのレベルを指定します。
ステップ 5	**device-role** {**host** | **switch**} 例： `Device(config-device-tracking)# device-role switch`	ポートに接続されているデバイスの役割を指定します。デフォルトは **host** です。
ステップ 6	**limit address-count** *value* 例： `Device(config-device-tracking)# limit address-count 1000`	ポートで使用できる IPv6 アドレスの数を制限します。
ステップ 7	**trusted-port** 例：	信頼できるポートにするポートを設定します。

	コマンドまたはアクション	目的
	`Device(config-device-tracking)# trusted-port`	
ステップ 8	**end** 例： `Device(config-device-tracking)# end`	NDインスペクション ポリシー コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 9	**show device-tracking policy** *example_policy* 例： `Device# show device-tracking policy example_policy`	デバイストラッキング インスペクションの設定を確認します。

IPv6 デバイス トラッキング ポリシーの設定

（注） IPv6 スヌーピングポリシー機能は廃止されました。コマンドはCLIに表示され、設定できますが、代わりにスイッチ統合セキュリティ機能（SISF）ベースのデバイストラッキング機能を使用することを推奨します。

デバイス トラッキング ポリシーを設定するには、特権EXECモードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**device-tracking policy** *policy-name* 例： `Device(config)# device-tracking policy example_policy`	デバイス トラッキング ポリシーを作成し、IPv4 または IPv6 デバイストラッキング ポリシー コンフィギュレーション モードを開始します。
ステップ 4	{[**default**] | [**device-role** {**node** | **switch**}] | [**limit address-count** *value*] | [**no**] | [**protocol** {**dhcp** | **dhcp 6** | **arp ndp**}] | [**security-level**	データ アドレス グリーニングをイネーブルにし、さまざまな条件に対してメッ

	コマンドまたはアクション	目的

	{**glean** | **guard** | **inspect**}] | [**tracking** {**disable** [**stale-lifetime** [*seconds* | **infinite**] | **enable** [**reachable-lifetime** [*seconds* | **infinite**] }] | [**trusted-port**] }
	 例：
	 `Device(config-device-tracking)# security-level inspect`
	 例：
	 `Device(config-device-tracking policy)# trusted-port`
		セージを検証し、メッセージのセキュリティ レベルを指定します。
		• （任意）**default**：すべてをデフォルト オプションに設定します。
		• （任意）**device-role**{**node**] | **switch**}：ポートに接続されたデバイスの役割を指定します。デフォルトは **node** です。
		• （任意）**limit address-count** *value*：ターゲットごとに許可されるアドレス数を制限します。
		• （任意）**no**：コマンドを無効にするか、またはそのデフォルトに設定します。
		• （任意）**protocol**{**dhcp** | **ndp**}：分析のために、スヌーピング機能にどのプロトコルをリダイレクトするかを指定します。デフォルトは、**dhcp** および **ndp** です。デフォルトを変更するには、**no protocol** コマンドを使用します。
		• （任意）**security-level**{**glean**|**guard**|**inspect**}：この機能によって適用されるセキュリティのレベルを指定します。デフォルトは **guard** です。
		glean：メッセージからアドレスを収集し、何も確認せずにバインディング テーブルに入力します。
		guard：アドレスを収集し、メッセージを検査します。さらに、RA および DHCP サーバ メッセージを拒否します。これがデフォルトのオプションです。
		inspect：アドレスを収集し、メッセージの一貫性と準拠を検証して、アドレスの所有権を適用します。

	コマンドまたはアクション	目的
		• （任意）**tracking** {**disable** | **enable**}：デフォルトのトラッキング動作を上書きし、トラッキング オプションを指定します。 • （任意）**trusted-port**：信頼できるポートを設定します。これにより、該当するターゲットに対するガードがディセーブルになります。信頼できるポートを経由して学習されたバインディングは、他のどのポートを経由して学習されたバインディングよりも優先されます。テーブル内にエントリを作成しているときに衝突が発生した場合、信頼できるポートが優先されます。
ステップ 5	**end** 例： `Device(config-device-tracking policy)# end`	IPv6 スヌーピング ポリシー コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 6	**show device-tracking policy** *policy-name* 例： `Device#show device-tracking policy example_policy`	デバイストラッキング ポリシーのポリシー設定を表示します。

次のタスク

インターフェイスまたは VLAN に IPv6 デバイストラッキング ポリシーをアタッチします。

IPv6 デバイス トラッキング ポリシーのインターフェイスへの適用

インターフェイスまたはVLANにIPv6デバイス トラッキング ポリシーをアタッチするには、特権 EXEC モードで次の手順を実行します。

手順

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

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface_type stack/module/port* 例： `Device(config)# interface gigabitethernet 1/1/4`	インターフェイスのタイプおよび識別子を指定し、インターフェイス コンフィギュレーション モードを開始します。

ステップ 4	**device-tracking** [**attach-policy** *policy_name* [**vlan** {*vlan_id* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids*}] | **vlan** {*vlan_id* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	例：
	`Device(config-if)# device-traking attach-policy example_policy`
	`Device(config-if)# device-traking vlan 111,112`
	`Device(config-if)# device-traking attach-policy example_policy vlan 111,112`
		インターフェイスまたはそのインターフェイス上の特定の VLAN にカスタム IPv6 スヌーピングポリシーを適用します。デフォルトポリシーをインターフェイスにアタッチするには、**attach-policy** キーワードを指定せずに **device-traking** コマンドを使用します。デフォルトポリシーをインターフェイス上の VLAN にアタッチするには、**device-traking vlan** コマンドを使用します。デフォルト ポリシーは、セキュリティ レベル **guard**、デバイス ロール **node**、プロトコル **ndp** および **dhcp** です。

ステップ 5	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device# show running-config`	インターフェイス コンフィギュレーション モードを終了しないで、ポリシーが特定のインターフェイスにアタッチされていることを確認します。

デバイス トラッキング ポリシーの VLAN へのグローバルな適用

複数のインターフェイスで IPv6 デバイス トラッキング ポリシーを VLAN にアタッチするには、特権 EXEC モードで次の手順を実行します。

手順

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

	コマンドまたはアクション	目的
	`Device> enable`	
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**vlan configuration** *vlan_list* 例： `Device(config)# vlan configuration 333`	IPv6 スヌーピングポリシーを適用する VLAN を指定し、VLAN インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**device-traking** [**attach-policy** *policy_name*] 例： `Device(config-vlan-config)#device-tracking attach-policy example_policy`	すべてのデバイスインターフェイスで、指定した VLAN に IPv6 スヌーピングポリシーを適用します。**attach-policy** オプションを使用しない場合、デフォルト ポリシーがアタッチされます。デフォルト ポリシーは、セキュリティ レベル **guard**、デバイス ロール **node**、プロトコル **ndp** および **dhcp** です。
ステップ 5	**end** 例： `Device(config-vlan-config)# end`	VLAN インターフェイス コンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。

IPv6 ソース ガードの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 source-guard policy** *policy_name* 例： `Device(config)# ipv6 source-guard policy example_policy`	IPv6 ソース ガード ポリシー名を指定し、IPv6 ソース ガード ポリシー コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**validate address** 例： `Device(config-sisf-sourceguard)# validate address`	アドレス検証機能をイネーブルにします。この機能は、検証プレフィックスと検証オプションをサポートしていません。
ステップ 5	**end** 例： `Device(config-sisf-sourceguard)# end`	IPv6 ソース ガード ポリシー コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 6	**show ipv6 source-guard policy** *policy_name* 例： `Device# show ipv6 source-guard policy example_policy`	ポリシー設定と、そのポリシーが適用されるすべてのインターフェイスを表示します。

次のタスク

インターフェイスに IPv6 ソース ガード ポリシーを適用します。

インターフェイスへの IPv6 ソースガードポリシーの適用

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/1/4`	インターフェイスのタイプおよび識別子を指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**ipv6 source-guard**［**attach-policy** <*policy_name*> ］ 例： `Device(config-if)# ipv6 source-guard attach-policy example_policy`	インターフェイスに IPv6 ソース ガード ポリシーをアタッチします。 **attach-policy** オプションを使用しない場合、デフォルト ポリシーがアタッチされます。

	コマンドまたはアクション	目的
ステップ 5	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 6	**show ipv6 source-guard policy** *policy_name* 例： `Device#(config)# show ipv6 source-guard policy example_policy`	ポリシー設定と、そのポリシーが適用されるすべてのインターフェイスを表示します。

IPv6 DHCP ガードポリシーの設定

IPv6 DHCP（DHCPv6）ガード ポリシーを設定するには、特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 dhcp guard policy** *policy-name* 例： `Device(config)# ipv6 dhcp guard policy example_policy`	DHCPv6 ガード ポリシー名を指定し、DHCPv6 ガード ポリシー コンフィギュレーション モードを開始します。
ステップ 4	**device-role** {**client** | **monitor** |**server**} 例： `Device(config-dhcp-guard)# device-role server`	（任意）特定の役割のデバイスからのものではないポート上の DHCPv6 応答および DHCPv6 アドバタイズメントをフィルタします。デフォルトは **client** です。 • **client**：デフォルト値。アタッチされたデバイスがクライアントであることを指定します。サーバメッセージにはこのポートでドロップされます。 • **server**：適用されたデバイスが DHCPv6 サーバであることを指定し

	コマンドまたはアクション	目的
		ます。このポートでは、サーバメッセージが許可されます。
ステップ 5	**trusted-port** 例： `Device(config-dhcp-guard)# trusted-port`	（任意）**trusted-port**：ポートを信頼モードに設定します。このポートでは、これ以上のポリシングは実行されません。 （注） 信頼できるポートを設定した場合、device-role オプションは使用できません。
ステップ 6	**end** 例： `Device(config-dhcp-guard)# end`	DHCPv6 ガード ポリシー コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 7	**show ipv6 dhcp guard policy** *policy_name* 例： `Device# show ipv6 dhcp guard policy example_policy`	（任意）IPv6 DHCP ガードポリシーの設定を表示します。*policy_name* 変数を省略すると、すべての DHCPv6 ポリシーが表示されます。

インターフェイスまたはインターフェイス上の VLAN への IPv6 DHCP ガードポリシーの適用

IPv6 バインディング テーブル コンテンツを設定するには、特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/1/4`	インターフェイスのタイプおよび識別子を指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**ipv6 dhcp guard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids*	DHCP ガード ポリシーをインターフェイスまたはそのインターフェイス上の特定の VLAN にアタッチします。

	コマンドまたはアクション	目的

	| **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	例：
	`Device(config-if)# ipv6 dhcp guard attach-policy example_policy`
	`Device(config-if)# ipv6 dhcp guard attach-policy example_policy vlan 222,223,224`
	`Device(config-if)# ipv6 dhcp guard vlan 222, 223,224`
		attach-policy オプションを使用しない場合、デフォルト ポリシーがアタッチされます。

ステップ 5	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

VLAN への IPv6 DHCP ガードポリシーのグローバル適用

複数のインターフェイス上の VLAN に IPv6 DHCP のガード ポリシーをアタッチするには、特権 EXEC モードで次の手順を実行してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**vlan configuration** *vlan_list* 例： `Device(config)# vlan configuration 334`	IPv6 スヌーピング ポリシーを適用する VLAN を指定し、VLAN インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**ipv6 dhcp guard** [**attach-policy** *policy_name*] 例： `Device(config-vlan-config)#ipv6 dhcp guard attach-policy example_policy`	すべてのスイッチおよびスタック インターフェイスで、IPv6 ネイバー探索ポリシーを指定した VLAN にアタッチします。**attach-policy** オプションを使用しない場合、デフォルト ポリシーがアタッ

	コマンドまたはアクション	目的
		チされます。デフォルト ポリシーは、device-role **client**、**no** trusted-port です。
ステップ 5	**end** 例： `Device(config-vlan-config)# end`	VLAN インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

IPv6 ルータ アドバタイズメント ガード ポリシーの設定

IPv6 ルータ アドバタイズメント ポリシーを設定するには、特権 EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ipv6 nd raguard policy** *policy-name* 例： `Device(config)# ipv6 nd raguard policy example_policy`	RA ガードポリシー名を指定し、RA ガードポリシー コンフィギュレーション モードを開始します。
ステップ 4	[**no**]**device-role** {**host** | **monitor** | **router** | **switch**} 例： `Device(config-nd-raguard)# device-role switch`	ポートに接続されているデバイスの役割を指定します。デフォルトは **host** です。 （注） ホスト側ポートとルータ側ポートの両方を備えたネットワークでは、ホスト側ポートまたは VLAN で **device-role host** を設定した RA ガードポリシーとともに、RA ガード機能が適切に動作できるように、ルータ側のポートで **device-role router** を設定した RA ガードポリシーを設定することが必須です。

	コマンドまたはアクション	目的
ステップ 5	**hop-limit** {**maximum** | **minimum**} *value* 例： `Device(config-nd-raguard)# hop-limit maximum 33`	ホップ制限値によるルータアドバタイズメントメッセージのフィルタリングをイネーブルにします。不正 RA メッセージは低いホップ制限値（IPv4 の Time to Live と同じ）を持つ可能性があるため、ホストによって受け入れられると、ホストが不正 RA メッセージジェネレータを超えて宛先にトラフィックを生成することができなくなります。指定されていないホップ制限値を持つ RA メッセージはブロックされます。 （1 ～ 255）最大および最小のホップ制限値の範囲。 設定されていない場合、このフィルタはディセーブルになります。 「**minimum**」を設定して、指定する値より低いホップ制限値を持つ RA メッセージをブロックします。 「**maximum**」を設定して、指定する値より高いホップ制限値を持つ RA メッセージをブロックします。
ステップ 6	**managed-config-flag** {**off** | **on**} 例： `Device(config-nd-raguard)# managed-config-flag on`	管理アドレス設定（「M」フラグ）フィールドに基づいてルータアドバタイズメントメッセージのフィルタリングを有効にします。「M」フィールドが 1 の不正 RA メッセージの結果としてホストが不正 DHCPv6 サーバを使用する場合があります。設定されていない場合、このフィルタはディセーブルになります。 **On**：「M」値が 1 の RA メッセージを受け入れて転送し、0 のものをブロックします。 **Off**：「M」値が 0 の RA メッセージを受け入れて転送し、1 のものをブロックします。
ステップ 7	**match** { **ipv6 access-list** *list* | **ra prefix-list** *list*} 例：	指定したプレフィックスリストまたはアクセス リストと照合します。

	コマンドまたはアクション	目的
	`Device(config-nd-raguard)# match ipv6 access-list example_list`	
ステップ 8	**router-preference maximum** {**high** | **medium** | **low**} 例： `Device(config-nd-raguard)# router-preference maximum high`	「Router Preference」フラグを使用したルータ アドバタイズメント メッセージのフィルタリングを有効にします。設定されていない場合、このフィルタはディセーブルになります。 • **high**：「Router Preference」が「high」、「medium」、または「low」に設定された RA メッセージを受け入れます。 • **medium**：「Router Preference」が「high」に設定された RA メッセージをブロックします。 • **low**：「Router Preference」が「medium」または「high」に設定された RA メッセージをブロックします。
ステップ 9	**trusted-port** 例： `Device(config-nd-raguard)# trusted-port`	信頼できるポートとして設定すると、すべての接続デバイスが信頼され、より詳細なメッセージ検証は実行されません。
ステップ 10	**end** 例： `Device(config-nd-raguard)# end`	RA ガード ポリシー コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 11	**show ipv6 nd raguard policy** *policy_name* 例： `Device# show ipv6 nd raguard policy example_policy`	（任意）ND ガードポリシーの設定を表示します。

インターフェイスへの IPv6 ルータ アドバタイズメント ガード ポリシーの適用

インターフェイスまたはそのインターフェイス上の VLAN に IPv6 ルータ アドバタイズメント ポリシーをアタッチするには、特権 EXEC モードで次の手順を実行してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/1/4`	インターフェイスのタイプおよび ID を指定し、インターフェイス コンフィギュレーション モードを開始します。

ステップ 4	**ipv6 nd raguard** [**attach-policy** *policy_name* [**vlan** {*vlan_ids* | **add** *vlan_ids* | **except** *vlan_ids* | **none** | **remove** *vlan_ids* | **all**}] | **vlan** [{*vlan_ids* | **add** *vlan_ids* | **except***vlan_ids* | **none** | **remove** *vlan_ids* | **all**}]
	例：
	`Device(config-if)# ipv6 nd raguard attach-policy example_policy`
	`Device(config-if)# ipv6 nd raguard attach-policy example_policy vlan 222,223,224`
	`Device(config-if)# ipv6 nd raguard vlan 222, 223,224`
		ネイバー探索検査ポリシーをインターフェイスまたはそのインターフェイス上の特定の VLAN にアタッチします。**attach-policy** オプションを使用しない場合、デフォルト ポリシーがアタッチされます。

ステップ 5	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

VLAN への IPv6 ルータ アドバタイズメント ガード ポリシーのグローバル適用

インターフェイスに関係なく VLAN に IPv6 ルータ アドバタイズメント ポリシーをアタッチするには、特権 EXEC モードで次の手順を実行してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**vlan configuration** *vlan_list* 例： `Device(config)# vlan configuration 335`	IPv6 RA ガードポリシーを適用する VLAN を指定し、VLAN インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**ipv6 dhcp guard** [**attach-policy** *policy_name*] 例： `Device(config-vlan-config)#ipv6 nd raguard attach-policy example_policy`	すべてのスイッチおよびスタック インターフェイスで、IPv6 RA ガード ポリシーを指定した VLAN にアタッチします。**attach-policy** オプションを使用しない場合、デフォルト ポリシーがアタッチされます。
ステップ 5	**end** 例： `Device(config-vlan-config)# end`	VLAN インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

ダイナミック ARP インスペクションに関する情報

ダイナミック ARP インスペクション（DAI）は、ネットワークのアドレス解決プロトコル（ARP）パケットを確認するセキュリティ機能です。DAI によって、ネットワーク管理者は、無効な MAC アドレスから IP アドレスへのバインディングがある ARP パケットを代行受信、記録、およびドロップすることができます。この機能により、ネットワークをある種の「中間者」攻撃から保護することができます。

前のセクションで説明したような ARP のポイズニング攻撃を防止するには、デバイスは有効な ARP 要求および応答だけがリレーされることを確認する必要があります。DAI は、すべての ARP 要求と応答を代行受信することによってこれらの攻撃を防ぎます。代行受信された各パケットは、ローカル ARP キャッシュが更新される前、またはパケットが適切な宛先に転送される前に、有効な MAC/IP アドレスのバインディングと照合されます。無効な ARP パケットはドロップされます。

DAI は、ARP パケットの有効性を、信頼性のあるデータベースに格納された有効な MAC/IP アドレスのバインディングに基づいて判別します。このデータベースは、DHCP スヌーピングが VLAN および問題のデバイスでイネーブルにされている場合に、DHCP スヌーピングの実行時

に構築されます。さらに、DAI は、静的に設定された IP アドレスを使用するホストを処理するために、ユーザーが設定した ARP ACL と ARP パケットを照合できます。

パケットの IP アドレスが無効である場合、または ARP パケットの本体にある MAC アドレスがイーサネット ヘッダーに指定されたアドレスと一致しない場合に、ARP パケットをドロップするように DAI を設定することもできます。

ダイナミック ARP インスペクションの設定

ダイナミック ARP インスペクションは、不正な IP/MAC アドレス バインディングを持つ ARP パケットを代行受信し、ログに記録して、廃棄します。宛先 MAC アドレス、送信側および宛先の IP アドレス、および送信元 MAC アドレスで追加検証を実行するように、デバイスを設定できます。

ダイナミック ARP インスペクションを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip arp inspection vlan {vlan_ID | vlan_range}** 例： `Device(config)# ip arp inspection vlan 1`	VLAN で DAI をイネーブルにします（デフォルトではディセーブル）。
ステップ 4	**interface** *interface-id* 例： `Device(config)# interface fastEthernet 3/3`	スイッチ B に接続するスイッチ A インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。 信頼できないインターフェイスの場合、デバイスはすべての ARP 要求と応答を代行受信します。ルータは、代行受信した各パケットが、IP アドレスと MAC アドレスとの有効なバインディングを持つことを確認してから、ローカル キャッシュを更新するか、適切な宛

	コマンドまたはアクション	目的
		先にパケットを転送します。デバイスは無効なパケットを廃棄し、**ip arp inspection vlan logging** グローバル コンフィギュレーション コマンドで指定されたログ設定に基づき、ログバッファに廃棄パケットを記録します。
ステップ 5	**ip arp inspection trust** 例： `Device(config-if)# ip arp inspection trust`	スイッチ間の接続を設定します。
ステップ 6	**ip arp inspection filter arp_acl_name vlan {vlan_ID | vlan_range} [static]** 例： `Device(config-if)# ip arp inspection filter test vlan 1`	ARP ACL を VLAN に適用します。 ARP ACL を VLAN に適用します。デフォルトでは、定義済みの ARP ACL は、どのような VLAN にも適用されません。 • arp-acl-name には、ACL の名前を指定します。 • vlan-range では、スイッチとホストが存在する VLAN を指定します。VLAN ID 番号により識別される単一の VLAN、ハイフンで区切られた VLAN 範囲、またはカンマで区切られた一連の VLAN を指定できます。指定できる範囲は 1 ～ 4094 です。 •（任意）**static** を指定すると、ARP ACL 内の暗黙的な拒否が明示的な拒否と見なされ、それ以前に指定された ACL 句に一致しないパケットは廃棄されます。DHCP バインディングは使用されません。このキーワードを指定しない場合は、ACL 内にはパケットを拒否する明示的な拒否が存在しないことになります。この場合は、ACL 句に一致しないパケットを許可するか拒否するかは、DHCP バインディングによって決定されます。

	コマンドまたはアクション	目的
ステップ 7	**ip arp inspection limit {rate pps [burst interval seconds] | none}** 例： `Device(config-if)# ip arp inspection limit rate pps 1`	インターフェイス上の着信 ARP 要求および ARP 応答のレートを制限します。デフォルト レートは、信頼できないインターフェイスでは 15 pps、信頼できるインターフェイスでは無制限です。バースト インターバルは 1 秒です。
ステップ 8	**exit** 例： `Device(config-if)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 9	**errdisable recovery cause arp-inspection**	（任意）ダイナミック ARP インスペクションの errdisable ステートからのエラー回復をイネーブルにし、ダイナミック ARP インスペクションの回復メカニズムで使用する変数を設定します。
ステップ 10	**ip arp inspection validate {[src-mac] [dst-mac] [ip]}** 例： `Device(config)# ip inspection validate ip`	着信 ARP パケットで特定の検査を実行します。デフォルトでは、検証は実行されません。 キーワードの意味は次のとおりです。 • **src-mac** では、イーサネットヘッダーの送信元 MAC アドレスと ARP 本文の送信元 MAC アドレスが比較されます。この検査は、ARP 要求および ARP 応答の両方に対して実行されます。イネーブルにすると、異なる MAC アドレスを持つパケットは無効パケットとして分類され、廃棄されます。 • **dst-mac** では、イーサネットヘッダーの宛先 MAC アドレスと ARP 本文の宛先 MAC アドレスが比較されます。この検査は、ARP 応答に対して実行されます。イネーブルにすると、異なる MAC アドレスを持つパケットは無効パケットとして分類され、廃棄されます。 • **ip** では、ARP 本文から、無効な IP アドレスや予期しない IP アドレスがないかを確認します。アドレス

	コマンドまたはアクション	目的
		には 0.0.0.0、255.255.255.255、およびすべての IP マルチキャスト アドレスが含まれます。送信元 IP アドレスはすべての ARP 要求および ARP 応答内で検査され、宛先 IP アドレスは ARP 応答内だけで検査されます。 少なくとも 1 つのキーワードを指定する必要があります。コマンドを実行するたびに、その前のコマンドの設定は上書きされます。つまり、コマンドが src および dst mac の検証をイネーブルにし、別のコマンドが IP 検証だけをイネーブルにすると、2 番めのコマンドによって src および dst mac の検証がディセーブルになります。
ステップ 11	**ip arp inspection log-buffer entries number**	DAI のログ バッファ サイズを設定します（有効範囲は 0 ～ 1024）。
ステップ 12	**ip arp inspection log-buffer logs number_of_messages interval length_in_seconds**	DAI のログ バッファを設定します。
ステップ 13	**ip arp inspection vlan vlan_range logging {acl-match {matchlog | none} | dhcp-bindings {all | none | permit}}**	各 VLAN に対するログ フィルタリングを設定します。
ステップ 14	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 15	**show ip arp inspection vlan** *vlan-range* 例： `Device# show ip arp insepction vlan 1-2`	選択した範囲の VLAN の統計情報を表示します。

インターフェイス テンプレートに関する情報

インターフェイス テンプレートは、複数のコマンドを同時に設定してターゲット（インターフェイスなど）に関連付けるメカニズムを提供します。インターフェイステンプレートは、特定のポートに適用できる設定またはポリシーのコンテナです。

インターフェイス テンプレートは、ACL を他のコマンドとともに効率的にインターフェイスに適用する方法を実現します。ACL をインターフェイスに適用するには、まずインターフェイス テンプレート内で ACL を設定してから、任意の数のインターフェイスにテンプレートを適用します。ACL が設定された単一のテンプレートは、任意の数の物理インターフェイスまたは仮想インターフェイスに適用できます。

（注） インターフェイス テンプレートは、SVI または EtherChannel ではサポートされていません。

インターフェイス テンプレートの設定

インターフェイス テンプレートを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**template <name>** 例： `Device(config)# template test`	Pls で入力を提供します。
ステップ 4	**ip access-group <acl> in | out** 例： `Device(config-template)# ip access-group <acl> in | out`	指定した IPv6 アクセスリストをテンプレートに適用します。
ステップ 5	**ipv6 traffic-filter <acl> in | out** 例： `Device(config-template)# ip access-group <acl> in | out`	指定した IPv6 アクセス リストを、前のステップで指定したインターフェイスに適用します。
ステップ 6	**source template**テンプレート名 例： `Device(config-if)# source template test`	Pls で入力を提供します。

	コマンドまたはアクション	目的
ステップ 7	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

タイムドメイン反射率計に関する情報

タイムドメイン反射率計を使用して、パルス信号を導体に送信することで導体を分析し、反射された波形の極性、振幅およびラウンドトリップ時間を調べることができます。

特定の伝送メディア内の信号の伝播速度を予測し、その反射が送信元に戻るまでにかかる時間を測定することで、ケーブルテスターから反射ポイントまでの距離を測定することが可能です。また、元のパルスの極性および振幅をその反射率と比較することによって、異なるタイプの障害（たとえば、開いたペアまたは短絡したペア）を区別できます。

タイムドメイン反射率計の設定

インターフェイス テンプレートを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**test cable-diagnostics tdr {interface { Starts the TDR test. interface-number }}** 例： `Device(config)# test cable-diagnostics tdr {interface { Starts the TDR test. interface-number }}`	TDR テストを開始します。
ステップ 4	**show cable-diagnostics tdr {interfaces}** 例： `Device(config)# show cable-diagnostics tdr {interfaces}`	TDR テストのカウンタ情報を表示します。interface-number

	コマンドまたはアクション	目的
ステップ 5	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

Cisco C-SM-16P4M2X、SM-X-16G4M2X、C-SM-40P4M2X、SM-X-40G9M2X サービスモジュールのトラブルシューティング

デバッグログのトラブルシューティングと収集を行うには、次のコマンドを使用します。

- **show platform** コマンドを使用して、モジュールのステータスを確認します。
- 関連する VLAN が作成されたかどうかを確認するには、**show vlan id** <*id_number*> コマンドを使用します。
- ポートがスパニングツリープロトコルによってブロックされていないこと、またはUDLD、ポートセキュリティなどによって error-disabled になっていることを確認します。
- Cisco C-SM-16P4M2X、SM-X-16G4M2X、C-SM-40P4M2X または SM-X-40G9M2X が同じルータに挿入されている場合、Cisco 16 ポートサービスモジュールが優先されます。ルータがリブートし、「レガシースイッチングモード」ではなく「次世代スイッチングモード」で動作します。リロード後は、Cisco 4 ポートおよび 8 ポートが非アクティブになり、Cisco 16 ポートがアクティブになります。

関連資料

関連項目	マニュアル タイトル
Cisco SM-X-16G4M2X サービスモジュール向けのハードウェアの取り付け手順	Cisco SM-X-16G4M2X EtherSwitch サービスモジュールの取り付け
設定に関する一般情報およびコマンドリファレンス	Software Configuration Guide for the Cisco 4000 Integrated Services Router
Cisco 4000 ISR 向けの法規制の遵守に関する情報	Regulatory Compliance and Safety Information for the Cisco 4000 Integrated Services Router
『Software Activation on Cisco Integrated Services Routers and Cisco Integrated Service Routers G2』	Software Activation on Cisco Integrated Services Routers and Cisco Integrated Service Routers G2

表記法

このマニュアルでは、次の表記法を使用しています。

表記法	説明
bold フォント	コマンド、キーワード、およびユーザーが入力するテキストは**bold**で記載されます。
イタリック体	文書のタイトル、新規用語、強調する用語、およびユーザーが値を指定する関数は、イタリック体で示しています。
[ ]	角カッコの中の要素は、省略可能です。
{x | y | z }	どれか 1 つを選択しなければならない必須キーワードは、波カッコで囲み、縦棒で区切って示しています。
[x | y | z]	どれか 1 つを選択できる省略可能なキーワードは、角カッコで囲み、縦棒で区切って示しています。
string	引用符を付けない一組の文字。string の前後には引用符を使用しません。引用符を使用すると、その引用符も含めて string とみなされます。
courier フォント	システムが表示する端末セッションおよび情報は、courier フォントで示しています。
<>	パスワードのように出力されない文字は、山カッコで囲んで示しています。
[ ]	システムプロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。
!、#	コードの先頭に感嘆符（!）またはポンド記号（#）がある場合には、コメント行であることを示します。

マニュアルの入手方法およびテクニカル サポート

マニュアルの入手、Cisco Bug Search Tool（BST）の使用、サービス要求の送信、追加情報の収集の詳細については、『What's New in Cisco Product Documentation』を参照してください。このドキュメントは、http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html から入手できます。

『What's New in Cisco Product Documentation』では、シスコの新規および改訂版の技術マニュアルの一覧を、RSS フィードとして購読できます。また、リーダー アプリケーションを使用して、コンテンツをデスクトップに配信することもできます。RSS フィードは無料のサービスです。