Flexible NetFlow の概要

NetFlow は、ルータを通過するパケットの統計情報が得られる Cisco IOS テクノロジーです。NetFlow は、IP ネットワークから実際の IP データを取得するための標準規格です。 NetFlow は、ネットワークとセキュリティの監視、ネットワーク計画、トラフィック分析、および IP アカウンティングをイネーブルにするためのデータを提供します。

Flexible NetFlow は、実際の要件に合わせてトラフィック分析パラメータをカスタマイズする機能を追加することで、以前の NetFlow よりも改善されています。 Flexible NetFlow では、トラフィック分析のための非常に複雑な構成を作成したり、再利用可能な構成コンポーネントを使用してデータをエクスポートすることが容易になります。

機能情報の確認

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。 最新の警告および機能情報については、『Bug Search Tool』および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。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

Flexible NetFlow の前提条件

- Flexible NetFlow の key 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していること。
  - **match flow**
  - **match interface**
  - **match** {**ipv4** | **ipv6**}
  - **match routing**
  - **match transport**
- 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していること。
  - **collect counter**
  - **collect flow**
  - **collect interface**
  - **collect**{**ipv4** | **ipv6**}
  - **collect routing**
  - **collect timestamp sys-uptime**
  - **collect transport**
- ネットワーキング デバイスで、Flexible NetFlow がサポートされたシスコ リリースが稼働していること。

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。
- シスコ エクスプレス フォワーディングまたは分散型シスコ エクスプレス フォワーディングのいずれかが、使用中のルータおよびFlexible NetFlowをイネーブルにするすべてのインターフェイスでイネーブルにされていること。

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。
- シスコ エクスプレス フォワーディング IPv6 または分散型シスコ エクスプレス フォワーディングのいずれかが、使用中のルータおよび Flexible NetFlow をイネーブルにするすべてのインターフェイスでイネーブルにされていること。

Flexible NetFlow について

Flexible NetFlow の概要

Flexible NetFlow では、トラフィック分析のための非常に複雑な構成を作成したり、再利用可能な構成コンポーネントを使用してデータをエクスポートすることが容易になります。

NetFlow 一般的な使用方法

一般的に、NetFlow は次のような、重要なカスタマー アプリケーションのいくつかで使用されます。

- ネットワークのモニタリング。 NetFlow データでは、ほぼリアルタイムのさまざまなネットワークのモニタリング機能を利用できます。 ネットワーク オペレータはフローベースの分析手法を使用して、個々のルータおよびスイッチに関連付けられたトラフィック パターンやネットワーク全体のトラフィック パターンを視覚化し（集約トラフィックまたはアプリケーションベースのビューの場合）、予防的な問題の検出、効率的なトラブルシューティング、迅速な問題解決を実現します。
- アプリケーションのモニタリングとプロファイリング。 NetFlow のデータを利用すると、ネットワーク マネージャはネットワーク全体における、アプリケーション使用状況を時間ベースで詳細に調べることができます。 この情報は、新しいサービスを計画および理解し、ネットワーク リソースおよびアプリケーション リソースを割り当て（たとえば、Web サーバのサイズ設定と VoIP の配置）、顧客の要求を迅速に満たすために使用されます。
- ユーザのモニタリングとプロファイリング。 ネットワーク エンジニアは NetFlow データを使用すると、顧客やユーザによるネットワーク リソースおよびアプリケーション リソースの利用について詳しく理解できます。 この情報を使用して、潜在的なセキュリティやポリシーの違反を検出して解決するために、アクセス、バックボーン、アプリケーション リソースを効率的に計画して割り当てることができます。
- ネットワーク プランニング。 NetFlow を使用すると、長期間にわたってデータをキャプチャすることによって、ネットワークの成長を追跡して予測し、ルーティング デバイス、ポート、および高帯域幅のインターフェイスの数を増やすためのアップグレードを計画する機会が得られます。 NetFlow サービス データによって、ピアリング、バックボーンのアップグレード、ルーティング ポリシーのネットワーク計画を最適化できます。 NetFlow はネットワークのパフォーマンス、容量、および信頼性を最大化すると同時に、ネットワーク運用の総コストを最小限に抑えるために役立ちます。 NetFlow により、望ましくない WAN トラフィックが検出され、帯域幅と Quality of Service（QoS）が検証され、新しいネットワーク アプリケーションの分析が可能になります。 NetFlow からは、ネットワークの運用コストを削減するための有用な情報が得られます。
- セキュリティの分析。 NetFlow では、分散 DoS（dDoS）攻撃、ウイルス、およびワームをリアルタイムで識別して分類します。 ネットワークの動作が変化すると、Flexible NetFlow デー

タに明らかな異常が表れます。 このデータは、セキュリティ侵害の過程を調べ、再現するための貴重な科学捜査上のツールでもあります。

- 課金とアカウンティング。 NetFlow データを使用すると細かい設定が可能な計測（たとえば、IP アドレス、パケット数やバイト数、タイム スタンプ、タイプ オブ サービス（ToS）、アプリケーション ポートなどの詳細を含むフロー データ）ができるため、非常に柔軟かつ詳細なリソース使用率のアカウンティングを実現できます。 サービス プロバイダーは、時刻、帯域幅の使用率、アプリケーションの使用率、Quality of Service などに基づく課金のためにこの情報を使用できます。 企業のお客様は、リソースの使用率に応じた部門別のチャージバックやコスト割り当てに、これらの情報を利用することがあります。
- NetFlow データのウェアハウジングとデータ マイニング。 NetFlow のデータ（または、そこから得られた情報）を保管し、後から取り出して分析することで、積極的なマーケティングおよびカスタマー サービス プログラムをサポートできます（たとえば、内部および外部ユーザが、どのアプリケーションとサービスを使用しているかを調べ、サービスの向上、広告などをそのユーザ向けにターゲティングできます）。 さらに、Flexible NetFlow データにより、市場調査の担当者は、企業およびサービス プロバイダーに関連した人、物、場所、および期間の情報を入手できます。

以前の NetFlow および Flexible NetFlow でのフローの使用

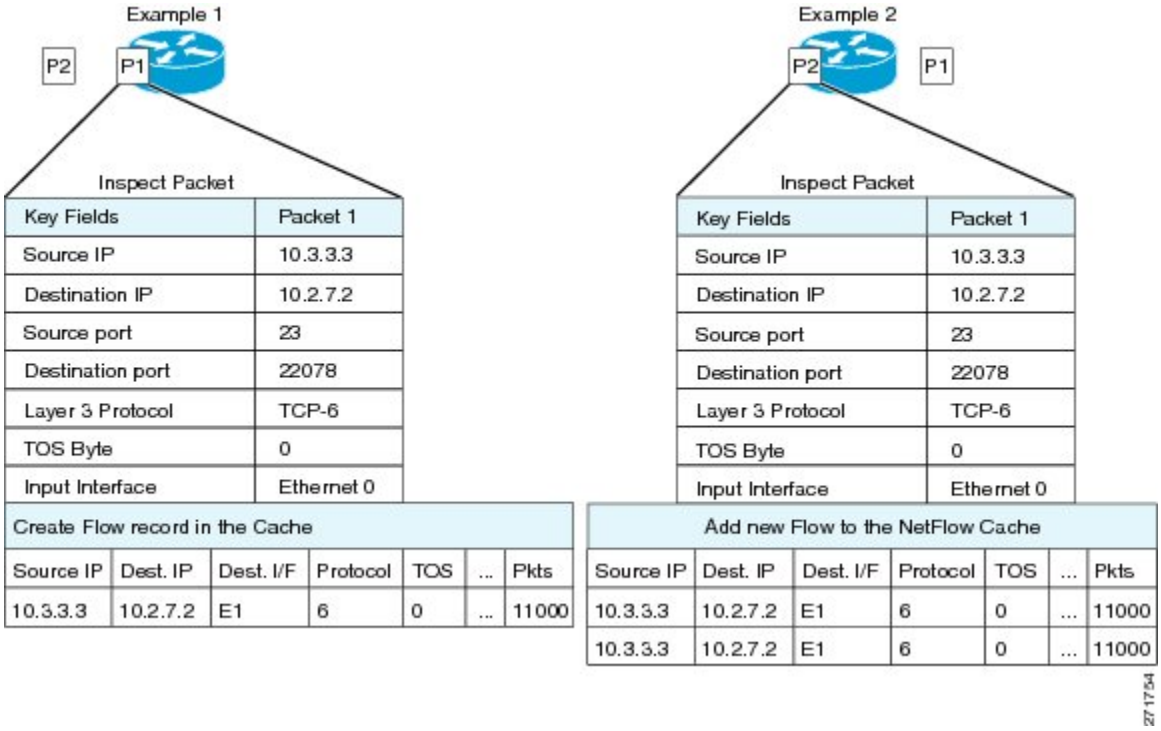
以前の NetFlow および Flexible NetFlow の両方でフローの概念を使用します。 フローは、特定の送信元と特定の宛先の間のパケットのストリームとして定義します。

以前の NetFlow および Flexible NetFlow の両方で、ネットワーク トラフィックの監視中にキャッシュ内にいつ新しいフローを作成する必要があるかを判断するための条件として、IP データグラムの key フィールドの値（IP 送信元アドレスまたは宛先アドレスおよび送信元または宛先のトランスポート プロトコル ポートなど）を使用します。 データグラムの key フィールドのデータの値が既存のフローに関して一意である場合、新しいフローが作成されます。

以前の NetFlow および Flexible NetFlow の両方で、フローからキャプチャされるデータのフィールドを識別するための条件として、nonkey フィールドを使用します。 フローには、nonkey フィールドの値からキャプチャされたデータが格納されます。

次の図に、パケットを検査し、キャッシュ内のフロー レコードを作成するプロセスの例を示します。 この例では、送信元と宛先の IP アドレスの key フィールドに異なる値があるため、2 つの固有のフローが作成されます。

図 *1* : パケットの検査



以前の NetFlow と Flexible NetFlow

以前の NetFlow は、フローの判定に固定 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。

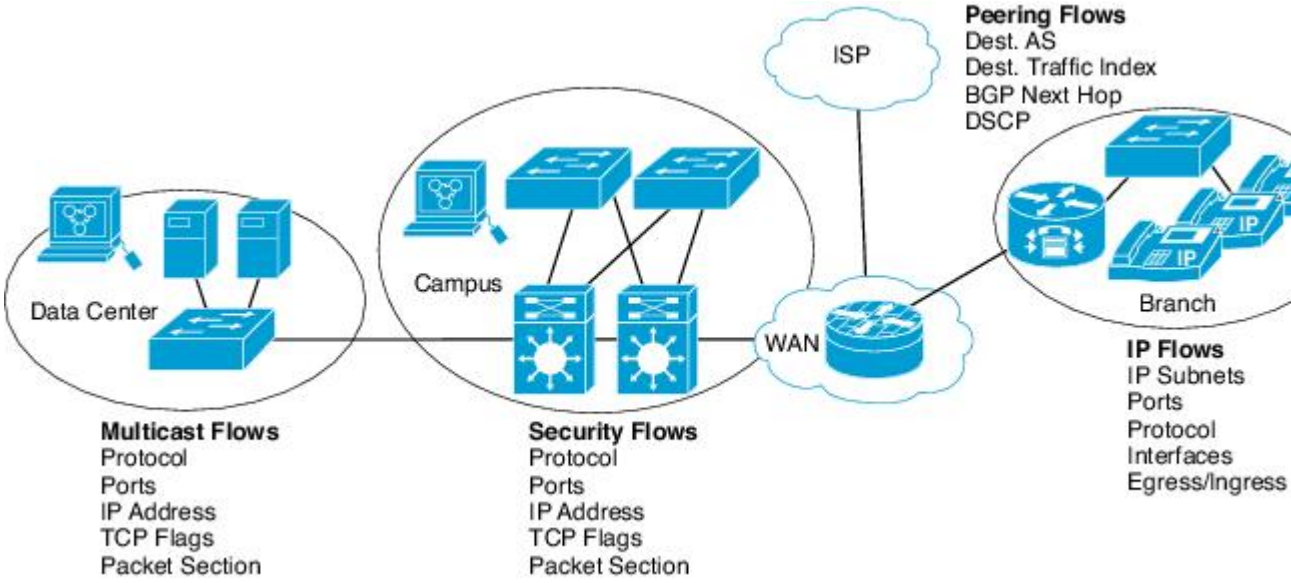
- スケーラビリティ、フロー情報の集約などの、大容量フロー認識。
- セキュリティの監視と dDoS の検出および識別のための拡張されたフロー インフラストラクチャ。
- フロー情報をネットワーク内の特定のサービスまたはオペレーションに適応させるパケットからの新しい情報。 利用できるフロー情報は、Flexible NetFlow ユーザがカスタマイズ可能。
- シスコの柔軟で拡張可能な NetFlow Version 9 エクスポート フォーマットの活用。
- IP アカウンティング、ボーダー ゲートウェイ プロトコル（BGP）ポリシー アカウンティング、永続的キャッシュなどの多数のアカウンテ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。

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。
- マルチプロトコル ラベル スイッチング（MPLS）または IP コア ネットワーク、およびサービス クラスごとの各ネクスト ホップの宛先に着信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。

次の図に、ネットワーク内への Flexible NetFlow の展開方法の例を示します。

図 *2*：*Flexible NetFlow* の代表的な導入



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。

Records

Flexible NetFlow では、key フィールドと nonkey フィールドの組み合わせがレコードと呼ばれます。 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。 Flexible NetFlow の機能を完全に利用するには、次の項で説明するように、カスタマイズした独自のレコードを作成する必要があります。

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。

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。

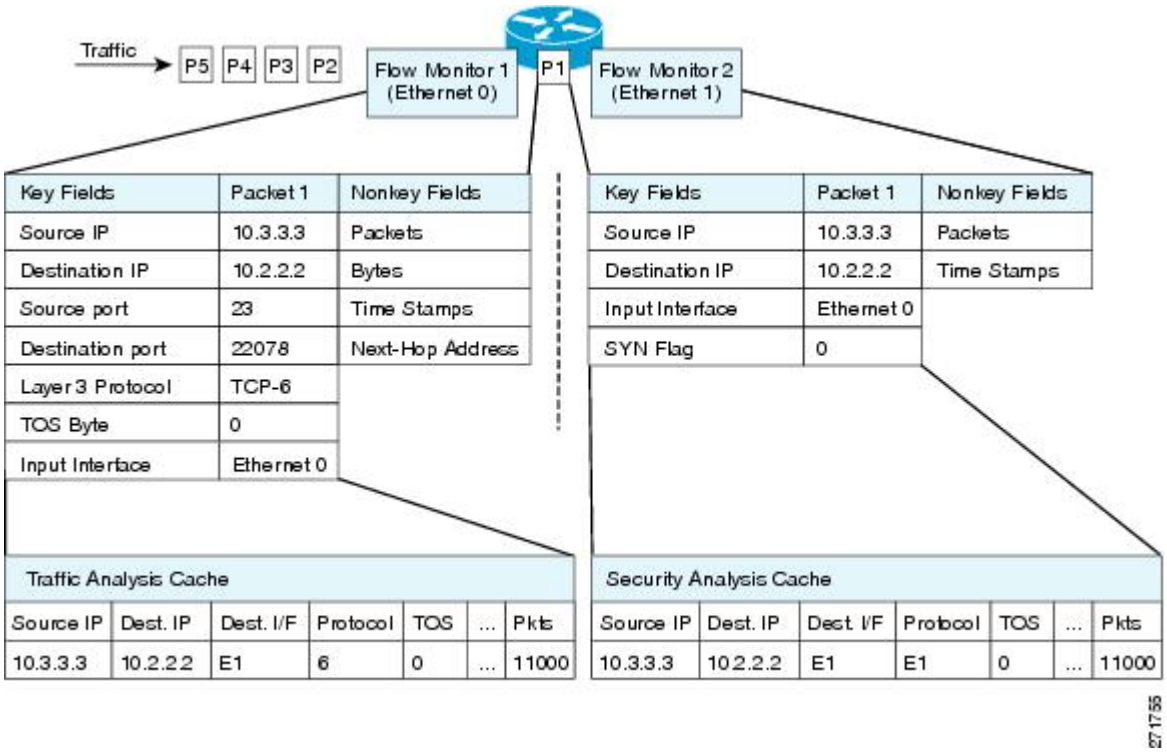
フロー モニタ

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。

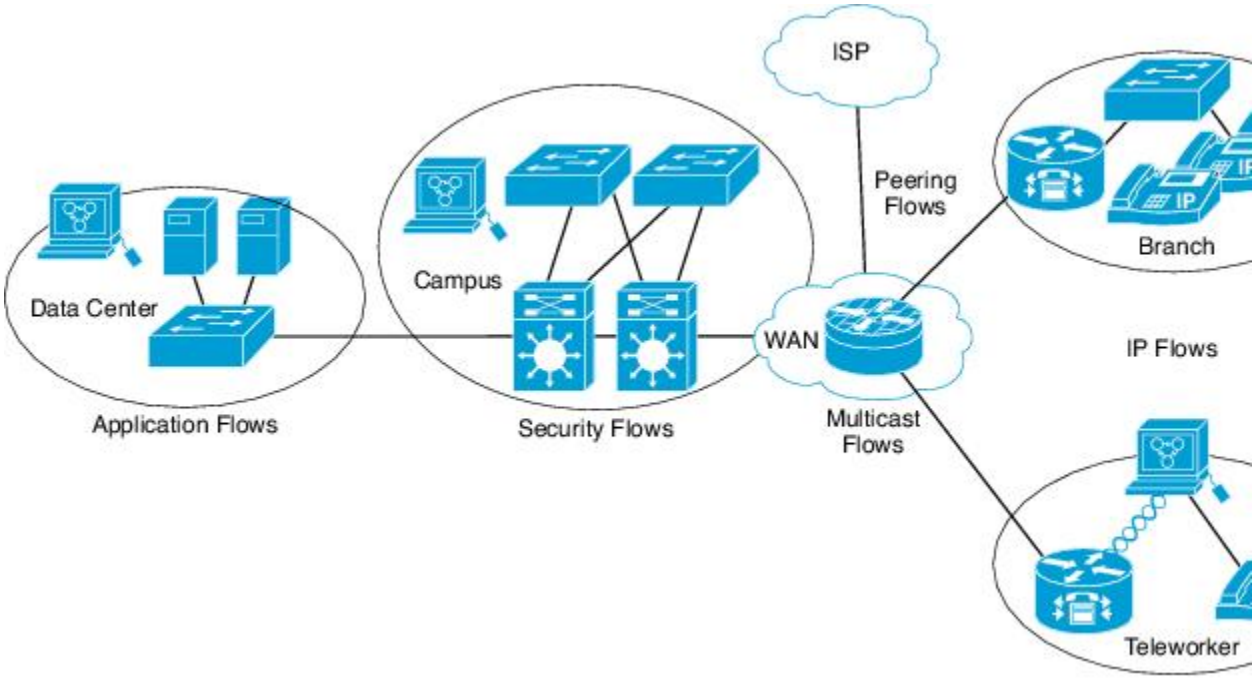
Flexible NetFlow は、同じトラフィックのさまざまなタイプの分析を実行するために使用できます。 次の図では、入力インターフェイス上の標準トラフィック分析のために設計されたレコードと、出力インターフェイス上のセキュリティ分析のために設計されたレコードを使用してパケット 1 が分析されます。

図 *3* : *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



次の図に、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 4：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



Normal

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 と timeout inactive の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。

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

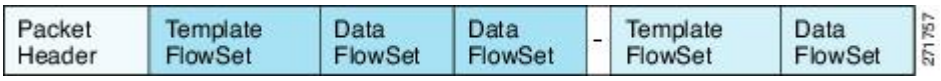
NetFlow の基本出力はフロー レコードです。 NetFlow が改良され、フロー レコードのいくつかのフォーマットが向上しました。 NetFlow エクスポート フォーマットの最新の進化は、バージョン

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。

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、1 つのエクスポート パケット内で混在できます。

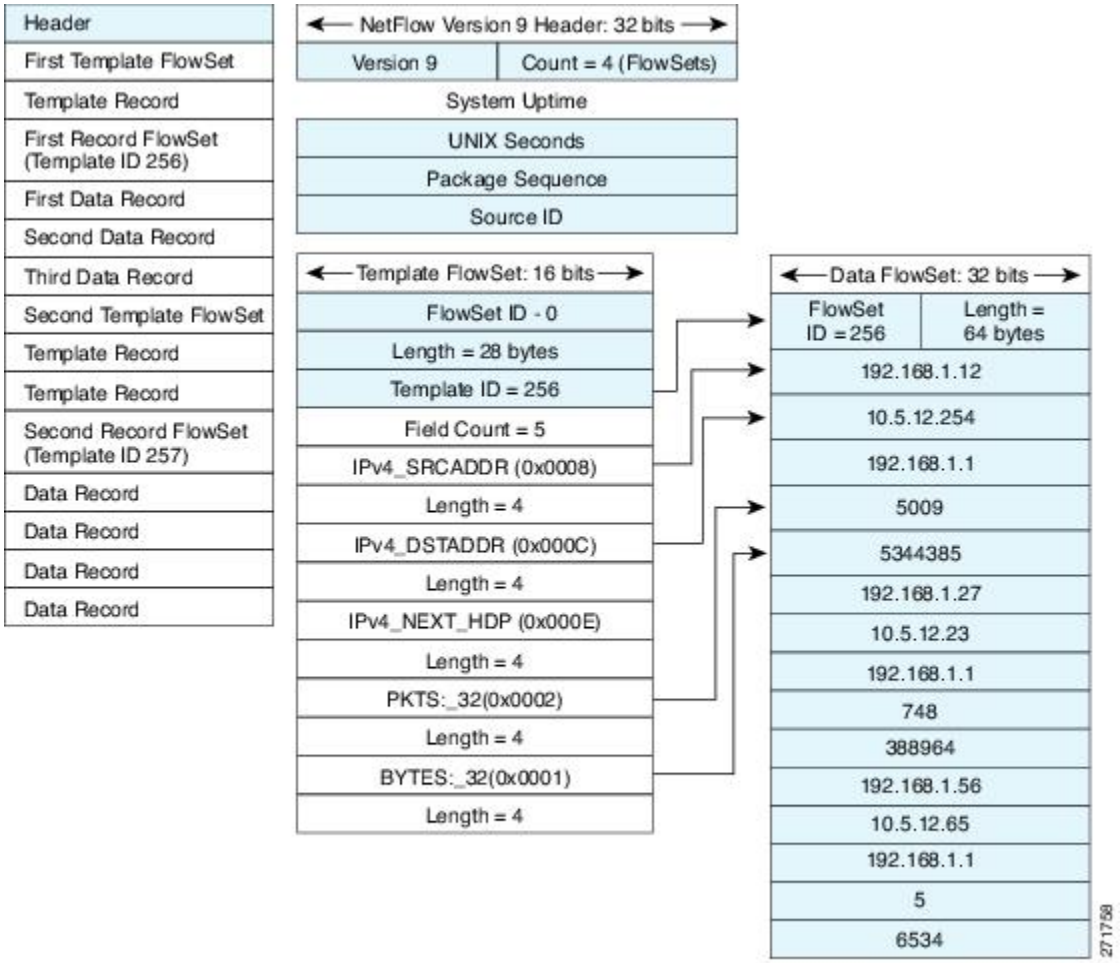
図 *5*：バージョン *9* エクスポート パケット



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。 次の図に、ヘッダー、

テンプレート フローセットおよびデータ フローセットを含む、NetFlow バージョン 9 エクスポート フォーマットの詳細な例を示します。

図 *6*：*NetFlow Version 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 サンプラ

フロー サンプラは、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。

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、サンプラとフロー モニタを組み合わせます。

Flexible NetFlow でのセキュリティ監視

Flexible NetFlow をネットワーク攻撃検出ツールとして IP ヘッダーのすべての部分および均等なパケット セクションを追跡する機能とともに使用して、この情報をフローに特徴付けることができます。 セキュリティ監視システムでは Flexible NetFlow データを分析でき、ネットワーク上の問題を見つけた場合に、特定の情報を追跡して攻撃パターンまたはワーム伝播の詳細を識別するように設定される仮想バケットまたは仮想キャッシュを作成できます。 特定の情報を入力フィルタリング（たとえば、特定の宛先へのすべてのフローのフィルタリング）と組み合わせて動的にキャッシュを作成する機能を備えた Flexible NetFlow は、強度なセキュリティ監視ツールです。

宛先サーバへのオープンな TCP 要求をフラッディングする（たとえば、SYN フラッド攻撃）ために TCP フラグが使用される場合、1 つの共通タイプの攻撃が発生します。 攻撃デバイスは TCP SYN のストリームを特定の宛先アドレスに送信しますが、TCP スリーウェイ ハンドシェイクの一部として、サーバ SYN-ACK に応答して ACK を送信することはありません。 セキュリティ検出サーバに必要なフロー情報では、宛先のアドレスまたはサブセット、TCP フラグ、パケット数などの 3 つの key フィールドの追跡が要求されます。 セキュリティ検出サーバでは一般的な Flexible NetFlow 情報がモニタされ、このデータによって、Flexible NetFlow でルータのコンフィギュレーションに新しいフローを動的に作成することで、この特定の攻撃の詳細ビューがトリガーされます。 新しいフロー モニタには Flexible NetFlow キャッシュに表示されるトラフィックを制限するための入力フィルタリングと、TCP ベースの攻撃を診断するための特定の情報の追跡が含まれます。 この場合、ユーザはサーバの宛先アドレスまたはサブネットへのすべてのフロー情報をフィルタリングして、セキュリティ検出サーバでの評価のために必要な情報量を制限できます。 セキュリティ検出サーバでこの攻撃を理解したと判断された場合、別のフロー モニタをプログラミングして、パケット内の署名を詳細に確認するペイロード情報またはパケットのセクションを収集してエクスポートします。 この例は、Flexible NetFlow をセキュリティ インシデントの検出に使用できる多数の方法の 1 つにすぎません。

Flexible NetFlow の分析に使用されるトラフィックの識別基準

事前定義されている Flexible NetFlow レコードがトラフィック要件に適していない場合は、Flexible NetFlow **collect** コマンドおよび **match** コマンドを使用してユーザ定義（カスタム）レコードを作成できます。 カスタマイズしたレコードを作成する前に、key フィールドおよび nonkey フィールドに対して使用する基準を決定する必要があります。

ネットワーク攻撃検出用のカスタム レコードを作成する場合は、適切な key および nonkey フィールドをレコードに含めることで、ルータが攻撃の分析と対処に必要なフローを作成し、データをキャプチャする必要があります。 たとえば、一般的なサービス拒否（DoS）攻撃である SYN フラッド攻撃では、宛先ホストに対するオープン TCP 要求をフラッディングするために TCP フラグ

が使用されます。 通常の TCP 接続が開始されると、宛先ホストは送信元ホストからの SYN（同期/開始）パケットを受信し、SYN ACK（同期応答確認）を返信します。 宛先ホストは、接続を確立する前に、SYN ACK への ACK（応答確認）を受信する必要があります。 これは、「TCP スリーウェイ ハンドシェイク」と呼ばれます。宛先ホストが SYN ACK への ACK を待機している間、宛先ホスト上のサイズが制限された接続キューは、完了するまで待機しながら、接続を記録します。 ACK は SYN ACK の数ミリ秒後に到着すると予想されるため、このキューは通常、すぐに空になります。 TCP SYN 攻撃ではこの設計を悪用し、攻撃元ホストでランダムな送信元アドレスを使用して被害ホストに対する TCP SYN パケットを生成します。 被害を受けた宛先ホストは SYN ACK をランダムな送信元アドレスに返信し、接続キューにエントリが追加されます。 SYN ACK は適切でないか、または存在していないホストを宛先にするため、TCP スリーウェイ ハンドシェイクの最後の部分が完了せず、エントリはタイマーが期限切れになるまで、通常は約 1 分間、接続キューに残ります。 送信元ホストがランダムな IP アドレスから TCP SYN パケットを迅速に生成する場合、接続キューがいっぱいになる可能性があり、正当なユーザに対する TCP サービス（電子メール、ファイル転送、WWW など）が拒否されるおそれがあります。

このタイプの DoS 攻撃に対するセキュリティ監視レコードに必要な情報には、次の key フィールドおよび nonkey フィールドが含まれることがあります。

- key フィールド
  - 宛先 IP アドレスまたは宛先 IP サブセット
  - TCP フラグ
  - パケット数
- nonkey フィールド
  - 宛先 IP アドレス
  - 送信元 IP アドレス
  - インターフェイス入力および出力

ヒント　ユーザの多くは、これらの key フィールドおよび nonkey フィールドを使用して、DoS 攻撃の詳細な Flexible NetFlow ビューをトリガーする一般的な Flexible NetFlow モニタを設定します。

Flexible NetFlow での以前の NetFlow のエミュレートの利点

Flexible NetFlow で以前の NetFlow をエミュレートすると、カスタム ユーザ定義レコードを設計して設定する代わりに、事前定義済みのレコードを使用できるため、より迅速に Flexible NetFlow を導入できます。 以前の NetFlow と同様に操作を開始するために必要なのは、フロー モニタを設定して Flexible NetFlow のインターフェイスに適用することだけです。 NetFlow コレクタなどのアプリケーションで収集するデータを分析する場合は、任意のエクスポータを追加できます。

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。 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にすることもできますが、上級のユーザであれば Flexible NetFlow フロー レコード コンフィギュレーション モードで **collect** および **match** コマンドを使用して独自のレコード フォーマットを作成することもできます。

フロー エクスポータは、Flexible NetFlow で収集されるデータを NetFlow Collection Engine などのリモート システムへ送信するために使用されます。 エクスポータでは、トランスポート プロトコルとして UDP、エクスポート フォーマットとしてバージョン 9 が使用されます。

以前の NetFlow を使い慣れていれば、以前の NetFlow をエミュレートする場合に Flexible NetFlow で収集してエクスポートするデータのフォーマットおよび内容を理解しているはずです。 データの分析に同じ手法を使用できます。

Flexible NetFlow の事前定義済みレコード

Flexible NetFlow の事前定義済みレコードは、以前の NetFlow の入力キャッシュおよび出力キャッシュと集約キャッシュに基づいています。 以前の NetFlow 集約キャッシュと、それに対応する事前定義済みの Flexible NetFlow のレコードの間の相違点は、事前定義済みレコードでは集約を実行しないということです。 Flexible NetFlow の事前定義済みレコードは、ユーザ定義（カスタム）レコードを関連付けるのと同じ方法で、Flexible NetFlow フロー モニタに関連付けられています。

Flexible NetFlow の事前定義済みレコードの利点

以前の NetFlow または集約キャッシュ付きの以前の NetFlow を使用している場合は、Flexible NetFlow に移行するときに、Flexible NetFlow で使用可能な定義済みレコードを使用して、分析用に同じトラフィック データのキャプチャを継続することができます。 多くのユーザが、既存の Flexible NetFlow レコードが大半のトラフィック分析要件に適していることを理解できるでしょう。

事前定義済みレコード「NetFlow Original」および「NetFlow IPv4 Original Input」

Flexible NetFlow の事前定義済みレコード「NetFlow Original」および「NetFlow IPv4 Original Input」は、key フィールドと nonkey フィールドが同じであるため、同じように使用できます。 次の表に、Flexible NetFlow の事前定義済みレコード「NetFlow Original」と「NetFlow IPv4 Original Input」の key および nonkey フィールドを示します。

表 *1*：*Flexible NetFlow* の事前定義済みレコード「*NetFlow Original*」と「*NetFlow IPv4 Original Input*」で使用される *key* および *nonkey* フィールド

フィールド	**key** フィールドまたは **nonkey** フィールド	定義
IP ToS	Key	タイプ オブ サービス（ToS）フィールドの値。

フィールド	**key** フィールドまたは **nonkey** フィールド	定義
IP Protocol	Key	IP プロトコル フィールドの値。
IP Source Address	Key	IP 送信元アドレス。
IP Destination Address	Key	IP 宛先アドレス。
Transport Source Port	Key	トランスポート層の送信元ポート フィールドの値。
Transport Destination Port	Key	トランスポート層の宛先ポート フィールドの値。
Interface Input	Key	トラフィックが受信されたインターフェイス。
Flow Sampler ID	Key	フロー サンプラの ID 番号（フロー サンプリングがイネーブルにされている場合）。
IP Source AS	Nonkey	送信元自律システム番号。
IP Destination AS	Nonkey	宛先自律システム番号。
IP Next Hop Address	Nonkey	ネクスト ホップの IP アドレス。
IP Source Mask	Nonkey	IP 送信元アドレスのマスク。
IP Destination Mask	Nonkey	IP 宛先アドレスのマスク。
TCP Flags	Nonkey	TCP フラグ フィールドの値。
Interface Output	Nonkey	トラフィックが送信されたインターフェイス。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。

フィールド	key フィールドまたは nonkey フィールド	定義
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「NetFlow IPv4 Original Output」

Flexible NetFlow の事前定義済みレコード「NetFlow IPv4 Original Output」は、Cisco IOS Release 12.3(11)T でリリースされた以前の NetFlow 出力 NetFlow アカウンティング機能をエミュレートするために使用されます。 次の表に、Flexible NetFlow の事前定義済みレコード「NetFlow IPv4 Original Output」の key および nonkey フィールドとカウンタを示します。

表 *2*：*Flexible NetFlow* の事前定義済みレコード「*NetFlow IPv4 Original Output*」で使用される *key* および *nonkey* フィールド

フィールド	key フィールドまたは nonkey フィールド	定義
IP ToS	Key	ToS フィールドの値。
IP Protocol	Key	IP プロトコル フィールドの値。
IP Source Address	Key	IP 送信元アドレス。
IP Destination Address	Key	IP 宛先アドレス。
Transport Source Port	Key	トランスポート層の送信元ポート フィールドの値。
Transport Destination Port	Key	トランスポート層の宛先ポート フィールドの値。
Interface Output	Key	トラフィックが送信されたインターフェイス。

フィールド	key フィールドまたは nonkey フィールド	定義
Flow Sampler ID	Key	フロー サンプラの ID 番号（フロー サンプリングがイネーブルにされている場合）。
IP Source AS	Nonkey	送信元自律システム番号。
IP Destination AS	Nonkey	宛先自律システム番号。
IP Next Hop Address	Nonkey	ネクスト ホップの IP アドレス。
IP Source Mask	Nonkey	IP 送信元アドレスのマスク。
IP Destination Mask	Nonkey	IP 宛先アドレスのマスク。
TCP Flags	Nonkey	TCP フラグ フィールドの値。
Interface Input	Nonkey	トラフィックが受信されたインターフェイス。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

例：IPv4 および IPv6 トラフィック用の Flexible NetFlow Egress Accounting の設定, （58 ページ）の設定では、Flexible NetFlow の事前定義済みレコード「NetFlow Original Output」を使用します。

事前定義済みレコード「NetFlow IPv6 Original Input」

次の表に、Flexible NetFlow の事前定義済みレコード「NetFlow IPv6 Original Input」の key および nonkey フィールドとカウンタを示します。

表 *3*：*Flexible NetFlow* の事前定義済みレコード「*NetFlow IPv6 Original Input*」で使用される *key* および *nonkey* フィールド

フィールド	**key** フィールドまたは **nonkey** フィールド	定義
Traffic Class	Key	トラフィック クラス フィールドの値。
Flow Label	Key	フロー ラベル。
Protocol	Key	プロトコル フィールドの値。
Extension Map	Key	拡張マップ ビットマップの値。
IP Source Address	Key	IP 送信元アドレス。
IP Destination Address	Key	IP 宛先アドレス。
Transport Source Port	Key	トランスポート層の送信元ポート フィールドの値。
Transport Destination Port	Key	トランスポート層の宛先ポート フィールドの値。
Interface Input	Key	トラフィックが受信されたインターフェイス。
Flow Direction	Key	フローの方向。
Flow Sampler	Key	フロー サンプラの ID 番号（フロー サンプリングがイネーブルにされている場合）。
Routing Source AS	Nonkey	送信元自律システム番号。
Routing Destination AS	Nonkey	宛先自律システム番号。
Routing Next-hop Address	Nonkey	ネクスト ホップの IP アドレス。
IP Source Mask	Nonkey	IP 送信元アドレスのマスク。

フィールド	key フィールドまたは nonkey フィールド	定義
IP Destination Mask	Nonkey	IP 宛先アドレスのマスク。
Transport TCP Flags	Nonkey	TCP フラグ フィールドの値。
Interface Output	Nonkey	トラフィックが送信されたインターフェイス。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「NetFlow IPv6 Original Output」

次の表に、Flexible NetFlow の事前定義済みレコード「NetFlow IPv6 Original Output」の key および nonkey フィールドとカウンタを示します。

表 *4*：*Flexible NetFlow* の事前定義済みレコード「*NetFlow IPv6 Original Output*」で使用される *key* および *nonkey* フィールド

フィールド	key フィールドまたは nonkey フィールド	定義
Traffic Class	Key	トラフィック クラス フィールドの値。
Flow Label	Key	フロー ラベル。
Protocol	Key	プロトコル フィールドの値。
Extension Map	Key	拡張マップ ビットマップの値。

フィールド	**key** フィールドまたは **nonkey** フィールド	定義
IP Source Address	Key	IP 送信元アドレス。
IP Destination Address	Key	IP 宛先アドレス。
Transport Source Port	Key	トランスポート層の送信元ポート フィールドの値。
Transport Destination Port	Key	トランスポート層の宛先ポート フィールドの値。
Interface Output	Key	トラフィックが送信されたインターフェイス。
Flow Direction	Key	フローの方向。
Flow Sampler	Key	フロー サンプラの ID 番号（フロー サンプリングがイネーブルにされている場合）。
Routing Source AS	Nonkey	送信元自律システム番号。
Routing Destination AS	Nonkey	宛先自律システム番号。
Routing Next-hop Address	Nonkey	ネクスト ホップの IP アドレス。
IP Source Mask	Nonkey	IP 送信元アドレスのマスク。
IP Destination Mask	Nonkey	IP 宛先アドレスのマスク。
Transport TCP Flags	Nonkey	TCP フラグ フィールドの値。
Interface Input	Nonkey	トラフィックが受信されたインターフェイス。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。

フィールド	key フィールドまたは nonkey フィールド	定義
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「Autonomous System」

Flexible NetFlow の事前定義済みレコード「Autonomous System」では、自律システム間のトラフィック フロー データに基づいてフローが作成されます。 Flexible NetFlow の事前定義済みレコード「Autonomous System」では、以前の NetFlow の「Autonomous System」の集約キャッシュと同じ key フィールドおよび nonkey フィールドが使用されます。

（注） この事前定義済みレコードは、IPv4 および IPv6 トラフィックの分析に使用できます。

次の表に、Flexible NetFlow の事前定義済みレコード「Autonomous System」で使用される key フィールドおよび nonkey フィールドを示します。

表 *5*：*Flexible NetFlow* の事前定義済みレコード「*Autonomous System*」で使用される *key* フィールドおよび *nonkey* フィールド

フィールド	key フィールドまたは nonkey フィールド	定義
IP Source AS	Key	送信元 IP アドレスの自律システム（ピアまたはオリジン）。
IP Destination AS	Key	宛先 IP アドレスの自律システム（ピアまたはオリジン）。
Interface Input	Key	トラフィックが受信されたインターフェイス。
Interface Output	Key	トラフィックが送信されたインターフェイス。
Flow Direction	Key	フローが監視されている方向。

フィールド	key フィールドまたは nonkey フィールド	定義
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「Autonomous System ToS」

Flexible NetFlow の事前定義済みレコード「Autonomous System ToS」では、自律システム間のトラフィック フロー データおよびタイプ オブ サービス（ToS）トラフィック フロー データに基づいてフローが作成されます。 Flexible NetFlow の事前定義済みレコード「Autonomous System ToS」では、以前の NetFlow の「Autonomous System ToS」の集約キャッシュと同じ key フィールドおよび nonkey フィールドが使用されます。

（注） この事前定義済みレコードは、IPv4 トラフィックの分析だけに使用できます。

ヒント この事前定義済みレコードは、自律システム間のトラフィック フロー データの生成に特に役立ちます。

次の表に、Flexible NetFlow の事前定義済みレコード「Autonomous System ToS」で使用される key フィールドおよび nonkey フィールドを示します。

表 *6*：*Flexible NetFlow* の事前定義済みレコード「*Autonomous System ToS*」で使用される *key* フィールドおよび *nonkey* フィールド

フィールド	key フィールドまたは nonkey フィールド	定義
IP ToS	Key	ToS フィールドの値。

フィールド	key フィールドまたは nonkey フィールド	定義
IP Source autonomous system	Key	送信元 IP アドレスの自律システム（ピアまたはオリジン）。
IP Destination autonomous system	Key	宛先 IP アドレスの自律システム（ピアまたはオリジン）。
Interface Input	Key	トラフィックが受信されたインターフェイス。
Interface Output	Key	トラフィックが送信されたインターフェイス。
Flow Direction	Key	フローが監視されている方向。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「BGP Next-Hop」

Flexible NetFlow の事前定義済みレコード「BGP Next-Hop」では、ボーダー ゲートウェイ プロトコル（BGP）のトラフィック フロー データに基づいてフローが作成されます。

（注） この事前定義済みレコードは、IPv6 トラフィックの分析だけに使用できます。

次の表に、Flexible NetFlow の事前定義済みレコード「BGP Next-Hop」で使用される key フィールドおよび nonkey フィールドを示します。

表 *7*：***Flexible NetFlow*** の事前定義済みレコード「***BGP Next-Hop***」で使用される ***key*** フィールドおよび ***nonkey*** フィールド

フィールド	**key** フィールドまたは **nonkey** フィールド	定義
Routing Source AS	Key	送信元 IP アドレスの自律システム。
Routing Destination AS	Key	宛先 IP アドレスの自律システム。
Routing Next-hop Address IPv6 BGP	Key	BGP ネクスト ホップの IPv6 アドレス。
Interface Input	Key	トラフィックが受信されたインターフェイス。
Interface Output	Key	トラフィックが送信されたインターフェイス。
Flow Direction	Key	フローが監視されている方向。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。
Timestamp Sys-uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Timestamp Sys-uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「BGP Next-Hop ToS」

Flexible NetFlow の事前定義済みレコード「BGP Next-Hop ToS」では、BGP トラフィック フローデータおよびToS トラフィック フローデータに基づいてフローが作成されます。 Flexible NetFlow の事前定義済みレコード「BGP Next-Hop ToS」では、以前の NetFlow の「BGP Next-Hop ToS」の集約キャッシュと同じ key フィールドおよび nonkey フィールドが使用されます。

（注） この事前定義済みレコードは、IPv4 トラフィックの分析だけに使用できます。

次の表に、事前定義済みレコード「BGP Next-Hop ToS」で使用される key フィールドおよび nonkey フィールドを示します。

表 *8*：*Flexible NetFlow* の事前定義済みレコード「*BGP Next-Hop ToS*」で使用される *key* フィールドおよび *nonkey* フィールド

フィールド	**key** フィールドまたは **nonkey** フィールド	定義
IP ToS	Key	ToS フィールドの値。
IP Source autonomous system	Key	送信元 IP アドレスの自律システム（ピアまたはオリジン）。
IP Destination autonomous system	Key	宛先 IP アドレスの自律システム（ピアまたはオリジン）。
IPv4 Next Hop Address BGP	Key	BGP ネクスト ホップの IPv4 アドレス。
Interface Input	Key	トラフィックが受信されたインターフェイス。
Interface Output	Key	トラフィックが送信されたインターフェイス。
Flow Direction	Key	フローが監視されている方向。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「Destination Prefix」

Flexible NetFlow の事前定義済みレコード「Destination Prefix」では、送信先プレフィックス トラフィック フローデータに基づいてフローが作成されます。 Flexible NetFlow の事前定義済みレコード「Destination Prefix」では、以前の NetFlow の「Destination Prefix」の集約キャッシュと同じ key フィールドおよび nonkey フィールドが使用されます。

（注） この事前定義済みレコードは、IPv4 および IPv6 トラフィックの分析に使用できます。

次の表に、Flexible NetFlow の事前定義済みレコード「Destination Prefix」で使用される key フィールドおよび nonkey フィールドを示します。

表 *9*：*Flexible NetFlow* の事前定義済みレコード「*Destination Prefix*」で使用される *key* フィールドおよび *nonkey* フィールド

フィールド	key フィールドまたは nonkey フィールド	定義
IP Destination autonomous system	Key	宛先 IP アドレスの自律システム（ピアまたはオリジン）。
IPv4 or IPv6 Destination Prefix	Key	宛先プレフィックス マスクを使って AND 検索された宛先 IP アドレス。
IPv4 or IPv6 Destination Mask	Key	宛先プレフィックス内のビット数。
Interface Output	Key	トラフィックが送信されたインターフェイス。
Flow Direction	Key	フローが監視されている方向。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

フィールド	key フィールドまたは nonkey フィールド	定義
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「Destination Prefix ToS」

Flexible NetFlow の事前定義済みレコード「Destination Prefix ToS」では、送信先プレフィックス トラフィック フロー データおよび ToS トラフィック フロー データに基づいてフローが作成されます。 Flexible NetFlow の事前定義済みレコード「Destination Prefix ToS」では、以前の NetFlow の「Destination Prefix ToS」の集約キャッシュと同じ key フィールドおよび nonkey フィールドが使用されます。

この事前定義済みレコードは特に、NetFlow 対応デバイスを通過するネットワーク トラフィックの宛先を確認できるデータをキャプチャするために役立ちます。

（注） この事前定義済みレコードは、IPv4 トラフィックの分析だけに使用できます。

次の表に、Flexible NetFlow の事前定義済みレコード「Destination Prefix ToS」で使用される key フィールドおよび nonkey フィールドを示します。

表 *10*：*Flexible NetFlow* の事前定義済みレコード「*Destination Prefix ToS*」で使用される *key* フィールドおよび *nonkey* フィールド

フィールド	key フィールドまたは nonkey フィールド	定義
IP ToS	Key	ToS フィールドの値。
IP Destination autonomous system	Key	宛先 IP アドレスの自律システム（ピアまたはオリジン）。
IPv4 Destination Prefix	Key	宛先プレフィックス マスクを使って AND 検索された宛先 IP アドレス。
IPv4 Destination Mask	Key	宛先プレフィックス内のビット数。
Interface Output	Key	トラフィックが送信されたインターフェイス。

フィールド	key フィールドまたは nonkey フィールド	定義
Flow Direction	Key	フローが監視されている方向。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「Prefix」

Flexible NetFlow の事前定義済みレコード「Prefix」では、トラフィック フロー データの送信元プレフィックスおよび送信先プレフィックスに基づいてフローが作成されます。 Flexible NetFlow の事前定義済みレコード「Prefix」では、以前の NetFlow の「Prefix」の集約キャッシュと同じ key フィールドおよび nonkey フィールドが使用されます。

（注） この事前定義済みレコードは、IPv4 および IPv6 トラフィックの分析に使用できます。 IPv6 トラフィックの場合、最小プレフィックス マスク長は 0 ビットと見なされます。

次の表に、Flexible NetFlow の事前定義済みレコード「Prefix」で使用される key フィールドおよび nonkey フィールドを示します。

表 *11*：*Flexible NetFlow* の事前定義済みレコード「*Prefix*」で使用される *key* フィールドおよび *nonkey* フィールド

フィールド	key フィールドまたは nonkey フィールド	定義
IP Source autonomous system	Key	送信元 IP アドレスの自律システム（ピアまたはオリジン）。

フィールド	**key** フィールドまたは **nonkey** フィールド	定義
IP Destination autonomous system	Key	宛先 IP アドレスの自律システム（ピアまたはオリジン）。
IPv4 or IPv6 Source Prefix	Key	送信元 IP アドレスと送信元プレフィックスマスクの論理積。または、集約されたフローが属す送信元 IP アドレスのプレフィックス。
IPv4 or IPv6 Source Mask	Key	送信元プレフィックス内のビット数。
IPv4 or IPv6 Destination Prefix	Key	宛先プレフィックス マスクを使って AND 検索された宛先 IP アドレス。
IPv4 or IPv6 Destination Mask	Key	宛先プレフィックス内のビット数。
Interface Input	Key	トラフィックが受信されたインターフェイス。
Interface Output	Key	トラフィックが送信されたインターフェイス。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「Prefix Port」

Flexible NetFlow の事前定義済みレコード「Prefix Port」では、トラフィック フロー データの送信元プレフィックスとポートおよび送信先プレフィックスとポートに基づいてフローが作成されます。 Flexible NetFlow の事前定義済みレコード「Prefix Port」では、以前の NetFlow の「Prefix Port」の集約キャッシュと同じ key フィールドおよび nonkey フィールドが使用されます。

この事前定義済みレコードは特に、NetFlow 対応デバイスを通過するネットワーク トラフィックの送信元と宛先を確認できるデータをキャプチャするために役立ちます。

（注） この事前定義済みレコードは、IPv4 トラフィックの分析だけに使用できます。

次の表に、Flexible NetFlow の事前定義済みレコード「Prefix Port」で使用される key フィールドおよび nonkey フィールドを示します。

表 *12*：*Flexible NetFlow* の事前定義済みレコード「*Prefix Port*」で使用される *key* フィールドおよび *nonkey* フィールド

フィールド	key フィールドまたは nonkey フィールド	定義
IP ToS	Key	ToS フィールドの値。
IP Protocol	Key	IP プロトコル フィールドの値。
IPv4 Source Prefix	Key	送信元 IP アドレスと送信元プレフィックスマスクの論理積。または、集約されたフローが属す送信元 IP アドレスのプレフィックス。
IPv4 Source Mask	Key	送信元プレフィックス内のビット数。
IPv4 Destination Prefix	Key	宛先プレフィックス マスクを使って AND 検索された宛先 IP アドレス。
IPv4 Destination Mask	Key	宛先プレフィックス内のビット数。
Transport Source Port	Key	トランスポート層の送信元ポート フィールドの値。

フィールド	key フィールドまたは nonkey フィールド	定義
Transport Destination Port	Key	トランスポート層の宛先ポートフィールドの値。
Interface Input	Key	トラフィックが受信されたインターフェイス。
Interface Output	Key	トラフィックが送信されたインターフェイス。
Flow Direction	Key	フローが監視されている方向。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「Prefix ToS」

Flexible NetFlow の事前定義済みレコード「Prefix ToS」では、送信元プレフィックスおよび送信先プレフィックスと ToS トラフィック フロー データに基づいてフローが作成されます。 Flexible NetFlow の事前定義済みレコード「Prefix ToS」では、以前の NetFlow の「Prefix ToS」の集約キャッシュと同じ key フィールドおよび nonkey フィールドが使用されます。

この事前定義済みレコードは特に、NetFlow 対応デバイスを通過するネットワーク トラフィックの送信元と宛先を確認できるデータをキャプチャするために役立ちます。

（注） この事前定義済みレコードは、IPv4 トラフィックの分析だけに使用できます。

次の表に、Flexible NetFlow の事前定義済みレコード「Prefix ToS」で使用される key フィールドおよび nonkey フィールドを示します。

表 *13*：*Flexible NetFlow* の事前定義済みレコード「*Prefix ToS*」で使用される ***key*** フィールドおよび ***nonkey*** フィールド

フィールド	**key** フィールドまたは **nonkey** フィールド	定義
IP ToS	Key	ToS フィールドの値。
IP Source autonomous system	Key	送信元 IP アドレスの自律システム（ピアまたはオリジン）。
IP Destination autonomous system	Key	宛先 IP アドレスの自律システム（ピアまたはオリジン）。
IPv4 Source Prefix	Key	送信元 IP アドレスと送信元プレフィックスマスクの論理積。または、集約されたフローが属す送信元 IP アドレスのプレフィックス。
IPv4 Source Mask	Key	送信元プレフィックス内のビット数。
IPv4 Destination Prefix	Key	宛先プレフィックス マスクを使って AND 検索された宛先 IP アドレス。
IPv4 Destination Mask	Key	宛先プレフィックス内のビット数。
Interface Input	Key	トラフィックが受信されたインターフェイス。
Interface Output	Key	トラフィックが送信されたインターフェイス。
Flow Direction	Key	フローが監視されている方向。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。

フィールド	key フィールドまたは nonkey フィールド	定義
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「Protocol Port」

Flexible NetFlow の事前定義済みレコード「Protocol Port」では、トラフィック フロー データのプロトコルとポートに基づいてフローが作成されます。 Flexible NetFlow の事前定義済みレコード「Protocol Port」では、以前の NetFlow の「Protocol Port」の集約キャッシュと同じ key フィールドおよび nonkey フィールドが使用されます。

（注） この事前定義済みレコードは、IPv4 および IPv6 トラフィックの分析に使用できます。

次の表に、Flexible NetFlow の事前定義済みレコード「Protocol Port」で使用される key フィールドおよび nonkey フィールドを示します。

表 *14*：*Flexible NetFlow* の事前定義済みレコード「*Protocol Port*」で使用される *key* フィールドおよび *nonkey* フィールド

フィールド	key フィールドまたは nonkey フィールド	定義
IP Protocol	Key	IP プロトコル フィールドの値。
Transport Source Port	Key	トランスポート層の送信元ポート フィールドの値。
Transport Destination Port	Key	トランスポート層の宛先ポート フィールドの値。
Flow Direction	Key	フローが監視されている方向。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。

フィールド	key フィールドまたは nonkey フィールド	定義
Counter Packets	Nonkey	フロー内で認識されたパケット数。
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「Protocol Port ToS」

Flexible NetFlow の事前定義済みレコード「Protocol Port ToS」では、トラフィック データのプロトコル、ポート、および ToS 値に基づいてフローが作成されます。 Flexible NetFlow の事前定義済みレコード「Protocol Port ToS」では、以前の NetFlow の「Protocol Port ToS」の集約キャッシュと同じ key フィールドおよび nonkey フィールドが使用されます。

この事前定義済みレコードは、データをキャプチャし、トラフィック タイプごとのネットワーク使用状況を調べる場合に、特に役立ちます。

（注） この事前定義済みレコードは、IPv4 トラフィックの分析だけに使用できます。

次の表に、Flexible NetFlow の事前定義済みレコード「Protocol Port ToS」で使用される key フィールドおよび nonkey フィールドを示します。

表 *15*：*Flexible NetFlow* の事前定義済みレコード「*Protocol Port ToS*」で使用される *key* フィールドおよび *nonkey* フィールド

フィールド	key フィールドまたは nonkey フィールド	定義
IP ToS	Key	ToS フィールドの値。
IP Protocol	Key	IP プロトコル フィールドの値。
Transport Source Port	Key	トランスポート層の送信元ポート フィールドの値。

フィールド	key フィールドまたは nonkey フィールド	定義
Transport Destination Port	Key	トランスポート層の宛先ポートフィールドの値。
Flow Direction	Key	フローが監視されている方向。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「Source Prefix」

Flexible NetFlow の事前定義済みレコード「Source Prefix」では、ネットワーク トラフィックの送信元プレフィックスに基づいてフローが作成されます。 Flexible NetFlow の事前定義済みレコード「Source Prefix」では、以前の NetFlow の「Source Prefix」の集約キャッシュと同じ key フィールドおよび nonkey フィールドが使用されます。

（注） この事前定義済みレコードは、IPv4 および IPv6 トラフィックの分析に使用できます。

次の表に、Flexible NetFlow の事前定義済みレコード「Source Prefix」で使用される key フィールドおよび nonkey フィールドを示します。

表 *16*：*Flexible NetFlow* の事前定義済みレコード「*Source Prefix*」で使用される *key* フィールドおよび *nonkey* フィールド

フィールド	key フィールドまたは nonkey フィールド	定義
IP Source autonomous system	Key	送信元 IP アドレスの自律システム（ピアまたはオリジン）。

フィールド	key フィールドまたは nonkey フィールド	定義
IPv4 or IPv6 Source Prefix	Key	送信元 IP アドレスと送信元プレフィックスマスクの論理積。または、集約されたフローが属す送信元 IP アドレスのプレフィックス。
IPv4 or IPv6 Source Mask	Key	送信元プレフィックス内のビット数。
Interface Input	Key	トラフィックが受信されたインターフェイス。
Flow Direction	Key	フローが監視されている方向。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

事前定義済みレコード「Source Prefix ToS」

Flexible NetFlow の事前定義済みレコード「Source Prefix ToS」では、ネットワーク トラフィックの送信元プレフィックスと ToS 値に基づいてフローが作成されます。 Flexible NetFlow の事前定義済みレコード「Source Prefix ToS」では、以前の NetFlow の「Source Prefix ToS」の集約キャッシュと同じ key フィールドおよび nonkey フィールドが使用されます。

この事前定義済みレコードは特に、NetFlow 対応デバイスを通過するネットワーク トラフィックの送信元を確認できるデータをキャプチャするために役立ちます。

（注） この事前定義済みレコードは、IPv4 トラフィックの分析だけに使用できます。

次の表に、Flexible NetFlow の事前定義済みレコード「Source Prefix ToS」で使用される key フィールドおよび nonkey フィールドを示します。

表 *17*：*Flexible NetFlow* の事前定義済みレコード「*Source Prefix ToS*」で使用される *key* フィールドおよび *nonkey* フィールド

フィールド	**key** フィールドまたは **nonkey** フィールド	定義
IP ToS	Key	ToS フィールドの値。
IP Source autonomous system	Key	送信元 IP アドレスの自律システム（ピアまたはオリジン）。
IPv4 Source Prefix	Key	送信元 IP アドレスと送信元プレフィックスマスクの論理積。または、集約されたフローが属す送信元 IP アドレスのプレフィックス。
IPv4 Source Mask	Key	送信元プレフィックス内のビット数。
Interface Input	Key	トラフィックが受信されたインターフェイス。
Flow Direction	Key	フローが監視されている方向。
Counter Bytes	Nonkey	フロー内で認識されたバイト数。
Counter Packets	Nonkey	フロー内で認識されたパケット数。
Time Stamp System Uptime First	Nonkey	最初のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。
Time Stamp System Uptime Last	Nonkey	最後のパケットが交換されたときのシステム稼働時間（ミリ秒単位。このデバイスが最初にブートしてからの時間）。

Flexible NetFlow の設定方法

カスタマイズしたフロー レコードの設定

カスタム フロー レコードを設定するには、次の作業を実行します。

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 1 つ以上の **match** 基準が必須で、通常は nonkey フィールドとして使用する 1 つ以上の **collect** 基準があります。

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。この作業では、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。 必要に応じてこの作業の手順を変更し、要件に合わせてカスタム フロー レコードを作成します。

手順の概要

1. **enable**
2. **configure terminal**
3. **flow record** *record-name*
4. **description** *description*
5. **match** {**ipv4** | **ipv6**} {**destination** | **source**} **address**
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。
7. **collect interface** {**input** | **output**}
8. 必要に応じてステップ 7 を繰り返し、レコードの追加 nonkey フィールドを設定します。
9. **end**
10. **show flow record** *record-name*
11. **show running-config flow record** *record-name*

手順の詳細

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

	コマンドまたはアクション	目的
ステップ 3	**flow record** *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** *description* 例： `Device(config-flow-record)# description Used for basic traffic analysis`	（任意）フロー レコードの説明を作成します。
ステップ 5	**match** {**ipv4** | **ipv6**} {**destination** | **source**} **address** 例： `Device(config-flow-record)# match ipv4 destination address`	フロー レコードの key フィールドを設定します。 （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。 **match ipv4** コマンドで使用可能なその他の key フィールド、および key フィールドの設定に使用可能なその他の **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	**collect interface** {**input** | **output**} 例： `Device(config-flow-record)# collect interface input`	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。 （注） 次に、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する例を示します。 nonkey フィールドの設定に使用可能なその他の **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 8	必要に応じてステップ 7 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 9	**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。

	コマンドまたはアクション	目的
ステップ 10	**show flow record** *record-name* 例： `Device# show flow record FLOW_RECORD-1`	（任意）指定されたフロー レコードの現在のステータスを表示します。
ステップ 11	**show running-config flow record** *record-name* 例： `Device# show running-config flow record FLOW_RECORD-1`	（任意）指定されたフロー レコードの設定を表示します。

フロー レコードの現在のステータスの表示

フロー レコードの現在のステータスを表示するには、次のオプション作業を実行します。

手順の概要

1. **enable**
2. **show flow record**

手順の詳細

ステップ 1 **enable**

enable コマンドによって、特権 EXEC モードを開始します（プロンプトが表示されたらパスワードを入力します）。

例：

```
Device> enable
Device#
```

ステップ 2 **show flow record**

show flow record コマンドでは、指定するフロー モニタの現在のステータスを表示します。

例：

```
Device# show flow record

flow record FLOW-RECORD-2:
  Description:        Used for basic IPv6 traffic analysis
  No. of users:       1
  Total field space:  53 bytes
```

```
  Fields:
    match ipv6 destination address
    collect counter bytes
    collect counter packets
flow record FLOW-RECORD-1:
  Description:        Used for basic IPv4 traffic analysis
  No. of users:       1
  Total field space:  29 bytes
  Fields:
    match ipv4 destination address
    collect counter bytes
    collect counter packets
```

フロー レコード設定の確認

入力したコンフィギュレーション コマンドを確認するには、次のオプション作業を実行します。

手順の概要

1. **enable**
2. **show running-config flow record**

手順の詳細

ステップ 1 **enable**

enable コマンドによって、特権EXECモードを開始します（プロンプトが表示されたらパスワードを入力します）。

例：

```
Device> enable
Device#
```

ステップ 2 **show running-config flow record**

show running-config flow record コマンドでは、指定するフロー モニタのコンフィギュレーション コマンドを表示します。

例：

```
Device# show running-config flow record

Current configuration:
!
flow record FLOW-RECORD-2
 description Used for basic IPv6 traffic analysis
 match ipv6 destination address
 collect counter bytes
 collect counter packets
!
flow record FLOW-RECORD-1
 description Used for basic IPv4 traffic analysis
```

```
 match ipv4 destination address
 collect counter bytes
 collect counter packets
 collect timestamp sys-uptime first
 collect timestamp sys-uptime last
!
```

事前定義済みレコードを使用した IPv4 または IPv6 トラフィックのフロー モニタの設定

フロー モニタの Flexible NetFlow の事前定義済みレコード「NetFlow IPv4/IPv6 Original Input」を使用した IPv4/IPv6 トラフィックに対するフロー モニタを設定するには、次の必須作業を実行します。

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。 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にすることもできますが、上級のユーザであれば Flexible NetFlow フロー レコード コンフィギュレーション モードで **collect** および **match** コマンドを使用して独自のレコード フォーマットを作成することもできます。

（注） フローモニタのレコード フォーマットを **record** コマンドで変更するには、その前にフロー モニタを適用してある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。

手順の概要

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **description** *description*
5. **record netflow {ipv4 | ipv6} original-input**
6. **end**
7. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]][**statistics**]]
8. **show running-config flow monitor** *monitor-name*

手順の詳細

	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** *description* 例： `Device(config-flow-monitor)# description Used for monitoring IPv4 traffic`	（任意）フロー モニタの説明を作成します。
ステップ 5	**record netflow {ipv4 | ipv6} original-input** 例： `Device(config-flow-monitor)# record netflow ipv4 original-input`	フロー モニタのレコードを指定します。
ステップ 6	**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 7	**show flow monitor** [[**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。

	コマンドまたはアクション	目的
ステップ 8	**show running-config flow monitor** *monitor-name* 例： `Device# show flow monitor FLOW_MONITOR-1`	（任意）指定されたフロー モニタの設定を表示します。

フロー モニタ用のフロー エクスポータの設定

詳細な分析や保管を目的として、Flexible NetFlow によって収集されるデータをリモート システムにエクスポートするためにフローモニタ用のフローエクスポータを設定するには、次のオプション作業を実行します。

フロー エクスポータは、Flexible NetFlow で収集されるデータを NetFlow Collection Engine などのリモート システムへ送信するために使用されます。 エクスポータでは、トランスポート プロトコルとして UDP、エクスポート フォーマットとしてバージョン 9 が使用されます。

（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 アドレスのいずれかを使用して宛先にエクスポートできます。

手順の概要

1. **enable**
2. **configure terminal**
3. **flow exporter** *exporter-name*
4. **description** *description*
5. **destination** {*hostname* | *ip-address*} [**vrf** *vrf-name*]
6. **export-protocol** {**netflow-v5** | **netflow-v9** | **ipfix**}
7. **transport udp** *udp-port*
8. **exit**
9. **flow monitor** *flow-monitor-name*
10. **exporter** *exporter-name*
11. **end**
12. **show flow exporter** *exporter-name*
13. **show running-config flow exporter** *exporter-name*

手順の詳細

	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 exporter** *exporter-name* 例： `Device(config)# flow exporter EXPORTER-1`	フロー エクスポータを作成し、Flexible NetFlow 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。 • このコマンドでは、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 4	**description** *description* 例： `Device(config-flow-exporter)# description Exports to datacenter`	（任意）フロー エクスポータの説明を作成します。
ステップ 5	**destination** {*hostname* | *ip-address*} [**vrf** *vrf-name*] 例： `Device(config-flow-exporter)# destination 172.16.10.2`	エクスポータでデータを送信する宛先システムのホスト名または IP アドレスを指定します。 （注） IPv4 または IPv6 アドレスのいずれかを使用して宛先にエクスポートできます。
ステップ 6	**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 7	**transport udp** *udp-port* 例： `Device(config-flow-exporter)# transport udp 65`	トランスポート プロトコルとして UDP を設定し、エクスポートされる Flexible NetFlow トラフィックを宛先システムが待機する UDP 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 8	**exit** 例： `Device(config-flow-exporter)# exit`	Flexible NetFlow フロー エクスポータ コンフィギュレーション モードを終了して、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	**flow monitor** *flow-monitor-name* 例： `Device(config)# flow monitor FLOW-MONITOR-1`	事前に作成されたフローモニタに対して 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。
ステップ 10	**exporter** *exporter-name* 例： `Device(config-flow-monitor)# exporter EXPORTER-1`	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	**show flow exporter** *exporter-name* 例： `Device# show flow exporter FLOW_EXPORTER-1`	（任意）指定されたフローエクスポータの現在のステータスを表示します。
ステップ 13	**show running-config flow exporter** *exporter-name* 例： `Device<# show running-config flow exporter FLOW_EXPORTER-1`	（任意）指定されたフローエクスポータの設定を表示します。

カスタマイズしたフロー モニタの作成

カスタム フロー モニタを作成するには、次の必須作業を実行します。

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。

はじめる前に

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、すべてのインターフェイスから適用済みのフロー モニタを削除する必要があります。 **ip flow monitor** コマンドについては、『*Cisco IOS Flexible NetFlow Command Reference*』を参照してください。

手順の概要

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **description** *description*
5. **record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6**} *record* [**peer**]}
6. **cache** {**entries** *number* | **timeout** {**active** | **inactive** | **update**} *seconds* | **type** {**immediate** | **normal** | **permanent**}}
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。
8. **statistics packet protocol**
9. **statistics packet size**
10. **exporter** *exporter-name*
11. **end**
12. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]] [**statistics**]]
13. **show running-config flow monitor** *monitor-name*

手順の詳細

	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）。

	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	**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** *description* 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* | **type** {**immediate** | **normal** | **permanent**}} 例： `Device(config-flow-monitor)# cache type normal`	（任意）フロー モニタ キャッシュ パラメータ（タイムアウト値、キャッシュ エントリ数、キャッシュ タイプなど）を変更します。 • **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。
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。	—
ステップ 8	**statistics packet 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。

	コマンドまたはアクション	目的
ステップ 9	**statistics packet 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	**show flow monitor** [[**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	**show running-config flow monitor** *monitor-name* 例： `Device# show running-config flow monitor FLOW_MONITOR-1`	（任意）指定されたフロー モニタの設定を表示します。

フロー モニタの現在のステータスの表示

フロー モニタの現在のステータスを表示するには、次のオプション作業を実行します。

手順の概要

1. **enable**
2. **show flow monitor** *monitor-name*

手順の詳細

ステップ 1 **enable**

enable コマンドによって、特権 EXEC モードを開始します（プロンプトが表示されたらパスワードを入力します）。

例：

```
Device> enable
Device#
```

ステップ 2 **show flow monitor** *monitor-name*

show flow monitor コマンドでは、指定するフロー モニタの現在のステータスを表示します。

例：

```
Device# show flow monitor FLOW-MONITOR-1

Flow Monitor FLOW-MONITOR-1:
  Description:       Used for basic ipv4 traffic analysis
  Flow Record:       FLOW-RECORD-1
  Flow Exporter:     EXPORTER-1
  Cache:
    Type:              normal
    Status:            allocated
    Size:              1000 entries / 50052 bytes
    Inactive Timeout:  15 secs
    Active Timeout:    1800 secs
    Update Timeout:    1800 secs
```

フロー モニタ キャッシュ内のデータの表示

フロー モニタ キャッシュ内のデータを表示するには、次のオプション作業を実行します。

はじめる前に

フロー モニタ キャッシュ内のフローを表示するためには、NetFlow original レコードで定義された基準に適合するトラフィックを受信するインターフェイスに、入力フロー モニタを適用する必要があります。

手順の概要

1. **enable**
2. **show flow monitor name** *monitor-name* **cache format record**

手順の詳細

ステップ 1 **enable**

enable コマンドによって、特権EXECモードを開始します（プロンプトが表示されたらパスワードを入力します）。

例：

```
Device> enable
Device#
```

ステップ 2 **show flow monitor name** *monitor-name* **cache format record**

show flow monitor name *monitor-name* **cache format record** コマンド文字列では、フロー モニタのステータス、統計情報、およびキャッシュ内のフロー データを表示します。

例：

```
Device# show flow monitor name FLOW-MONITOR-1 cache format record

Cache type:                               Normal
  Cache size:                               1000
  Current entries:                             4
  High Watermark:                              4
  Flows added:                               101
  Flows aged:                                 97
    - Active timeout    (  1800 secs)          3
    - Inactive timeout (    15 secs)          94
    - Event aged                               0
    - Watermark aged                           0
    - Emergency aged                           0
IPV4 DESTINATION ADDRESS:  172.16.10.5
ipv4 source address:       10.10.11.1
trns source port:          25
trns destination port:     25
counter bytes:             72840
counter packets:           1821
IPV4 DESTINATION ADDRESS:  172.16.10.2
ipv4 source address:       10.10.10.2
trns source port:          20
trns destination port:     20
counter bytes:             3913860
counter packets:           7326
IPV4 DESTINATION ADDRESS:  172.16.10.200
ipv4 source address:       192.168.67.6
trns source port:          0
trns destination port:     3073
counter bytes:             51072
counter packets:           1824

Device# show flow monitor name FLOW-MONITOR-2 cache format record

Cache type:                               Normal
  Cache size:                               1000
  Current entries:                             2
  High Watermark:                              3
  Flows added:                                95
  Flows aged:                                 93
    - Active timeout    (  1800 secs)          0
    - Inactive timeout (    15 secs)          93
    - Event aged                               0
    - Watermark aged                           0
    - Emergency aged                           0
```

```
IPV6 DESTINATION ADDRESS:  2001:DB8:4:ABCD::2
ipv6 source address:       2001:DB8:1:ABCD::1
trns source port:          33572
trns destination port:     23
counter bytes:             19140
counter packets:           349
IPV6 DESTINATION ADDRESS:  FF02::9
ipv6 source address:       FE80::A8AA:BBFF:FEBB:CC03
trns source port:          521
trns destination port:     521
counter bytes:             92
counter packets:           1
```

フロー モニタ設定の確認

入力したコンフィギュレーション コマンドを確認するには、次のオプション作業を実行します。

手順の概要

1. **enable**
2. **show running-config flow monitor**

手順の詳細

ステップ 1 **enable**

enable コマンドによって、特権 EXEC モードを開始します（プロンプトが表示されたらパスワードを入力します）。

例：

```
Device> enable
Device#
```

ステップ 2 **show running-config flow monitor**

show running-config flow monitor コマンドでは、指定したフロー モニタのコンフィギュレーション コマンドを表示します。

例：

```
Device# show running-config flow monitor FLOW-MONITOR-1

Current configuration:
!
flow monitor FLOW-MONITOR-1
 description Used for basic ipv4 traffic analysis
 record FLOW-RECORD-1
 exporter EXPORTER-1
 cache entries 1000
!
```

インターフェイスへのフロー モニタの適用

フローモニタをアクティブ化する前に、1つ以上のインターフェイスに適用する必要があります。フロー モニタをアクティブ化するには、次の必須作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. {**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**}
5. ステップ 3 および 4 を繰り返して、トラフィックをモニタするデバイスの他のインターフェイスでフロー モニタをアクティブ化します。
6. **end**
7. **show flow interface** *type number*
8. **show flow monitor name** *monitor-name* **cache format record**

手順の詳細

	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	**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。

	コマンドまたはアクション	目的
ステップ 4	{**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**} 例： `Device(config-if)# ip flow monitor FLOW-MONITOR-1 input`	作成済みのフローモニタを、トラフィックの分析対象となるインターフェイスに割り当てることで、そのフロー モニタをアクティブにします。
ステップ 5	ステップ 3 および 4 を繰り返して、トラフィックをモニタするデバイスの他のインターフェイスでフロー モニタをアクティブ化します。	—
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。
ステップ 7	**show flow interface** *type number* 例： `Device# show flow interface GigabitEthernet 0/0/0`	指定されたインターフェイスの Flexible NetFlow のステータス（イネーブルまたはディセーブル）を表示します。
ステップ 8	**show flow monitor name** *monitor-name* **cache format record** 例： `Device# show flow monitor name FLOW_MONITOR-1 cache format record`	指定されたフローモニタのステータス、統計情報、およびキャッシュ内のフローデータを表示します。

インターフェイスで Flexible NetFlow がイネーブル化されていることの確認

インターフェイスで Flexible NetFlow がイネーブルになっていることを確認するには、次のオプション作業を実行します。

手順の概要

1. **enable**
2. **show flow interface** *type number*

手順の詳細

ステップ 1 **enable**

enable コマンドによって、特権EXECモードを開始します（プロンプトが表示されたらパスワードを入力します）。

例：

```
Device> enable
Device#
```

ステップ 2 **show flow interface** *type number*

show flow interface コマンドでは、インターフェイスで Flexible NetFlow がイネーブルになっていることを確認します。

例：

```
Device# show flow interface GigabitEthernet 0/0/0

Interface GigabitEthernet0/0/0
  FNF:  monitor:          FLOW-MONITOR-1
        direction:        Input
        traffic(ip):      on
  FNF:  monitor:          FLOW-MONITOR-2
        direction:        Input
        traffic(ipv6):    on
Device# show flow interface GigabitEthernet 1/0/0
Interface GigabitEthernet1/0/0
  FNF:  monitor:          FLOW-MONITOR-1
        direction:        Output
        traffic(ip):      on
  FNF:  monitor:          FLOW-MONITOR-2
        direction:        Input
        traffic(ipv6):    on
```

Flexible NetFlow の設定例

例：IPv4 トラフィック用の Flexible NetFlow 事前定義済みレコードの設定

次に、Flexible NetFlow の事前定義済みレコード「BGP ToS Next-Hop」を使用してフローモニタを設定し、IPv4 トラフィックをモニタする方法の例を示します。

この例は、グローバル コンフィギュレーション モードで開始します。

```
!
flow monitor FLOW-MONITOR-1
 record netflow ipv4 bgp-nexthop-tos
 exit
```

```
!
ip cef
!
interface Ethernet 0/0
 ip address 172.16.6.2 255.255.255.0
 ip flow monitor FLOW-MONITOR-1 input
!
```

例：IPv6 トラフィック用の Flexible NetFlow 事前定義済みレコードの設定

次に、Flexible NetFlow の事前定義済みレコード「Source Prefix」を使用してフロー モニタを設定し、IPv6 トラフィックをモニタする方法の例を示します。

この例は、グローバル コンフィギュレーション モードで開始します。

```
!
flow monitor FLOW-MONITOR-2
 record netflow ipv6 source-prefix
 exit
ip cef
ipv6 cef
!
interface GigabitEthernet 0/0/0
 ipv6 address 2001:DB8:2:ABCD::2/48
 ipv6 flow monitor FLOW-MONITOR-2 input
!
```

例：数が制限されたフローを使用する通常のフロー レコード キャッシュの設定

次に、ルータのすべてのインターフェイス上のタイプ オブ サービス（ToS）フィールドを監視するための設定例を示します。 この例は、**show flow monitor** コマンドを使用して、ルータで分析用の追加データを収集することを目的としているため、エクスポータは設定されていません。

この例は、グローバル コンフィギュレーション モードで開始します。

```
!
flow record QOS_RECORD
 description UD: Flow Record to monitor the use of TOS within this router/network
 match interface input
 match interface output
 match ipv4 tos
 collect counter packets
 collect counter bytes
 exit
!
flow monitor QOS_MONITOR
 description UD: Flow Monitor which watches the limited combinations of interface and TOS
 record QOS_RECORD
 cache type normal
 cache entries 8192   ! 2^5 (combos of interfaces) * 256 (values of TOS)
 exit
!
interface GigabitEthernet0/0/0
 ip flow monitor QOS_MONITOR input
 exit
!
interface GigabitEthernet0/1/0
```

```
 ip flow monitor QOS_MONITOR input
 exit
!
interface GigabitEthernet0/2/0
 ip flow monitor QOS_MONITOR input
 exit
!
```

show flow monitor コマンドでは、キャッシュの現在のステータスを表示します。

```
Router# show flow monitor QOS_MONITOR cache

  Cache type:            Normal
  Cache size:                              8192
  Current entries:                            2
  High Watermark:                             2
  Flows added:                                2
  Updates sent         (  1800 secs)          0
```

例：IPv6 トラフィックのモニタリング用のカスタム フロー レコード キャッシュの設定

次の例では、IPv6 トラフィック監視用のカスタム フロー レコード キャッシュを作成します。

この例は、グローバル コンフィギュレーション モードで開始します。

```
!
ip cef
ipv6 cef
!
flow record FLOW-RECORD-2
 description Used for basic IPv6 traffic analysis
 match ipv6 destination address
 collect counter bytes
 collect counter packets
!
flow monitor FLOW-MONITOR-2
 description Used for basic IPv6 traffic analysis
 record FLOW-RECORD-2
 cache entries 1000
!
interface GigabitEthernet0/0/0
 ipv6 address 2001:DB8:2:ABCD::2/48
 ipv6 flow monitor FLOW-MONITOR-2 input
!
interface GigabitEthernet1/0/0
 ipv6 address 2001:DB8:3:ABCD::1/48
 ipv6 flow monitor FLOW-MONITOR-2 output
!
```

例：数が制限されたフローを使用する永続的なフローレコードキャッシュの設定

次に、ルータのすべてのインターフェイス上のタイプ オブ サービス（ToS）フィールドを監視するための設定例を示します。 この例は、**show flow monitor** コマンドを使用して、ルータで分析用の追加データを収集することを目的としているため、エクスポータは設定されていません。

この例は、グローバル コンフィギュレーション モードで開始します。

```
!
ip cef
!
flow record QOS_RECORD
 description UD: Flow Record to monitor the use of TOS within this router/network
 match interface input
 match interface output
 match ipv4 tos
 collect counter packets
 collect counter bytes
 exit
!
flow monitor QOS_MONITOR
 description UD: Flow Monitor which watches the limited combinations of interface and TOS
 record QOS_RECORD
 cache type permanent
 cache entries 8192   ! 2^5 (combos of interfaces) * 256 (values of TOS)
 exit
!
interface ethernet0/0
 ip flow monitor QOS_MONITOR input
 exit
!
interface ethernet0/1
 ip flow monitor QOS_MONITOR input
 exit
!
interface ethernet0/2
 ip flow monitor QOS_MONITOR input
 exit
!
interface serial2/0
 ip flow monitor QOS_MONITOR input
 exit
!
interface serial2/1
 ip flow monitor QOS_MONITOR input
!
```

show flow monitor コマンドでは、キャッシュの現在のステータスを表示します。

```
Router# show flow monitor QOS_MONITOR cache
  Cache type:                               Permanent
  Cache size:                                    8192
  Current entries:                                  2
  High Watermark:                                   2
  Flows added:                                      2
  Updates sent         (  1800 secs)                0
```

例：IPv4およびIPv6トラフィック用のFlexible NetFlow Egress Accountingの設定

次に、IPv4 および IPv6 トラフィック用の Flexible NetFlow 出力アカウンティングを設定する方法の例を示します。

この例は、グローバル コンフィギュレーション モードで開始します。

```
!
flow record v4_r1
match ipv4 tos
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
```

```
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow record v6_r1
match ipv6 traffic-class
match ipv6 protocol
match ipv6 source address
match ipv6 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow monitor FLOW-MONITOR-1
 record v4_r1
 exit
!
!
flow monitor FLOW-MONITOR-2
 record v6_r1
 exit
!
ip cef
ipv6 cef
!
interface GigabitEthernet0/0/0
 ip address 172.16.6.2 255.255.255.0
 ipv6 address 2001:DB8:2:ABCD::2/48
 ip flow monitor FLOW-MONITOR-1 output
 ipv6 flow monitor FLOW-MONITOR-2 output
!
```

例：Flexible NetFlow サブインターフェイス サポートの設定

次の例は、IPv4 トラフィック用の Flexible NetFlow サブインターフェイス サポートを設定する方法を示しています。

この例は、グローバル コンフィギュレーション モードで開始します。

```
!
flow record v4_r1
match ipv4 tos
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow monitor FLOW-MONITOR-1
 record v4_r1
 exit
!
ip cef
!
interface Ethernet0/0.1
 ip address 172.16.6.2 255.255.255.0
 ip flow monitor FLOW-MONITOR-1 input
!
```

次に、IPv6 トラフィック用の NetFlow サブインターフェイス サポートをエミュレートするために Flexible NetFlow を設定する方法の例を示します。

この例は、グローバル コンフィギュレーション モードで開始します。

```
!
flow record v6_r1
match ipv6 traffic-class
match ipv6 protocol
match ipv6 source address
match ipv6 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
collect timestamp absolute first
collect timestamp absolute last
!
flow monitor FLOW-MONITOR-2
 record v6_r1
 exit
!
ip cef
ipv6 cef
!
interface Ethernet0/0.1
 ipv6 address 2001:DB8:2:ABCD::2/48
 ipv6 flow monitor FLOW-MONITOR-2 input
!
```

例：Flexible NetFlow での複数エクスポート先の設定

次の例は、Flexible NetFlow マルチ エクスポート先を設定する方法を示しています。

この例は、グローバル コンフィギュレーション モードで開始します。

```
!
flow exporter EXPORTER-1
 destination 172.16.10.2
 transport udp 90
 exit
!
flow exporter EXPORTER-2
 destination 172.16.10.3
 transport udp 90
 exit
!
flow monitor FLOW-MONITOR-1
 record netflow-original
 exporter EXPORTER-2
 exporter EXPORTER-1
 exit
!
ip cef
!
interface GigabitEthernet0/0/0
 ip address 172.16.6.2 255.255.255.0
 ip flow monitor FLOW-MONITOR-1 input
!
```

その他の関連資料

関連資料

関連項目	参照先
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』
Flexible NetFlow の概念情報および設定作業	『*Flexible NetFlow* コンフィギュレーション ガイド』
Flexible NetFlow コマンド	『*Cisco IOS Flexible NetFlow Command Reference*』

標準/RFC

標準	タイトル
この機能によりサポートされる新規または変更された標準/RFC はありません。	—

MIB

MIB	MIB のリンク
なし	選択したプラットフォーム、シスコ ソフトウェア リリース、およびフィーチャ セットの MIB を検索してダウンロードする場合は、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	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

Flexible NetFlow の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。 この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *18* : *Flexible NetFlow* の機能情報

機能名	リリース	機能情報
Flexible NetFlow	12.2(33)SRC 12.2(50)SY 12.4(9)T 15.0(1)SY 15.0(1)SY1 Cisco IOS XE Release 3.1S	

機能名	リリース	機能情報
		Flexible NetFlow が導入されました。 この機能のサポートは、Cisco 7200 シリーズ ルータ用として Cisco IOS Release 12.2(33)SRC で追加されました。 次のコマンドが導入または変更されました。**cache**（Flexible NetFlow）、**clear flow exporter**、**clear flow monitor**、**clear sampler**、**collect counter**、**collect flow**、**collect interface**、**collect ipv4**、**collect ipv4 destination**、**collect ipv4 fragmentation**、**collect ipv4 section**、**collect ipv4 source**、**collect ipv4 total-length**、**collect ipv4 ttl**、**collect routing**、**collect timestamp sys-uptime**、**collect transport**、**collect transport icmp ipv4**、**collect transport tcp**、**collect transport udp**、**debug flow exporter**、**debug flow monitor**、**debug flow record**、**debug sampler**、**description**（Flexible NetFlow）、**destination**、**dscp**（Flexible NetFlow）、**exporter**、**flow exporter**、**flow monitor**、**flow platform**、**flow record**、**ip flow monitor**、**match flow**、**match interface**（Flexible NetFlow）、**match ipv4**、**match ipv4 destination**、**match ipv4 fragmentation**、**match ipv4 section**、**match ipv4 source**、**match ipv4 total-length**、**match ipv4 ttl**、**match routing**、**match transport**、**match transport icmp ipv4**、**match transport tcp**、**match transport udp**、**mode**（Flexible NetFlow）、

機能名	リリース	機能情報
		option（Flexible NetFlow）、**record**、**sampler**、**show flow exporter**、**show flow interface**、**show flow monitor**、**show flow record**、**show sampler**、**source**（Flexible NetFlow）、**statistics packet**、**template data timeout**、**transport**（Flexible NetFlow）