Flexible NetFlow の設定

Flexible NetFlow の前提条件

次に、Flexible NetFlow コンフィギュレーションの前提条件を示します。

- 送信元インターフェイスを設定する必要があります。送信元インターフェイスを設定しなかった場合、エクスポータはディセーブル ステートのままです。
- フロー モニタごとに、有効なレコード名を設定する必要があります。
- IPv6 宛先サーバにフロー レコードをエクスポートするには、IPv6 ルーティングをイネーブルにする必要があります。
- IPFIX 形式の NetFlow レコードをエクスポートするには、フロー エクスポータに IPFIX エクスポート プロトコルを設定する必要があります。
- Flexible NetFlow の key フィールドについて、『Cisco IOS Flexible NetFlow Command Reference』の次のコマンドに定義されている内容をよく理解する必要があります。
  - **match datalink**：データリンク（レイヤ 2）フィールド
  - **match flow**：フィールド識別フロー
  - **match interface**：インターフェイス フィールド
  - **match ipv4**：IPv4 フィールド
  - **match ipv6**：IPv6 フィールド

- **match transport**：トランスポート層フィールド
- **match wireless**：ワイヤレス フィールド
- **match flow cts**：CTS フィールド

- Flexible NetFlow の nonkey フィールドについて、『Cisco IOS Flexible NetFlow Command Reference』で次のコマンドに定義されている内容をよく理解する必要があります。
  - **collect counter**：カウンタ フィールド
  - **collect flow**：フィールド識別フロー
  - **collect interface**：インターフェイス フィールド
  - **collect timestamp**：タイムスタンプ フィールド
  - **collect transport**：トランスポート層フィールド
  - **collect wireless**：ワイヤレス フィールド

IPv4 トラフィック

- ネットワーキング デバイスが IPv4 ルーティング用に設定されていること。
- Cisco Express Forwarding またはdistributed Cisco Express Forwarding のいずれかが、デバイスおよびFlexible NetFlow を有効化するすべてのインターフェイスで有効化されていること。

IPv6 トラフィック

- ネットワーキング デバイスが、IPv6 ルーティング用に設定されていること。
- Cisco Express Forwarding IPv6 または分散型 Cisco Express Forwarding のいずれかが、デバイスおよび Flexible NetFlow を有効化するすべてのインターフェイスで有効化されていること。

Flexible Netflow に関する制約事項

次に、Flexible NetFlow に関する制約事項を示します。

- Flexible NetFlow は、L2 ポートチャネル インターフェイスではサポートされませんが、L2 ポートチャネル メンバー ポートではサポートされます。
- Flexible NetFlow は、L3 ポートチャネル インターフェイスではサポートされませんが、L3 ポートチャネル メンバー ポートではサポートされます。
- Traditional NetFlow（TNF）のアカウンティングはサポートされていません。

- Flexible NetFlow バージョン 9 およびバージョン 10 のエクスポート フォーマットがサポートされています。ただし、エクスポート プロトコルが設定されていない場合は、バージョン 9 のエクスポート フォーマットがデフォルトで適用されます。
- マイクロフロー ポリシング機能は FNF と NetFlow ハードウェア リソースを共有します。
- インターフェイスおよび方向ごとに、1 つのフロー モニタのみサポートされます。
- レイヤ 2、IPv4、および IPv6 のトラフィック タイプがサポートされています。異なるトラフィック タイプの複数のフロー モニタを、指定したインターフェイスと方向に適用できます。同じトラフィック タイプの複数のフロー モニタを指定したインターフェイスと方向には適用できません。
- レイヤ 2、VLAN、WLAN、およびレイヤ 3 インターフェイスがサポートされています。ただし、デバイスは SVI およびトンネルをサポートしていません。
- 次のサイズの NetFlow テーブルがサポートされています。

トリム レベル	入力 NetFlow テーブル	出力 NetFlow テーブル
LAN ベース	サポート対象外	サポート対象外
IP Base	8 K	16 K
IP サービス	8 K	16 K

- スイッチのタイプに応じて、スイッチには 1 個または 2 個の転送 ASIC があります。上記の表に示されているのは、ASIC ごとの容量です。
- スイッチは、1 個または 2 個の ASIC をサポートできます。各 TCAM が最大 6K 入力エントリおよび 12K 出力エントリを処理できる一方、各 ASIC は 8K 入力および 16 K 出力エントリを処理できます。
- NetFlow テーブルは個別のコンパートメントにあり、組み合わせることはできません。パケットを処理した ASIC のテーブルに応じて、対応した ASIC のテーブルにフローが作成されます。
- NetFlow ハードウェアの実装では、4 台のハードウェア サンプラーがサポートされています。1/2 ～ 1/1024 のサンプラー レートを選択できます。ランダム サンプリング モードのみがサポートされています。
- マイクロフロー ポリシング機能（ワイヤレス実装の場合にのみ有効）では、フル フロー モードでのみ NetFlow を使用できます。NetFlow ポリシングは使用できません。マイクロフロー QoS の妨げになるため、ワイヤレス トラフィックにはサンプラーを適用できません。
- ワイヤレス トラフィックでは、フル フロー アカウンティングだけがサポートされています。
- NetFlow ハードウェアの内部では、ハッシュ テーブルが使用されています。ハードウェア内でハッシュ衝突が発生する場合があります。したがって、内部の連想メモリ（CAM）

でオーバーフローが発生しても、実際の NetFlow テーブルの使用率は約 80 % しかない場合があります。

- フローに使用されるフィールドによって異なりますが、単一のフローは2個の連続したエントリを取得できます。IPv6 フローも 2 個のエントリを取得します。この場合、NetFlow エントリを効果的に使用すれば、テーブル サイズの半分で済みます。これは、上記のハッシュ衝突の制限とは別です。
- デバイスは、最大 15 個のフロー モニタをサポートしています。
- SSID ベースの NetFlow アカウンティングがサポートされています。SSID はインターフェイスと同様の方法で扱われます。ただし、ユーザ ID などの一部のフィールドはサポートされていません。
- NetFlow ソフトウェアの実装では、分散 NetFlow エクスポートがサポートされるため、フローが作成された同じデバイスからフローがエクスポートされます。
- 入力フローは最初にフローのパケットを受信した ASIC にあります。出力フローは、パケットが実際に デバイス セットアップを残した ASIC にあります。
- バイト カウント フィールドのレポート値（「bytes long」と呼ばれる）は、レイヤ2パケット サイズの18バイトです。従来のイーサネット トラフィック（802.3）の場合、これは正確です。他のすべてのイーサネット タイプの場合、このフィールドは正確ではありません。「bytes layer2」フィールドを使用すると、常に正確なレイヤ2パケット サイズが報告されます。サポートされる Flexible NetFlow フィールドについては、サポートされている Flexible NetFlow フィールド （18 ページ） を参照してください。
- AVC フロー モニタの IPFIX エクスポータの設定はサポートされていません。
- Flexible NetFlow エクスポートは、イーサネット管理ポート（Gi0/0）ではサポートされていません。
- フロー レコードに送信元グループ タグ（SGT）と宛先グループ タグ（DGT）のフィールド（またはこの2つのいずれかのフィールド）だけが含まれる場合、両方の値を適用できないとしても、SGT と DGT に値ゼロを設定したフローが作成されます。フロー レコードには、SGT および DGT フィールドと一緒に、送信元および宛先 IP アドレスが含まれる必要があります。
- WLAN（SSID）では接続できないCTSフィールドを含むフロー レコードを使用したフロー モニタ。
- QoS のマークが付けられたパケットが出力方向に NetFlow が設定されているインターフェイスで受信されると、パケットの QoS 値がコレクタによってキャプチャされます。しかし、パケットが入力方向に NetFlow が設定されているインターフェイスで受信されると、パケットの QoS 値はコレクタによってキャプチャされません。

Flexible NetFlow に関する情報

Flexible NetFlow の概要

Flexible NetFlow ではフローを使用して、アカウンティング、ネットワーク モニタリング、およびネットワーク プランニングに関連する統計情報を提供します。

フローは送信元インターフェイスに届く単方向のパケット ストリームで、キーの値は同じです。キーは、パケット内のフィールドを識別する値です。フローを作成するには、フロー レコードを使用して、フロー固有のキーを定義します。

デバイス は、ネットワーク異常とセキュリティ問題の高度な検出をイネーブルにする Flexible NetFlow 機能をサポートします。Flexible NetFlow により、大量の定義済みフィールドの集合からキーを選択して、特定のアプリケーションに最適なフロー レコードを定義できます。

1 つのフローと見なされるパケットでは、すべてのキー値が一致している必要があります。フローは、設定したエクスポート レコード バージョンに基づいて、関係のある他のフィールドを集めることもあります。フローはFlexible NetFlow キャッシュに格納されます。

エクスポータを使用してFlexible NetFlowがフローのために収集するデータをエクスポートし、Flexible NetFlow コレクタなどのリモート システムにこのデータをエクスポートできます。Flexible NetFlow コレクタは、IPv4 または IPv6 アドレスを使用できます。

モニタを使用してフローのために収集するデータのサイズを定義します。モニタで、フロー レコードおよびエクスポータを Flexible NetFlow キャッシュ情報と結合します。

ワイヤレス Flexible NetFlow の概要

ワイヤレス Flexible NetFlow インフラストラクチャは次をサポートします。

- Flexible NetFlow バージョン 9.0 および 10
- ユーザ ベースのレート制限
- microflow ポリシング
- 音声およびビデオ フロー モニタリング
- 再帰アクセス コントロール リスト（ACL）

マイクロフロー ポリシングとユーザ ベースのレート制限

マイクロフロー ポリシングは、NetFlowテーブル内の各フローに2カラー、1 レートのポリサーと関連ドロップ統計情報を関連付けます。フロー マスクがすべてのパケット フィールドで構成される場合、この機能は「マイクロフロー ポリシング」と呼ばれます。フロー マスクが送信元または宛先のみで構成される場合、この機能は「ユーザ ベースのレート制限」と呼ばれます。

音声およびビデオ フロー モニタリング

音声およびビデオ フローはフル フロー マスク ベースのエントリです。ASIC は、ポリサー パラメータのプログラム、複数のフローでのポリサー共有、フローの IP アドレスとレイヤー 4 ポート番号の書き換えにおいて柔軟性を提供します。

（注） ダイナミック エントリの場合、NetFlow エンジンは、ポリシー（ACL/QoS ベース ポリシー）に基づいてフローに対して取得されたポリサー パラメータを使用します。ダイナミック エントリは複数のフロー間でポリサーを共有できません。

再帰 ACL

再帰 ACL により、上位層セッション情報に基づいて IP パケットをフィルタリングできます。ACL は発信トラフィックを許可し、信頼ネットワーク内で開始されたセッションに応じて、着信トラフィックを制限します。再帰 ACL は、再帰的なエントリと一致するデータ パケットによりアクティブにされるまで、フィルタリング メカニズムに対して透過的です。この時点では、一時 ACL エントリが作成され、IP 名付きアクセス リストに追加されています。再帰 ACL エントリを生成するデータ パケットから取得した情報は、許可/拒否ビット、送信元 IP アドレス、送信元ポート、宛先 IP アドレス、ポート、およびプロトコル タイプです。再帰 ACL エントリの評価において、プロトコル タイプが TCP または UDP の場合、ポート情報は正確に一致する必要があります。他のプロトコルの場合、一致するポート情報はありません。この ACL をインストールすると、通過する応答パケットに対してファイアウォールが開かれます。この時点では、ハッカーがファイアウォールの背後にあるネットワークにアクセスする危険性があります。この危険性を最小限に抑えるには、アイドル タイムアウト期間を定義できます。ただし、TCP の場合、2 つの FIN ビットまたは RST が検出された場合、ACL エントリが削除される可能性があります。

以前の NetFlow と Flexible NetFlow の利点

以前の NetFlow では、フローの判定に固定の 7 タプルの IP 情報を使用していました。

Flexible NetFlow ではフローをユーザが定義できます。次に、Flexible NetFlow の利点を示します。

- スケーラビリティ、フロー情報の集約などの、大容量フロー認識。
- セキュリティの監視と dDoS の検出および識別のための拡張されたフロー インフラストラクチャ。
- フロー情報をネットワーク内の特定のサービスまたはオペレーションに適応させるパケットからの新しい情報。利用できるフロー情報は、Flexible NetFlow ユーザがカスタマイズ可能。
- Cisco の柔軟で拡張可能な NetFlow Version 9 および Version 10 エクスポート フォーマットの活用。Version 10 エクスポート フォーマットでは、ワイヤレス クライアントの SSID の可変長フィールドをサポート。

- IP アカウンティング、ボーダー ゲートウェイ プロトコル（BGP）ポリシー アカウンティング、永続的キャッシュなどの多数のアカウンティング機能を置換するために使用できる包括的な IP アカウンティング機能。
- NetFlow の入出力アカウンティングのサポート。
- フロー アカウンティングのフル サポートおよびサンプリングした NetFlow アカウンティングのサポート。

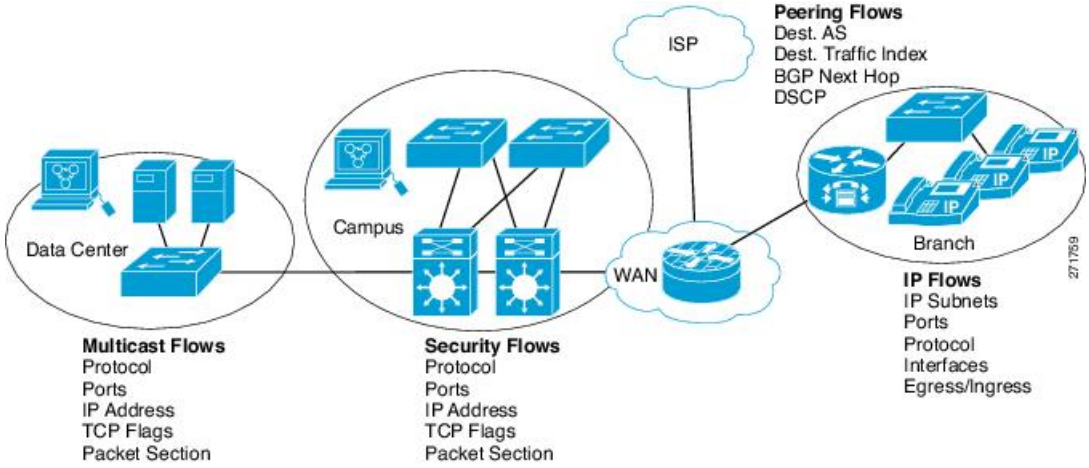
以前の NetFlow では、ネットワーク内のアクティビティを理解して、ネットワーク設計を最適化し、稼働コストを削減できます。

Flexible NetFlow では、ネットワークの動作を、ネットワーク内で使用されるさまざまなサービスに合わせた特定のフロー情報とともに、より効率的に理解できます。次に、Flexible NetFlow 機能用の適用例を示します。

- Flexible NetFlow は Cisco NetFlow をセキュリティ監視ツールとして拡張します。たとえば、ユーザがネットワーク内で特定のタイプの攻撃を検索できるように、パケット長や MAC アドレスのために新しいフロー キーを定義することができます。
- Flexible NetFlow を使用すると、TCP アプリケーションまたは UDP アプリケーションをパケット内のサービス クラス（CoS）ごとに明確に追跡することによって、ホスト間で送信されるアプリケーション トラフィックの量を迅速に識別できます。
- サービス クラスごとに各ネクストホップのマルチプロトコルラベルスイッチング（MPLS）か IP コア ネットワーク、およびその宛先を入力するトラフィックのアカウンティング。この機能では、エッジ間のトラフィック マトリクスを構築できます。

次の表に、Flexible NetFlow をネットワークに導入する方法の例を示します。

図 *1 : Flexible NetFlow* の通常の導入



Flexible NetFlow のコンポーネント

Flexible NetFlow は、いくつかのバリエーションで一緒に使用して、トラフィック分析およびデータ エクスポートに使用できるコンポーネントで構成されます。Flexible NetFlow のユーザ定義のフロー レコードおよびコンポーネントの構造では、最小限の数のコンフィギュレーション コマンドで、ネットワーキング デバイスでのトラフィック分析およびデータ エクスポートのためのさまざまなコンフィギュレーションの作成が容易になります。各フローモニタに、フロー レコード、フロー エクスポータ、およびキャッシュ タイプの固有の組み合わせを設定できます。フロー エクスポータの宛先 IP アドレスなどのパラメータを変更する場合、フロー エクスポータを使用するすべてのフロー モニタに対して自動的に変更されます。同じフロー モニタを複数のフローサンプラと組み合わると、さまざまなインターフェイス上でさまざな速度の同じタイプのネットワーク トラフィックをサンプリングできます。ここでは、Flexible NetFlow コンポーネントのその他の情報を提供します。

フロー レコード

Flexible NetFlow では、キー フィールドと非キー フィールドの組み合わせをレコードと呼びます。Flexible NetFlow のレコードは Flexible NetFlow フロー モニタに割り当てられ、フローデータの格納に使用されるキャッシュが定義されます。Flexible NetFlow には、Flexible NetFlow の使用を開始する際に役立ついくつかの事前定義済みのレコードが含まれています。

フロー レコードでは、フロー内のパケットを識別するために Flexible NetFlow で使用するキーとともに、Flexible NetFlow がフローについて収集する他の関連フィールドを定義します。キーと関連フィールドを任意の組み合わせで指定して、フロー レコードを定義できます。デバイスは、幅広いキー セットをサポートします。フロー レコードでは、フロー単位で収集するカウンタのタイプも定義します。64 ビットのパケットまたはバイト カウンタを設定できます。デバイスは、フロー レコードの作成時に、デフォルトとして次の match フィールドをイネーブルにします。

- match datalink：レイヤ 2 属性
- match flow direction：フローの方向を識別するフィールドとの一致を指定します。
- match interface：インターフェイス属性
- match ipv4：IPv4 属性
- match ipv6：IPv6 属性
- match transport：トランスポート層フィールド
- match wireless：ワイヤレス フィールド
- match flow cts：CTS フィールド

NetFlow の事前定義済みのレコード

Flexible NetFlow には事前定義済みのレコードがいくつか含まれ、それを使用してネットワーク トラフィックの監視を開始できます。事前定義済みのレコードは、Flexible NetFlow を迅速に導入するために役立ち、ユーザ定義のフロー レコードよりも簡単に使用できます。ネット

ワーク モニタリングのニーズを満たす定義済みのレコードのリストから選択できます。Flexible NetFlowが改良されると、一般的なユーザ定義のフロー レコードを事前定義済みレコードとして使用でき、簡単に導入できるようになります。

事前定義済みレコードにより、エクスポートされるデータのために既存のNetFlow コレクタ コンフィギュレーションとの下位互換性が確保されます。事前定義済みレコードは、それぞれ固有の key および nonkey フィールドの組み合わせを持ち、ルータで Flexible NetFlow をカスタマイズしなくても、ネットワーク内のさまざまなタイプのトラフィックを監視する、内蔵機能を提供します。

2 つの事前定義済みレコード（NetFlow original と NetFlow IPv4/IPv6 original output）は機能的に同等で、以前の（入力）NetFlow、および以前の NetFlow の 出力 NetFlow アカウンティング機能をそれぞれエミュレートします。その他の Flexible NetFlow の事前定義済みレコードのいくつかは、以前の NetFlow で利用できる集約キャッシュ方式に基づきます。以前の NetFlow で利用できる集約キャッシュ方式に基づく Flexible NetFlow の事前定義済みレコードでは、集約を実行しません。代わりに、事前定義済みレコードによって各フローが個別に追跡されます。

ユーザ定義レコード

Flexible NetFlow では、key および nonkey フィールドを指定し、実際の要件に合わせてデータ収集をカスタマイズすることで、Flexible NetFlow フロー モニタ キャッシュ用の独自のレコードを定義できます。Flexible NetFlow フロー モニタ キャッシュに対して独自のレコードを定義する場合、ユーザ定義レコードと呼ばれます。nonkeyフィールドの値は、フロー内のトラフィックに関する追加情報を提供するためにフローに追加されます。nonkeyフィールドの値の変更によって新しいフローが作成されることはありません。ほとんどの場合、nonkeyフィールドの値はフロー内の最初のパケットからのみ取得されます。Flexible NetFlow を使用すると、nonkeyフィールドとして、フロー内のバイト数やパケット数などのカウンター値をキャプチャできます。

ユーザ定義レコードは、QoS および帯域幅監視、アプリケーションとユーザのトラフィック プロファイリング、dDoS 攻撃に対するセキュリティ監視などのアプリケーション用に作成できます。また、Flexible NetFlow には以前の NetFlow をエミュレートするいくつかの事前定義済みレコードも含まれています。Flexible NetFlow のユーザ定義レコードでは、ユーザが設定可能なサイズのパケットの連続するセクションを監視する機能を利用でき、key フィールドまたは nonkey フィールドとしてパケットのその他のフィールドや属性とともにフロー レコード内で使用します。セクションにはパケットのレイヤ 3 データが含まれる場合があります。パケットのセクション フィールドでは、ユーザが Flexible NetFlow の事前定義済みレコードの対象外のパケット フィールドを監視できます。事前定義済みキーで収集されないパケット フィールドの分析機能によって、さらに詳細なトラフィック モニタリングが可能になるため、dDoS 攻撃の調査に役立ち、URL モニタリングなど他のセキュリティ アプリケーションの実装が可能になります。

Flexible NetFlow では、事前定義済みタイプのユーザが設定可能なサイズのパケット セクションが提供されます。次の Flexible NetFlow コマンド（Flexible NetFlow フロー レコード コンフィギュレーション モードで使用される）をパケット セクションの事前定義済みタイプの設定に使用できます。

- **collectipv4sectionheadersize** *bytes*：各パケットの IPv4 ヘッダーの先頭から *bytes* 引数で指定されたバイト数のキャプチャを開始します。

- **collectipv4sectionpayloadsize** *bytes*：各パケットの IPv4 ヘッダーの直後からバイトのキャプチャを開始します。キャプチャされるバイト数は *bytes* 引数で指定されます。
- **collectipv6sectionheadersize** *bytes*：各パケットの IPv6 ヘッダーの先頭から *bytes* 引数で指定されたバイト数のキャプチャを開始します。
- **collectipv6sectionpayloadsize** *bytes*：各パケットの IPv6 ヘッダーの直後からバイトのキャプチャを開始します。キャプチャされるバイト数は *bytes* 引数で指定されます。

bytes 値は、フロー レコードのこれらのフィールドのサイズ（バイト単位）です。パケットの対応フラグメントが要求されたセクションサイズよりも小さい場合、Flexible NetFlow はフローレコード内の残りのセクション フィールドを 0 で埋めます。パケット タイプが要求されたセクション タイプと一致しなかった場合、Flexible NetFlow はフロー レコード内のセクション フィールド全体を 0 で埋めます。

Flexible NetFlow では、ヘッダーおよびパケット セクションのタイプに新しいバージョン 9 エクスポート フォーマット フィールド タイプが追加されます。Flexible NetFlow は NetFlow コレクタに、対応するバージョン 9 エクスポート テンプレート フィールドで設定されたセクション サイズを通知します。ペイロード セクションには、対応する長さフィールドがあり、収集されるセクションの実際のサイズを収集するために使用できます。

Flexible NetFlow の match パラメータ

次の表で、Flexible NetFlow の match パラメータについて説明します。フロー レコードごとに、次の match パラメータを 1 つ以上設定する必要があります。

表 *1 : match* パラメータ

コマンド	目的
match datalink {**dot1q** | **ethertype** | **mac** | **vlan** }	データ リンクまたはレイヤ 2 フィールドとの一致を指定します。次のコマンド オプションが使用可能です。 • **dot1q**：dot1q フィールドと一致します。 • **ethertype**：パケットの ethertype と一致します。 • **mac**：送信元または宛先の MAC フィールドと一致します。 • **vlan**：パケットが配置される VLAN と一致します（入力または出力）。
match flow direction	フローを識別するフィールドとの一致を指定します。

コマンド	目的
match interface {**input** | **output**}	インターフェイス フィールドとの一致を指定します。次のコマンド オプションが使用可能です。 • **input**：入力インターフェイスと一致します。 • **output**：出力インターフェイスと一致します。
match ipv4 {**destination** | **protocol** | **source** | **tos** | **ttl** | **version**}	IPv4 フィールドとの一致を指定します。次のコマンド オプションが使用可能です。 • **destination**：IPv4 宛先アドレス ベースのフィールドと一致します。 • **protocol**：IPv4 プロトコルと一致します。 • **source**：IPv4 送信元アドレス ベースのフィールドと一致します。 • **tos**：IPv4 タイプ オブ サービス フィールドと一致します。 • **ttl**：IPv4 存続時間フィールドと一致します。 • **version**：IPv4 ヘッダーの IP バージョンと一致します。

コマンド	目的
match ipv6 {**destination** | **hop-limit** | **protocol** | **source** | **traffic-class** | **version** }	IPv6 フィールドとの一致を指定します。次のコマンド オプションが使用可能です。 • **destination**：IPv6 宛先アドレス ベースのフィールドと一致します。 • **hop-limit**：IPv6 ホップ リミット フィールドと一致します。 • **protocol**：IPv6 ペイロード プロトコル フィールドと一致します。 • **source**：IPv6 送信元アドレス ベースのフィールドと一致します。 • **traffic-class**：IPv6 トラフィック クラスと一致します。 • **version**：IPv6 ヘッダーの IP バージョンと一致します。
match transport {**destination-port** | **igmp** | **icmp** | **source-port**}	トランスポート層フィールドとの一致を指定します。次のコマンド オプションが使用可能です。 • **destination-port**：転送先ポートと一致します。 • **icmp**：ICMP IPv4 および IPv6 フィールドを含む ICMP フィールドと一致します。 • **igmp**：IGMP フィールドと一致します。 • **source-port**：転送元ポートと一致します。
match flow cts {source | destination} group-tag	FNF レコードの CTS フィールドのサポートとの一致を指定します。次のコマンド オプションが使用可能です。 • **source**：ドメインを入力する CTS の送信元と一致します。 • **destination**：ドメインを脱退する CTS の宛先と一致します。

Flexible NetFlow の collect パラメータ

次の表で、Flexible NetFlow の collect パラメータについて説明します。

表 *2 : collect* パラメータ

コマンド	目的
collect counter { **bytes** { **layer2** { **long** } | **long** } | **packets** { **long** } }	カウンタ フィールドの合計バイト数と合計パケット数を収集します。
collect interface {**input** | **output**}	入力または出力インターフェイスからフィールドを収集します。
collect timestamp absolute {**first** | **last**}	最初のパケットが確認された絶対時間、または最新のパケットが最後に確認された絶対時間のフィールドを収集します（ミリ秒）。
collect transport tcp flags	次の転送 TCP フラグを収集します。 • **ack**：TCP 確認応答フラグ • **cwr**：TCP 輻輳ウィンドウ縮小フラグ • **ece**：TCP ECN エコー フラグ • **fin**：TCP 終了フラグ • **psh**：TCP プッシュ フラグ • **rst**：TCP リセット フラグ • **syn**：TCP 同期フラグ • **urg**：TCP 緊急フラグ （注） デバイスでは、収集する TCP フラグを指定できません。転送 TCP フラグの収集のみ指定できます。すべての TCP フラグはこのコマンドで収集されます。

フロー エクスポータ

フロー エクスポータでは、フロー モニタ キャッシュ内のデータをリモート システム（たとえば、分析および保管のために NetFlow コレクタを実行するサーバ）にエクスポートします。フロー エクスポータは、コンフィギュレーションで別のエンティティとして作成されます。フロー エクスポータは、フロー モニタにデータ エクスポート機能を提供するためにフロー モニタに割り当てられます。複数のフロー エクスポータを作成して、1 つまたは複数のフロー モニタに適用すると、いくつかのエクスポート先を指定することができます。1 つのフロー エクスポータを作成し、いくつかのフロー モニタに適用することができます。

NetFlow データ エクスポート フォーマット バージョン 10（IPFIX）

Internet Protocol Flow Information Export（IPFIX）、つまりバージョン 10 は、事前に定義されたか、またはユーザ定義のフロー レコードを収集し、エクスポートするエクスポート プロトコルです。IPFIX は NetFlow バージョン 9に基づいた IETF 標準です。IPFIX 形式は NetFlow バージョン9として、個別のテンプレートとレコードについて同じ原則を保ちます。これにより、ワイヤレス クライアントの SSID の可変長フィールドをサポートします。IPFIX エクスポート プロトコルでは、デフォルトの宛先ポートは 4739、DSCP 値は 0、TTL は 255 です。

NetFlow データ エクスポート フォーマットのバージョン 9

NetFlow の基本出力はフロー レコードです。NetFlow が改良され、フロー レコードのいくつかのフォーマットが向上しました。NetFlow エクスポート フォーマットの最新の進化は、バージョン 9 と呼ばれます。NetFlow Version 9 エクスポート フォーマットの識別機能は、テンプレートがベースとなります。テンプレートは、レコードフォーマットの設計を拡張可能なものにします。NetFlowサービスが将来拡張されても、基本フロー レコードフォーマットを変更し続ける必要がありません。テンプレートを使用すると、次のいくつかの利点があります。

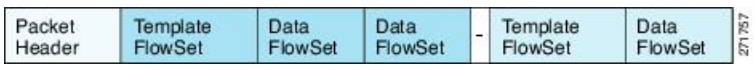
- NetFlow のコレクタを提供したり、サービスを表示したりするアプリケーションを作成するサードパーティ ビジネス パートナーは、新規の NetFlow 機能が追加されるたびにアプリケーションを再コンパイルする必要はありません。代わりに、既知のテンプレート フォーマットを記述する外部のデータ ファイルを使用することができます。
- 新規機能は、現在の導入環境を損ねることなく、NetFlow に迅速に追加できます。
- バージョン9フォーマットは新しいプロトコルや開発中のプロトコルに適応できるため、NetFlow はこれらのプロトコルに対して「将来的に対応」します。

NetFlow バージョン 9 エクスポート フォーマットは、次の特徴と機能を提供します。

- 可変フィールド仕様フォーマット
- IPv4 または IPv6 の宛先アドレスのエクスポートのサポート
- ネットワークをより効率的に利用可能

バージョン 9 のエクスポート フォーマットは、パケット ヘッダーとそれに続く 1 つ以上のテンプレート フロー セットまたはデータ フロー セットで構成されています。テンプレート フローセットでは、将来のデータ フローセットに表示されるフィールドの説明が提供されます。このようなデータ フロー セットは、後で同じエクスポート パケットまたは後続のエクスポート パケットで発生する可能性があります。テンプレート フロー セットおよびデータ フローセットは、次の図に示すように、単一のエクスポートパケットに混在させることができます。

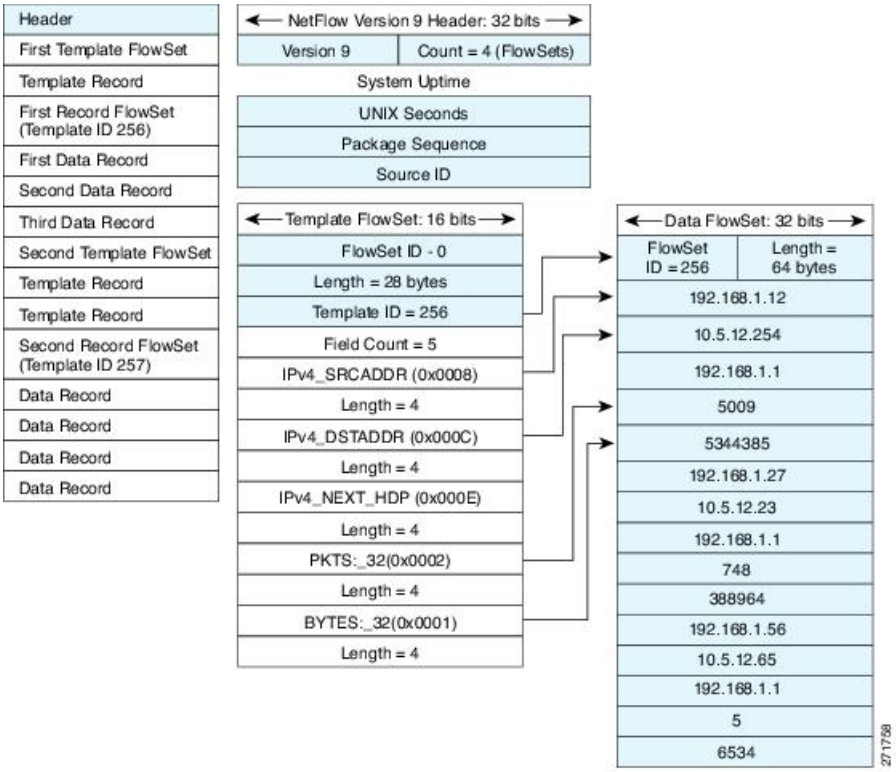
図 *2:* バージョン *9* エクスポート パケット



NetFlow Version 9 では、送信されるデータを NetFlow コレクタが理解できるように、テンプレート データを定期的にエクスポートします。また、テンプレートのデータ フロー セットもエクスポートします。Flexible NetFlow の主な利点は、ユーザがフロー レコードを設定すると、

バージョン9テンプレートに効率的に変換され、コレクタに転送されることです。下の図に、ヘッダー、テンプレート フロー セットおよびデータ フロー セットを含めて、NetFlow Version 9 エクスポート フォーマットの詳細な例を示します。

図 *3 : NetFlow* バージョン *9* エクスポート フォーマットの詳細例



バージョン 9 エクスポート フォーマットの詳細については、ホワイト ペーパー『Cisco IOS NetFlow Version 9 Flow-Record Format』を参照してください。次の URL から入手できます。http://www.cisco.com/en/US/tech/tk648/tk362/technologies_white_paper09186a00800a3db9.shtml

フロー モニタ

フロー モニタは Flexible NetFlow のネットワーク トラフィックの監視を実行するコンポーネントで、インターフェイスに適用されます。

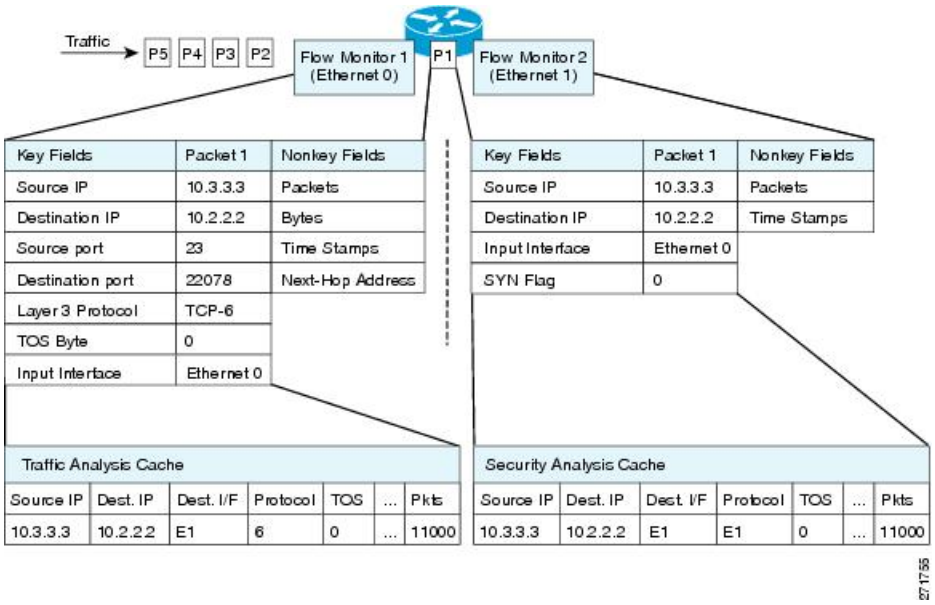
フロー モニタは、ユーザ定義のレコード、オプションのフロー エクスポータ、およびフロー モニタが最初のインターフェイスに適用されるときに自動的に作成されるキャッシュで構成されます。

フロー データはネットワーク トラフィックから収集され、フロー レコードの key フィールドおよび nonkey フィールドに基づいて監視プロセス中にフロー モニタ キャッシュに追加されます。

Flexible NetFlow は、同じトラフィックのさまざまなタイプの分析を実行するために使用できます。下の図では、入力インターフェイス上の標準トラフィック分析のために設計されたレ

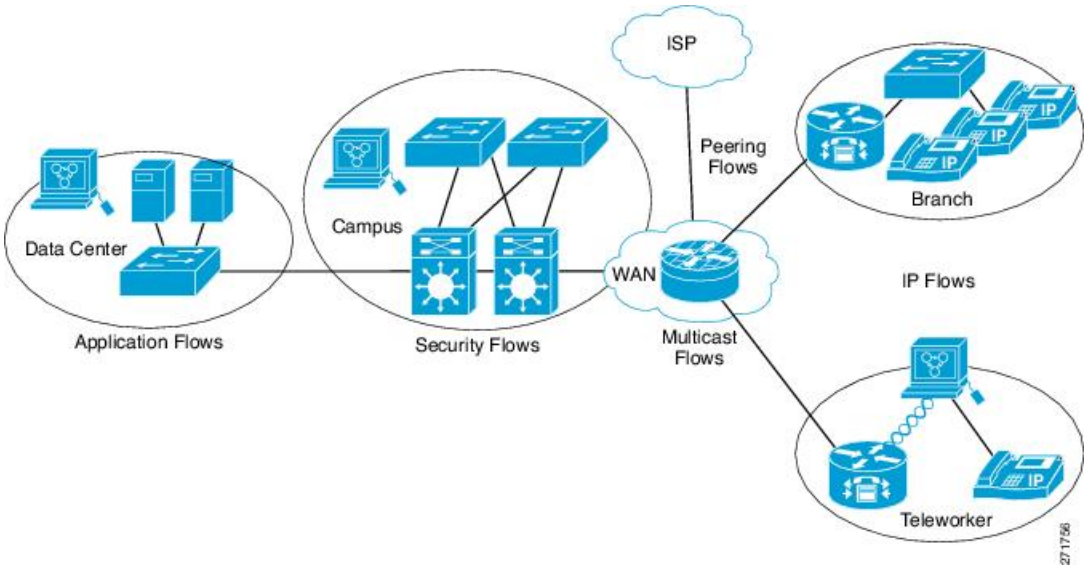
コードと、出力インターフェイス上のセキュリティ分析のために設計されたレコードを使用してパケット 1 が分析されます。

図 *4:2* つのフロー モニタを使用した同じトラフィックの分析例



下の図に、カスタム レコードを使用して複数のタイプのフロー モニタを適用するより複雑な方法の例を示します。

図 *5:* カスタム レコードでの複数のタイプのフロー モニタの複雑な使用例



3 つのタイプのフロー モニタ キャッシュがあります。フロー モニタの作成後に、そのフロー モニタで使用するキャッシュ タイプを変更します。3 タイプのフロー モニタ キャッシュについては、次の各項に説明があります。

標準

デフォルトのキャッシュ タイプは「normal」です。このモードでは、キャッシュ内のエントリが timeout active 設定と timeout inactive 設定に従って期限切れになります。キャッシュ エントリは、期限切れになるとキャッシュから削除され、設定されている何らかのエクスポータによってエクスポートされます。

即時

「immediate」タイプのキャッシュは、作成されるとすぐにレコードを期限切れにします。その結果、どのフローにも 1 パケットしか含まれません。キャッシュ内容を表示するコマンドでは、パケットの履歴が表示されます。

予想されるフローが非常に少なく、パケットが検出されてからレポートがエクスポートされるまでの遅延を最小限にする場合は、このモードが適しています。

注意 このモードでは大量のエクスポートデータが生じて、低速のリンクが過負荷状態になり、エクスポート先のシステムに著しく影響する可能性があります。処理するパケット数を削減するようにサンプリングを設定することをお勧めします。

（注） キャッシュ タイムアウト設定は、このモードでは何の効果もありません。

Permanent

タイプが「permanent」のキャッシュでは、フローが期限切れになることはありません。permanent キャッシュは、検出が予想されるフローの数が少なく、ルータに長期間の統計情報を保存する必要がある場合に便利です。たとえば、フロー レコード内の key フィールドが 8 ビット IP ToS フィールドだけで、256 フローだけを監視する場合があります。ネットワーク トラフィックの IP ToS フィールドの使用状況を長期間に渡って監視するには、permanent キャッシュを使用します。permanent キャッシュは、課金アプリケーション、および追跡対象が固定セットのフローに対する、全域におよぶトラフィック マトリクスに役立ちます。アップデート メッセージは、「timeout update」設定に従って設定されたすべてのフロー エクスポータに、定期的に送信されます。

（注） permanent モードでキャッシュがいっぱいになった場合は、新しいフローが監視されなくなります。そうなった場合は、キャッシュの統計情報に「Flows not added」というメッセージが示されます。

（注） permanent キャッシュでは、デルタ カウンタではなくアップデート カウンタが使用されます。そのため、フローがエクスポートされると、カウンタにはフローのライフタイム全体の総検出数が示され、最後のエクスポート送信後に検出された追加パケットは示されません。

フロー サンプラー

フローサンプラーは、ルータのコンフィギュレーションで別のコンポーネントとして作成されます。フロー サンプラーは、分析用に選択されるパケットの数を制限することで、 Flexible NetFlow を実行しているデバイス上の負荷を減らすために使用されます。

サンプラーはランダムサンプリング技術（モード）を使用します。つまり、サンプルを取得するときに、ランダムに選択したサンプリング位置が毎回使用されます。

フロー サンプリングでは、ルータのパフォーマンスに対するモニタリング精度が交換されます。サンプラーをフロー モニタに適用すると、フロー モニタが分析する必要のあるパケット数が減少するため、ルータでフロー モニタを実行するためのオーバーヘッド負荷が低下します。フロー モニタで分析されるパケット数が減少すると、フロー モニタのキャッシュに格納される情報の精度が、それに応じて低下します。

ip flow monitor コマンドを使用してインターフェイスに適用する場合、サンプラーとフロー モニタを組み合わせます。

サポートされている Flexible NetFlow フィールド

次の表では、さまざまなトラフィック タイプおよびトラフィック方向について、Flexible NetFlow（FNF）でサポートされるフィールドの統合リストを提供しています。

（注） パケットに VLAN フィールドがある場合、その長さは考慮されません。

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注意
Key または Collect フィールド							

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注意
インターフェイス入力	Yes	—	Yes	—	Yes	—	フロー モニタを入力方向に適用する場合： • **match** キーワードを使用し、入力インターフェイスを key フィールドとして使用します。 • **collect** キーワードを使用し、出力インターフェイスを collect フィールドとして使用します。このフィールドはエクスポートされるレコードに含まれますが、値は 0 になります。
インターフェイス出力	—	Yes	—	Yes	—	Yes	フロー モニタを出力方向に適用する場合： • **match** キーワードを使用し、出力インターフェイスを key フィールドとして使用します。 • **collect** キーワードを使用し、入力インターフェイスを collect フィールドとして使用します。このフィールドはエクスポートされるレコードに含まれますが、値は 0 になります。

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注意
Key フィールド							

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注意
フロー方向	Yes	Yes	Yes	Yes	Yes	Yes	
Ethertype	Yes	Yes	—	—	—	—	
VLAN 入力	Yes	—	Yes	—	Yes	—	スイッチポートでのみサポートされています。
VLAN 出力	—	Yes	—	Yes	—	Yes	スイッチポートでのみサポートされています。
dot1q VLAN 入力	Yes	—	Yes	—	Yes	—	スイッチポートでのみサポートされています。
dot1q VLAN 出力	—	Yes	—	Yes	—	Yes	スイッチポートでのみサポートされています。
dot1q 優先度	Yes	Yes	Yes	Yes	Yes	Yes	スイッチポートでのみサポートされています。
MAC 送信元アドレス入力	Yes	Yes	Yes	Yes	Yes	Yes	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注意
MAC 送信元アドレス出力	—	—	—	—	—	—	
MAC 宛先アドレス入力	Yes	—	Yes	—	Yes	—	
MAC 送信先アドレス出力	—	Yes	—	Yes	—	Yes	
IPv4 バージョン	—	—	Yes	Yes	Yes	Yes	
IPv4 TOS	—	—	Yes	Yes	Yes	Yes	
IPv4 プロトコル	—	—	Yes	Yes	Yes	Yes	送信元/宛先ポート、ICMP コード/タイプ、IGMP タイプ、TCP フラグのいずれかが使用されている場合に使用する必要があります。
IPv4 TTL	—	—	Yes	Yes	Yes	Yes	
IPv4 発信元アドレス	—	—	Yes	Yes	—	—	
IPv4 宛先アドレス	—	—	Yes	Yes	—	—	
ICMP IPv4 タイプ	—	—	Yes	Yes	—	—	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注意
ICMP IPv4 コード	—	—	Yes	Yes	—	—	
IGMP タイプ	—	—	Yes	Yes	—	—	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注意
Key フィールド（続き）							
IPv6 バージョン	—	—	Yes	Yes	Yes	Yes	IP バージョンと同じです。
IPv6 プロトコル	—	—	Yes	Yes	Yes	Yes	IP プロトコルと同じです。送信元/宛先ポート、ICMP コード/タイプ、IGMP タイプ、TCP フラグのいずれかが使用されている場合に使用する必要があります。
IPv6 送信元アドレス	—	—	—	—	Yes	Yes	
IPv6 宛先アドレス	—	—	—	—	Yes	Yes	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注意
IPv6 トラフィッククラス	—	—	Yes	Yes	Yes	Yes	IP TOS と同じです。
IPv6 ホップ リミット	—	—	Yes	Yes	Yes	Yes	IP TTL と同じです。
ICMP IPv6 タイプ	—	—	—	—	Yes	Yes	
ICMP IPv6 コード	—	—	—	—	Yes	Yes	
source-port	—	—	Yes	Yes	Yes	Yes	
dest-port	—	—	Yes	Yes	Yes	Yes	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注意
Collect フィールド							
Bytes long	Yes	Yes	Yes	Yes	Yes	Yes	パケットサイズ =（FCS を含むイーサネットフレームサイズ - 18 バイト） **推奨**： このフィールドを回避し、Bytes layer2 long を使用します。
Packets long	Yes	Yes	Yes	Yes	Yes	Yes	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注意
Timestamp absolute first	Yes	Yes	Yes	Yes	Yes	Yes	
Timestamp absolute last	Yes	Yes	Yes	Yes	Yes	Yes	
TCP フラグ	Yes	Yes	Yes	Yes	Yes	Yes	すべてのフラグを収集します。
Bytes layer2 long	Yes	Yes	Yes	Yes	Yes	Yes	

デフォルト設定

次の表は、デバイスに対する Flexible NetFlow のデフォルト設定を示します。

表 *3 :* デフォルトの *Flexible NetFlow* 設定

設定	デフォルト
フロー アクティブ タイムアウト	1800 秒
フロー タイムアウトの非アクティブ化	15 秒

Flexible NetFlow の設定方法

Flexible NetFlow を設定するには、次の一般的な手順に従います。

1. フローにキー フィールドおよび非キー フィールドを指定して、フロー レコードを作成します。
2. プロトコルを指定して任意のフローエクスポータを作成し、宛先ポート、宛先、およびその他のパラメータを転送します。
3. フロー レコードおよびフロー エクスポータに基づいて、フロー モニタを作成します。
4. 任意のサンプラーを作成します。
5. レイヤ 2 ポート、レイヤ 3 ポート、または VLAN にフロー モニタを適用します。
6. 必要に応じ、WLAN を設定してフロー モニタを適用します。

カスタマイズしたフロー レコードの設定

カスタマイズしたフロー レコードを設定するには、次のタスクを実行します。

カスタマイズしたフロー レコードは、特定の目的でトラフィック データを分析するために使用します。カスタマイズしたフロー レコードには、key フィールドとして使用する **match** 基準が 1 つ以上必要です。通常は nonkey フィールドとして使用する **collect** 基準が 1 つ以上あります。

カスタマイズしたフロー レコードの順列は、数百もの可能性があります。このタスクでは、可能性のある順列の 1 つを作成するための手順について説明します。必要に応じて当該タスクの手順を変更し、要件に合わせてカスタマイズしたフロー レコードを作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** **例**： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** **例**： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**flowrecord** *record-name* **例**： `Device(config)# flow record FLOW-RECORD-1`	フロー レコードを作成し、Flexible NetFlow フロー レコード コンフィギュレーション モードを開始します。 • このコマンドでは、既存のフロー レコードを変更することもできます。
ステップ 4	**description** 説明 **例**： `Device(config-flow-record)# description Used for basic traffic analysis`	**ipv4**（任意）フロー レコードの説明を作成します。
ステップ 5	**match** { | **ipv6**} {**destination** | **source**} **address** **例**：	フロー レコードの key フィールドを設定します。

	コマンドまたはアクション	目的
	`Device(config-flow-record)# match ipv4 destination address`	（注） この例では、IPv4宛先アドレスをレコードのkeyフィールドとして設定します。**matchipv4** コマンドで使用可能な他のkeyフィールド、およびkeyフィールドの設定に使用可能な他の**match**コマンドについては、『*Cisco IOS Flexible NetFlow Command Reference*』を参照してください。
ステップ6	必要に応じてステップ5を繰り返し、レコードの追加keyフィールドを設定します。	—

ステップ7	**match flow cts** {**source** | **destination**} **group-tag**
	例：
	`Device(config-flow-record)# match flow cts source group-tag`
	`Device(config-flow-record)# match flow cts destination group-tag`
		（注） この例では、CTSの送信元グループ タグと宛先グループ タグをレコードのキーフィールドとして設定します。**matchipv4** コマンドで使用可能な他のkeyフィールド、およびkeyフィールドの設定に使用可能な他の**match**コマンドについては、『*Cisco IOS Flexible NetFlow Command Reference*』を参照してください。

	コマンドまたはアクション	目的
		（注） • Ingress: • 着信パケットでは、ヘッダーがある場合、SGT にはヘッダーと同じ値が反映されます。値がない場合は、0 が示されます。 • DGT 値は入力ポートの SGACL 設定に依存しません。 • Egress: • SGT または CTS のいずれかの伝播が出力インターフェイス上で無効化されていると、SGT は 0 になります。 • 発信パケットで、（SGT、DGT）に対応する SGACL 設定が存在すれば、DGT はゼロ以外になります。 • SGACL が出力ポート/VLANで無効化されているか、またはグローバル SGACL の強制を無効化されている場合、DGT は 0 になります。
ステップ 8	**例**：	入力インターフェイスをレコードの nonkey フィールドとして設定します。

	コマンドまたはアクション	目的
		（注） この例では、入力インターフェイスをレコードの nonkey フィールドとして設定します。nonkey フィールドの設定に使用可能な他の **collect** コマンドについては、『*Cisco IOS Flexible NetFlow Command Reference*』を参照してください。
ステップ 9	必要に応じて上記のステップを繰り返し、レコードの追加 nonkey フィールドを設定します。	—
ステップ 10	**end** **例**： `Device(config-flow-record)# end`	Flexible NetFlow フロー レコード 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	**showflowrecord** *record-name* **例**： `Device# show flow record FLOW_RECORD-1`	（任意）指定したフロー レコードの現在のステータスが表示されます。
ステップ 12	**showrunning-configflowrecord** *record-name* **例**： `Device# show running-config flow record FLOW_RECORD-1`	（任意）指定したフロー レコードの設定が表示されます。

フロー エクスポータの作成

フロー エクスポートを作成して、フローのエクスポート パラメータを定義できます。

（注） フローエクスポータごとに、1 つ宛先のみがサポートされます。複数の宛先にデータをエクスポートする場合は、複数のフロー エクスポータを設定してフロー モニタに割り当てる必要があります。

IPv4 または IPv6 アドレスを使用して宛先にエクスポートできます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**flow exporter** *name* 例： `Device(config)# flow exporter ExportTest`	フローエクスポータを作成し、フローエクスポータ コンフィギュレーション モードを開始します。このコマンドを使用して既存のフローエクスポータを変更することもできます。
ステップ 3	**description** *string* 例： `Device(config-flow-exporter)# description ExportV9`	（任意）最大 63 文字で、このフローの説明を指定します。

ステップ 4	**destination** {*ipv4-address*|*ipv6-address*}
	例：
	`Device(config-flow-exporter)# destination 192.0.2.1 (IPv4 destination)`
	`Device(config-flow-exporter)# destination 2001:0:0:24::10 (IPv6 destination)`
		このエクスポータに IPv4/IPv6 宛先アドレスまたはホスト名を設定します。

ステップ 5	**dscp** *value* 例： `Device(config-flow-exporter)# dscp 0`	（任意）DSCP（DiffServ コードポイント）値を指定します。範囲は 0 ～ 63 です。デフォルトは 0 です。
ステップ 6	**source** { *source type* } 例： `Device(config-flow-exporter)# source gigabitEthernet1/0/1`	（任意）設定された宛先で NetFlow コネクタに到達するために使用するインターフェイスを指定します。送信元として次のインターフェイスを設定できます。 • **Auto Template**：自動テンプレート インターフェイス

	コマンドまたはアクション	目的
		• **Capwap**：Capwap トンネル インターフェイス • **GigabitEthernet**：Gigabit Ethernet IEEE 802 • **GroupVI**：グループ仮想インターフェイス • **Internal Interface**：内部インターフェイス • **Loopback**：ループバック インターフェイス • **Null**：ヌル インターフェイス • **Port-channel**：インターフェイスのイーサネット チャネル • **TenGigabitEthernet**：10 ギガビット イーサネット • **Tunnel**：トンネル インターフェイス • **Vlan**：Catalyst VLAN
ステップ 7	**transportudp** *number* 例： `Device(config-flow-exporter)# transport udp 200`	（任意）NetFlow コレクタに到達するために使用する UDP ポートを指定します。範囲は 0 ～ 65535 です。 プロトコルをエクスポートする IPFIX の場合、デフォルトの宛先ポートは 4739 です。
ステップ 8	**ttl** *seconds* 例： `Device(config-flow-exporter)# ttl 210`	（任意）エクスポータによって送信されるデータグラムの存続可能時間（TTL）値を設定します。範囲は 1 ～ 255 秒です。デフォルトは 255 です。
ステップ 9	**export-protocol** {**netflow-v5** |**netflow-v9** | **ipfix**} 例： `Device(config-flow-exporter)# export-protocol netflow-v9`	エクスポータで使用される NetFlow エクスポートプロトコルのバージョンを指定します。 • デフォルト値：**netflow-v9**.
ステップ 10	**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-flow-record)# end`	
ステップ 11	**show flow exporter** [**name** *record-name*] 例： `Device show flow exporter ExportTest`	（任意）NetFlow のフロー エクスポータ情報を表示します。
ステップ 12	**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

フロー レコードおよびフロー エクスポータに基づいて、フロー モニタを定義します。

カスタマイズしたフロー モニタの作成

カスタマイズしたフロー モニタを作成するには、この必須のタスクを実行します。

各フロー モニタには、専用のキャッシュが割り当てられています。フロー モニタごとに、キャッシュ エントリの内容およびレイアウトを定義するレコードが必要です。これらのレコード フォーマットは、事前定義済みのレコード フォーマットのいずれか、またはユーザ定義にすることができます。上級のユーザであれば**flowrecord** コマンドを使用して、カスタマイズしたフォーマットを作成することもできます。

始める前に

Flexible NetFlow の事前定義済みレコードの代わりにカスタマイズしたレコードを使用する場合は、このタスクを実行する前に、カスタマイズしたレコードを作成する必要があります。データをエクスポートするためにフロー エクスポータをフロー モニタに追加する場合は、このタスクを完了する前にエクスポータを作成する必要があります。

（注） フロー モニタで **record** コマンドのパラメータを変更する前に、**no ip flow monitor** コマンドを使用して、フロー モニタを適用したすべてのインターフェイスから、フロー モニタを削除しておく必要があります。**ip flowmonitor** コマンドの詳細については、『*Cisco IOS Flexible NetFlow Command Reference*』を参照してください。

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 モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**flow monitor** *monitor-name* 例： `Device(config)# flow monitor FLOW-MONITOR-1`	フロー モニタを作成し、Flexible NetFlow フロー モニタ コンフィギュレーション モードを開始します。 • このコマンドでは、既存のフロー モニタを変更することもできます。
ステップ 4	**description** 説明 例： `Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis`	（任意）フローモニタの説明を作成します。
ステップ 5	**record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6**} *record* [**peer**]} 例： `Device(config-flow-monitor)# record FLOW-RECORD-1`	フロー モニタのレコードを指定します。
ステップ 6	**cache** {**entries** *number* |**timeout** {**active** | **inactive** | **update**} *seconds* | {**immediate** | **normal** | **permanent**}} 例：	**timeout** キーワードに関連するキーワードの値は、キャッシュ タイプが **immediate** に設定されている場合には反映されません。 指定したフローモニタとフローキャッシュを関連付けます。
ステップ 7	必要に応じてステップ 6 を繰り返して、このフローモニタのキャッシュ パラメータの変更を完了します。	—

	コマンドまたはアクション	目的
ステップ 8	**statisticspacket protocol** 例： `Device(config-flow-monitor)# statistics packet protocol`	（任意）Flexible NetFlow モニタのプロトコル分散統計情報の収集をイネーブルにします。
ステップ 9	**statisticspacket size** 例： `Device(config-flow-monitor)# statistics packet size`	（任意）Flexible NetFlow モニタのサイズ分散統計情報の収集をイネーブルにします。
ステップ 10	**exporter** *exporter-name* 例： `Device(config-flow-monitor)# exporter EXPORTER-1`	（任意）事前に作成されたエクスポータの名前を指定します。
ステップ 11	**end** 例： `Device(config-flow-monitor)# end`	Flexible NetFlow フローモニタ 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 12	**showflowmonitor**[[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]] [**statistics**]] 例： `Device# show flow monitor FLOW-MONITOR-2 cache`	（任意）Flexible NetFlow フローモニタのステータスおよび統計情報を表示します。
ステップ 13	**showrunning-configflowmonitor** *monitor-name* 例： `Device# show running-config flow monitor FLOW_MONITOR-1`	（任意）指定したフローモニタの設定が表示されます。
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。

フロー サンプリングの設定および有効化フロー サンプラーの作成

フロー サンプラーを設定して有効化するには、この必須のタスクを実行します。

（注） 「NetFlow original」/「NetFlow IPv4 original input」/「NetFlow IPv6 original input」事前定義済みレコードをフロー モニタに指定して、以前の NetFlow をエミュレートする場合は、フロー モニタを入力（受信）トラフィックの分析だけに使用できます。

「NetFlow IPv4 original output」/「NetFlow IPv6 original output」事前定義済みレコードをフロー モニタに指定して、出力 NetFlow アカウンティング機能をエミュレートする場合は、フロー モニタを出力（発信）トラフィックの分析だけに使用できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**sampler** *sampler-name* 例： `Device(config)# sampler SAMPLER-1`	サンプラーを作成し、サンプラー コンフィギュレーション モードを開始します。 • このコマンドでは、既存のサンプラーを変更することもできます。
ステップ 4	**description** 説明 例： `Device(config-sampler)# description Sample at 50%`	（任意）フローサンプラーの説明を作成します。
ステップ 5	**mode** {**random**} **1 out-of** *window-size* 例： `Device(config-sampler)# mode random 1 out-of 2`	サンプラー モードおよびフローサンプラーのウィンドウ サイズを指定します。 • *window-size* 引数の範囲は、0 ～ 10242 ～ 32768 です。

	コマンドまたはアクション	目的
ステップ 6	**exit** 例： `Device(config-sampler)# exit`	サンプラー コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**interface** *type number* 例： `Device(config)# interface GigabitEthernet 0/0/0`	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 8	{**ip** | **ipv6**} **flowmonitor** *monitor-name* [[**sampler**] *sampler-name*] {**input** | **output**} 例： `Device(config-if)# ip flow monitor FLOW-MONITOR-1 sampler SAMPLER-1 input`	作成したフローモニタおよびフローサンプラーをインターフェイスに割り当てて、サンプリングをイネーブルにします。
ステップ 9	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 10	**showsamplersampler-name** 例： `Device# show sampler SAMPLER-1`	設定し有効化したフローサンプラーのステータスおよび統計情報を表示します。

インターフェイスへのフローの適用

フロー モニタおよびオプションのサンプラーをインターフェイスに適用できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type* 例： `Device(config)# interface`	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、インターフェイスを設定します。

	コマンドまたはアクション	目的
	GigabitEthernet1/0/1	Flexible NetFlow は、L2 ポートチャネル インターフェイスではサポートされませんが、L2 ポートチャネル メンバー ポートではサポートされます。 Flexible NetFlow は、L3 ポートチャネル メンバー ポートではサポートされませんが、L3 ポートチャネル インターフェイスではサポートされます。 インターフェイス コンフィギュレーション のコマンド パラメータは次のとおりです。 • **GigabitEthernet**：GigabitEthernet IEEE 802 • **Loopback**：ループバック インターフェイス • **TenGigabitEthernet**：10 ギガビット イーサネット • **Vlan**：Catalyst VLAN • **Range**：インターフェイス範囲 • **WLAN**：WLAN インターフェイス
ステップ 3	{**ip flow monitor** | **ipv6 flow monitor**}*name* [|**sampler** *name*] { **input**} 例： `Device(config-if)# ip flow monitor MonitorTest input`	入力または出力パケットに対応するインターフェイスに、IPv4 または IPv6 フロー モニタ、およびオプションのサンプラーを関連付けます。
ステップ 4	**end** 例： `Device(config-flow-monitor)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show flow interface** [*interface-type number*] 例： `Device# show flow interface`	（任意）インターフェイスの NetFlow 情報を表示します。

	コマンドまたはアクション	目的
ステップ 6	**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。

VLAN 上でのブリッジ型 NetFlow の設定

フロー モニタおよびオプションのサンプラーを VLAN に適用できます。

手順

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

ステップ 2	**vlan** [**configuration**] *vlan-id*
	例：
	`Device(config)# vlan configuration 30`
	`Device(config-vlan-config)#`
		VLAN または VLAN 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 flow monitor** *monitor name* [**sampler** *sampler name*] {**input** |**output**} 例： `Device(config-vlan-config)# ip flow monitor MonitorTest input`	入力または出力パケットに対応する VLANに、フローモニタおよびオプションのサンプラーを関連付けます。
ステップ 4	**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。

レイヤ 2 NetFlow の設定

Flexible NetFlow レコード内でレイヤ 2 キーを定義できます。このレコードを使用して、レイヤ 2 インターフェイスのフローをキャプチャできます。

手順

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

ステップ 2	**flow record** *name*
	例：
	`Device(config)# flow record L2_record`
	`Device(config-flow-record)#`
		フロー レコード 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	**match datalink** {**dot1q** |**ethertype** | **mac** | **vlan**} 例： `Device(config-flow-record)# match datalink ethertype`	レイヤ2属性をキーとして指定します。
ステップ 4	**end** 例： `Device(config-flow-record)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show flow record** [*name*] 例： `Device# show flow record`	（任意）インターフェイスの NetFlow 情報を表示します。
ステップ 6	**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。

データ リンクの入出力方向にフロー モニタを適用する WLAN 設定

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	**wlan [wlan-name { wlan-id SSID_NetworkName | wlan_id} | wlan-name | shutdown}** 例： `Device (config) # wlan wlan1`	WLAN コンフィギュレーション サブモードを開始します。 *wlan-id* はワイヤレス LAN の ID です。指定できる範囲は 1 ～ 64 です。 SSID_NetworkName は、最大 32 文字の英数字からなる SSID です。 （注） すでにこのコマンドを設定している場合は、**wlan** *wlan-name* コマンドを入力します。
ステップ 3	**datalink flow monitor monitor-name {input | output}** 例： `Device (config-wlan) # datalink flow monitor flow-monitor-1 {input | output}`	目的の方向のレイヤ2トラフィックにフロー モニタを適用します。
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 run wlan** *wlan-name* 例： `Device # show wlan mywlan`	（任意）設定を確認します。

例

IPV4 および IPv6 の入出力方向にフロー モニタを適用する WLAN 設定

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	**wlan {wlan-name { wlan-id SSID_NetworkName | wlan_id} | wlan-name | shutdown}** 例： `Device (config) # wlan wlan1`	WLAN コンフィギュレーション サブモードを開始します。 *wlan-id* はワイヤレス LAN の ID です。指定できる範囲は 1 ～ 64 です。 SSID_NetworkName は、最大 32 文字の英数字からなる SSID です。 （注） すでにこのコマンドを設定している場合は、**wlan** *wlan-name* コマンドを入力します。
ステップ 3	**{ip | ipv6} flow monitor monitor-name {input | output}** 例： `Device (config-wlan) # ip flow monitor flow-monitor-1 input`	入力または出力パケットに対応する WLAN にフロー モニタを関連付けます。
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 run wlan** *wlan-name* 例： `Device # show wlan mywlan`	（任意）設定を確認します。

例

Flexible NetFlow の監視

次の表にあるコマンドを使用して、Flexible NetFlow をモニタリングできます。

表 *4 : Flexible NetFlow* のモニタリング コマンド

コマンド	目的
show flow exporter [**broker** | **export-ids** | **name** | *name* | **statistics** | **templates**]	NetFlow のフロー エクスポータ情報と統計情報を表示します。
show flow exporter [**name** *exporter-name*]	NetFlow のフロー エクスポータ情報と統計情報を表示します。
show flow interface	NetFlowインターフェイスに関する情報を表示します。
show flow monitor [**name** *exporter-name*]	NetFlow のフロー モニタ情報と統計情報を表示します。
show flow monitor statistics	フロー モニタの統計情報を表示します。
show flow monitor cache format {**table** | **record** | **csv**}	指定された形式でフロー モニタのキャッシュの内容を表示します。
show flow record [**name** *record-name*]	NetFlowのフロー レコード情報を表示します。
show flow ssid	WLAN の NetFlow モニタのインストール ステータスを表示します。
show sampler [**broker** | **name** | *name*]	NetFlowサンプラに関する情報を表示します。
show wlan *wlan-name*	デバイスで設定された WLAN を表示します。

Flexible NetFlow の設定例

例：フローの設定

フローを作成し、そのフローをインターフェイスに適用する例を示します。

```
Device# configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.

Device(config)# flow export export1
Device(config-flow-exporter)# destination 10.0.101.254
Device(config-flow-exporter)# transport udp 2055
Device(config-flow-exporter)# exit
Device(config)# flow record record1
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match ipv4 protocol
Device(config-flow-record)# match transport source-port
Device(config-flow-record)# match transport destination-port
Device(config-flow-record)# match flow cts source group-tag
Device(config-flow-record)# match flow cts destination group-tag
Device(config-flow-record)# collect counter byte long
Device(config-flow-record)# collect counter packet long
Device(config-flow-record)# collect timestamp absolute first
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# exit
Device(config)# flow monitor monitor1
Device(config-flow-monitor)# record record1
Device(config-flow-monitor)# exporter export1
Device(config-flow-monitor)# exit
Device(config)# interface tenGigabitEthernet 1/0/1
Device(config-if)# ip flow monitor monitor1 input
Device(config-if)# end
```

例：IPv4 入力トラフィックのモニタリング

次の例は、IPv4 入力トラフィックをモニタする方法を示しています（int g1/0/11 は、int g1/0/36 および int g3/0/11 にトラフィックを送信します）。

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# flow record  fr-1
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match interface input
Device(config-flow-record)# collect counter bytes long
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# collect timestamp absolute first
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# collect counter bytes layer2 long
Device(config-flow-record)# exit

Device(config)# flow exporter fe-ipfix6
Device(config-flow-exporter)# destination 2001:0:0:24::10
Device(config-flow-exporter)# source Vlan106
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow exporter fe-ipfix
Device(config-flow-exporter)# description IPFIX format collector 100.0.0.80
Device(config-flow-exporter)# destination 100.0.0.80
Device(config-flow-exporter)# dscp 30
```

```
Device(config-flow-exporter)# ttl 210
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow exporter fe-1
Device(config-flow-exporter)# destination 10.5.120.16
Device(config-flow-exporter)# source Vlan105
Device(config-flow-exporter)# dscp 32
Device(config-flow-exporter)# ttl 200
Device(config-flow-exporter)# transport udp 2055

Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow monitor fm-1
Device(config-flow-monitor)# exporter fe-ipfix6
Device(config-flow-monitor)# exporter fe-ipfix
Device(config-flow-monitor)# exporter fe-1
Device(config-flow-monitor)# cache timeout inactive 60
Device(config-flow-monitor)#  cache timeout active 180
Device(config-flow-monitor)# record fr-1
Device(config-flow-monitor)# end

Device# show running-config interface g1/0/11
Device# show running-config interface g1/0/36
Device# show running-config interface g3/0/11
Device# show flow monitor fm-1 cache format table
```

例：IPv4 出力トラフィックのモニタリング

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# flow record  fr-1 out
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match interface output
Device(config-flow-record)# collect counter bytes long
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# collect timestamp absolute first
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# exit

Device(config)# flow exporter fe-1
Device(config-flow-exporter)# destination 10.5.120.16
Device(config-flow-exporter)# source Vlan105
Device(config-flow-exporter)# dscp 32
Device(config-flow-exporter)# ttl 200
Device(config-flow-exporter)# transport udp 2055
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow exporter fe-ipfix6
Device(config-flow-exporter)# destination 2001:0:0:24::10
Device(config-flow-exporter)# source Vlan106
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
```

```
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow exporter fe-ipfix
Device(config-flow-exporter)# description IPFIX format collector 100.0.0.80
Device(config-flow-exporter)# destination 100.0.0.80
Device(config-flow-exporter)# dscp 30
Device(config-flow-exporter)# ttl 210
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow monitor fm-1-output
Device(config-flow-monitor)# exporter fe-1
Device(config-flow-monitor)# exporter fe-ipfix6
Device(config-flow-monitor)# exporter fe-ipfix
Device(config-flow-monitor)# cache timeout inactive 50
Device(config-flow-monitor)#  cache timeout active 120
Device(config-flow-monitor)# record fr-1-out
Device(config-flow-monitor)# end

Device# show flow monitor fm-1-output cache format table
```

例：WLAN（入力方向）の IPv4 Flexible NetFlow の設定

次に、WLAN 入力方向で IPv4 Flexible NetFlow を設定する例を示します。

```
flow record WLAN-FLOW07
 description Working AP mac
 match datalink mac source address input
 match ipv4 tos
 match ipv4 protocol
 match ipv4 source address
 match ipv4 destination address
 match wireless ssid
 collect counter bytes long
 collect counter packets long
 collect wireless ap mac address
flow monitor WLAN-FLOW07
 exporter wlan-export
 cache timeout inactive 30
 cache timeout active 10
 record WLAN-FLOW07
wlan CC0506-CC0404
 ip flow monitor WLAN-FLOW07 input

Device#show flow monitor WLAN-FLOW07 cache
Cache type: Normal (Platform cache)
Cache size: Unknown
Current entries: 6

Flows added: 276
Flows aged: 270

    Active timeout ( 10 secs) 257
    Inactive timeout ( 30 secs) 13

DATALINK MAC SOURCE ADDRESS INPUT: 3CA9.F421.4E34
IPV4 SOURCE ADDRESS: 192.168.11.1
```

```
IPV4 DESTINATION ADDRESS: 10.29.5.6
WIRELESS SSID: CC0506-CC0404
IP TOS: 0x00
IP PROTOCOL: 6
counter bytes long: 66
counter packets long: 1
wireless ap mac address: B0AA.778E.EB60
```

例：WLAN（出力方向）の IPv6 および転送フラグ Flexible NetFlow の設定

次に、WLAN 出力方向で IPv6 および転送フラグ Flexible NetFlow を設定する例を示します。

```
Device# configure terminal
Device(config)# flow record fr_v6
Device(config-flow-record)# match ipv6 destination address
Device(config-flow-record)# match ipv6 source address
Device(config-flow-record)# match ipv6 hop-limit
Device(config-flow-record)# match ipv6 protocol
Device(config-flow-record)# match ipv6 traffic
Device(config-flow-record)# match ipv6 version
Device(config-flow-record)# match wireless ssid
Device(config-flow-record)# collect wireless ap mac address
Device(config-flow-record)# collect counter bytes long
Device(config-flow-record)# collect transport tcp flags
Device(config-flow-record)# exit

Device(config)# flow monitor fm_v6
Device(config-flow-monitor)# record fr_v6
Device(config-flow-monitor)# exit

Device(config)# wlan wlan_1
Device(config-wlan)# ipv6 flow monitor fm_v6 out
Device(config-wlan)# end

Device# show flow monitor fm_v6 cache
```

（注） デバイスでは、収集する TCP フラグを指定できません。転送 TCP フラグの収集のみ指定できます。

例：WLAN（入力および出力の両方向）の IPv6 Flexible NetFlow の設定

次に、双方向の WLAN 上で IPv6 Flexible NetFlow を設定する例を示します。

```
Device# configure terminal
Device (config)# flow record fr_v6
Device (config-flow-record)# match ipv6 destination address
Device (config-flow-record)# match ipv6 source address
Device (config-flow-record)# match ipv6 hop-limit
```

```
Device (config-flow-record)# match ipv6 protocol
Device (config-flow-record)# match ipv6 traffic
Device (config-flow-record)# match ipv6 version
Device (config-flow-record)# match wireless ssid
Device (config-flow-record)# collect wireless ap mac address
Device (config-flow-record)# collect counter packets long
Device (config-flow-record)# exit

Device (config)# flow monitor fm_v6
Device (config-flow-monitor)# record fr_v6
Device (config-flow-monitor)# exit

Device (config)# wlan wlan_1
Device (config-wlan)# ipv6 flow monitor fm_v6 in
Device (config-wlan)# ipv6 flow monitor fm_v6 out
Device (config-wlan)# end

Device# show flow monitor fm_v6 cache
```

例：ワイヤレス入力トラフィックのモニタリング

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# flow record fr-wlan-input
Device(config-flow-record)# match datalink mac source address input
Device(config-flow-record)# match datalink mac destination address input
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match wireless ssid
Device(config-flow-record)# collect counter bytes long
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# collect timestamp absolute first
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# exit

Device(config)# flow exporter fe-ipfix
Device(config-flow-exporter)# description IPFIX format collector 100.0.0.80
Device(config-flow-exporter)# destination 100.0.0.80
Device(config-flow-exporter)# dscp 30
Device(config-flow-exporter)# ttl 210
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow exporter fe-ipfix6
Device(config-flow-exporter)# destination 2001:0:0:24::10
Device(config-flow-exporter)# source Vlan106
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow monitor fm-wlan-input
Device(config-flow-monitor)# exporter fe-ipfix
Device(config-flow-monitor)# exporter fe-ipfix6
Device(config-flow-monitor)# cache timeout inactive 30
Device(config-flow-monitor)#  cache timeout active 180
Device(config-flow-monitor)# record fr-wlan-input
```

```
Device(config-flow-monitor)# end

Device# show running-config wlan nfl_1
Device# show flow monitor fm-wlan-input cache format table
```

その他の参考資料

関連資料

関連項目	マニュアル タイトル
Flexible NetFlow の CLI コマンド	『*Cisco Flexible NetFlow Command Reference (Catalyst 3850 Switches)*』 『*Flexible NetFlow Command Reference, Cisco IOS XE Release 3SE (Cisco WLC 5700 Series)*』

エラー メッセージ デコーダ

説明	リンク
このリリースのシステムエラーメッセージを調査し解決するために、エラー メッセージ デコーダ ツールを使用します。	https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

標準および RFC

標準/RFC	Title
RFC 3954	『Cisco Systems NetFlow Services Export Version 9』

MIB

MIB	MIB のリンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	Link
シスコのサポートWebサイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

Flexible NetFlow の機能情報

リリース	変更内容
Cisco IOS XE 3.2SE	この機能が導入されました。
Cisco IOS XE 3.3SE	次の新しいコマンドが追加されました。 • **match wireless ssid** • **collect wireless ap mac address**