



パケット キャプチャの設定

- [パケット キャプチャの前提条件 \(1 ページ\)](#)
- [パケット キャプチャの制約事項 \(2 ページ\)](#)
- [パケット キャプチャの概要 \(4 ページ\)](#)
- [パケット キャプチャの設定 \(16 ページ\)](#)
- [パケット キャプチャのモニタリング \(34 ページ\)](#)
- [その他の参考資料 \(52 ページ\)](#)

パケット キャプチャの前提条件

- パケット キャプチャは Cisco Catalyst 9500 シリーズ スイッチでサポートされています。
- Wireshark は Network Essentials または Network Advantage を実行しているスイッチのみでサポートされています。

組み込みパケット キャプチャ (EPC) のソフトウェア サブシステムは、その動作で CPU とメモリ リソースを消費します。さまざまなタイプの操作を行うために十分なシステム リソースを準備する必要があります。システム リソースを使用するためのガイドラインを以下の表に示します。

表 1: EPC サブシステムのシステム要件

システム リソース	要件
ハードウェア (Hardware)	CPU 利用率の要件は、プラットフォームによって異なります。
メモリ	パケット バッファは DRAM に保存されます。パケット バッファのサイズは、ユーザが指定します。
ディスクスペース	パケットは外部のデバイスにエクスポートできます。フラッシュ ディスクでの中間保管は必要ありません。

パケットキャプチャの制約事項

- Wireshark でのグローバルパケットキャプチャはサポートされていません。
- 表示フィルタは、Wireshark でサポートされています。
- Wireshark を設定するための CLI では、機能を EXEC モードからのみ実行する必要があります。通常は設定サブモードで発生するアクション（キャプチャポイントの定義など）は、代わりに EXEC モードから処理されます。すべての主要コマンドは NVGEN の対象ではなく、NSF と SSO のシナリオではスタンバイスーパーバイザに同期されません。
- インターフェイスの出力方向にキャプチャされたパケットは、rewrite (TTL、VLAN タグ、CoS、チェックサム、MAC アドレス、DSCP、precedent、UP などを含む) によって加えられる変更を反映しない場合があります。
- 入力および出力の両方のパケットの書き換え情報はキャプチャされません。
- ファイルサイズによる循環ファイル保存の制限はサポートされません。
- ファイル制限は、Network Essentials と Network Advantage のフラッシュのサイズに限定されます。
- Control and Provisioning of Wireless Access Points (CAPWAP) などのプロトコルのデコードは、Network Essentials と Network Advantage でサポートされています。
- Network Essentials および Network Advantage では、ファイルモードにおいて、パケットはエクスポートされずにファイルに書き込まれます。
- 組み込み型の Wireshark はサポートされていますが、次の制限があります。
 - キャプチャフィルタと表示フィルタはサポートされません。
 - アクティブなキャプチャの復号化は使用できません。
 - 出力形式は、以前のリリースとは異なります。
- 組み込みパケットキャプチャ (EPC) は、入力のマルチキャストパケットのみをキャプチャし、出力の複製パケットはキャプチャしません。

設定の制限

- 最大 8 つのキャプチャポイントを定義できますが、一度にアクティブにできるのは 1 つだけです。1 つ開始するには 1 つ停止する必要があります。
- VRF、管理ポート、プライベート VLAN はいずれも接続ポイントとして使用することはできません。
- Wireshark クラスマップでは、1 つの ACL (IPv4、IPv6、MAC) のみが許可されます。
- Wireshark は宛先 SPAN ポートでパケットをキャプチャできません。

- Wireshark は、キャプチャ ポイントにアタッチされる接続ポイント（インターフェイス）のいずれかが動作を停止するとキャプチャを停止します。たとえば、接続ポイントに関連付けられているデバイスがから切断された場合です。キャプチャを再開するには、手動で再起動する必要があります。
- CPU 注入されたパケットは、コントロールプレーンパケットと見なされます。したがって、これらのタイプのパケットはインターフェイスの出力キャプチャではキャプチャされません。
- MAC フィルタは、MAC アドレスに一致しても IP パケットをキャプチャしません。これは、すべてのインターフェイス（L2 スイッチポート、L3 ルーテッドポート）に適用されます。
- MAC ACL は、ARP などの非 IP パケットだけに使用されます。レイヤ 3 ポートまたは SVI ではサポートされません。
- MAC フィルタは、L3 インターフェイスと L2 パケット（ARP）をキャプチャすることはできません。
- IPv6 ベースの ACL は VACL ではサポートされません。
- レイヤ 2 EtherChannels はサポートされません。
- レイヤ 3 ポートチャネル サポートが使用できます。
- キャプチャがすでにアクティブである、または開始されている場合、キャプチャポイントパラメータを変更することはできません。
- ACL ロギングおよび Wireshark には互換性がありません。Wireshark はアクティブになると優先されます。任意のポートにロギング中の ACL にキャプチャされているものも含め、すべてのトラフィックが Wireshark にリダイレクトされます。Wireshark を開始する前に、ACL ロギングを非アクティブにすることをお勧めします。これを実行しないと、Wireshark のトラフィックは ACL ロギング トラフィックに汚染されます。
- Wireshark は floodblock によってドロップされるパケットをキャプチャしません。
- 同じポートの PACL および RACL の両方をキャプチャすると、1 つのコピーだけが CPU に送信されます。DTLS 暗号化 CAPWAP インターフェイスをキャプチャすると、暗号化されたものと復号化されたものの 2 つのコピーが Wireshark に送信されます。DTLS 暗号化 CAPWAP トラフィックを運ぶレイヤ 2 インターフェイスをキャプチャすると同じ動作が発生します。コア フィルタは外部 CAPWAP ヘッダーに基づいています。
- Cisco IOS XE Everest 16.5.1a 以降：
 - L3 ポート チャネルのサポートが追加されます。
 - 表示形式がマイナーチェンジされました。
 - cap ファイルのパケット数を表示する機能
 - キャプチャされたバッファをクリアすると、その内容とともにバッファも削除されます。パケット キャプチャがアクティブなときに実行することはできません。

- 追加の警告メッセージが、コントロールプレーンのキャプチャで表示されます。
- バッファモードでは、パケットの表示は停止後のみに許可されます。
- Network Essentials および Network Advantage では停止時にパケット統計情報が表示されます。
- pcap ファイルでキャプチャされたパケット数を問い合わせる機能。
- 表示が cap ファイルからの場合、packet-number を使用して指定されたパケットの詳細を表示できます。
- 表示フィルタは、ファイルモードで使用可能です。
- パケットキャプチャの統計情報（受信またはドロップされたパケットおよびバイト）は、キャプチャ中またはキャプチャ停止後のいずれかに表示できます。
- システムは、Wireshark でサポートされるように、pcap/cap ファイルの内容に関する統計情報について問い合わせることができます。
- パケットキャプチャセッションは、バッファのサイズに関係なく常にストリーミングモードです。ロックステップモードは使用できません。



警告 コントロールプレーンパケットは、レート制限とパフォーマンスへの影響はありません。コントロールプレーンパケットのキャプチャを制限するフィルタを使用してください。

- ユーザがスイッチポートからルーテッドポート（L2>L3）へ、またはその逆へインターフェイスを変更した場合、インターフェイスが再び起動したときに、そのキャプチャポイントを削除し、新しいファイルを作成する必要があります。キャプチャポイントの停止/開始が機能しません。
- ユーザがアクティブなキャプチャセッションで使用されたファイルを削除した場合、そのキャプチャセッションは新しいファイルを作成できないため、キャプチャされたすべてのパケットが失われます。ユーザは、キャプチャポイントを再起動する必要があります。

パケットキャプチャの概要

パケットキャプチャツールの概要

パケットキャプチャ機能は、オンボードのパケットキャプチャファシリティです。ネットワーク管理者がデバイスを出入りするかデバイスを通るパケットをキャプチャすることで、パケットをローカルで分析したり、Wireshark や Embedded Packet Capture (EPC) のようなツールを使用するオフライン分析に向けてパケットを保存してエクスポートしたりできるようにするものです。この機能は、デバイスがネットワークの管理と操作にアクティブに参加できるようにす

ることによって、ネットワーク操作を簡略化します。この機能は、パケットの形式に関する情報を収集することによって、トラブルシューティングを容易にします。また、アプリケーションの分析とセキュリティも容易にします。

Wireshark を使用する Embedded Packet Capture は、Network Essentials および Network Advantage ライセンスでサポートされています。

Wireshark について

Wireshark の概要

Wireshark は、複数のプロトコルをサポートし、テキストベース ユーザ インターフェイスで情報を提供するパケット アナライザ プログラムです。

Wireshark は、.pcap と呼ばれる既知の形式を使用してファイルへパケットをダンプし、個々のインターフェイスに対して適用されイネーブルになります。EXEC モードでインターフェイスを指定し、フィルタおよび他のパラメータも指定します。Wireshark アプリケーションは、**start** コマンドを入力した場合にだけ適用され、Wireshark が自動または手動でキャプチャを停止した場合にだけ削除されます。

キャプチャ ポイント

キャプチャ ポイントとは、Wireshark 機能の一元的なポリシー定義です。キャプチャ ポイントは、どのパケットをキャプチャするか、どこからキャプチャするか、キャプチャパケットに何を実行するか、およびいつ停止するかなど、Wireshark の特定のインスタンスに関連付けられたすべての特徴を説明します。キャプチャ ポイントは作成後に変更される場合があり、**start** コマンドを使用して明示的にアクティブ化しない限り、アクティブになりません。このプロセスは、キャプチャポイントのアクティブ化またはキャプチャポイントの開始といいます。キャプチャポイントは名前で識別され、手動または自動で非アクティブ化または停止する場合があります。

複数のキャプチャポイントを定義してできますが、一度にアクティブにできるのは1つだけです。1つ開始するには1つ停止する必要があります。

スタック構成のシステムの場合、キャプチャポイントはアクティブなメンバーによりアクティブ化されます。スイッチオーバーが発生すると、アクティブなすべてのパケット キャプチャセッションが終了し、再起動する必要があります。

接続ポイント

接続ポイントは、キャプチャ ポイントに関連付けられた論理パケットのプロセス パスのポイントです。接続ポイントはキャプチャポイントの属性です。接続ポイントに影響するパケットはキャプチャ ポイント フィルタに対してテストされます。一致するパケットはキャプチャポイントに関連する Wireshark インスタンスにコピーされ、送信されます。特定のキャプチャポイントを複数の接続ポイントに関連付けることができます。異なるタイプ接続ポイントの混合に制限はありません。一部の制限は、異なるタイプの添付ポイントを指定すると適用されま

す。接続ポイントは、常に双方向であるレイヤ 2 VLAN の接続ポイントを除き、方向性あり（入力/出力/両方）です。

スタック型システムの場合では、すべてのスタック メンバの接続ポイントに有効です。EPC は定義されたすべての接続ポイントからパケットをキャプチャします。ただし、これらのパケットはアクティブ メンバーでのみに処理されます。

フィルタ

フィルタは、Wireshark にコピーされ、渡されるキャプチャ ポイントの接続ポイントを通過するトラフィックのサブセットを識別し制限するキャプチャ ポイントの属性です。Wireshark で表示されるためには、パケットは接続ポイントと、キャプチャポイントに関連付けられたすべてのフィルタも通過する必要があります。

キャプチャ ポイントには以下のタイプのフィルタがあります。

- コア システム フィルタ：コア システム フィルタはハードウェアによって適用され、一致基準はハードウェアによって制限されます。このフィルタは、ハードウェア転送トラフィックが Wireshark の目的でソフトウェアにコピーするかどうかを決定します。
- キャプチャ フィルタ：キャプチャ フィルタは、Wireshark によって適用されます。一致基準は、コア システム フィルタによってサポートされるものよりも詳細に表示されます。コア フィルタを通過するが、キャプチャ フィルタに失敗するパケットは CPU/ソフトウェアにコピーされ、送信されますが、Wireshark プロセスによって廃棄されます。キャプチャ フィルタの構文は、表示フィルタの構文と同じです。



(注) Cisco Catalyst 9500 シリーズ スイッチ の Wireshark はキャプチャ フィルタの構文を使用しません。

- 表示フィルタ：表示フィルタは、Wireshark によって適用されます。その一致基準はキャプチャ フィルタと似ています。表示フィルタに失敗したパケットは表示されません。

コア システム フィルタ

クラス マップまたは ACL を使用して、または CLI を使用して明示的にコア システム フィルタの一致基準を指定できます。



(注) CAPWAP を接続ポイントとして指定すると、コア システム フィルタは使用されません。

一部のインストール済み環境では、承認プロセスが長い場合さらに遅延を引き起こす可能性がある の設定を変更する権限を取得する必要があります。これにより、ネットワーク管理者の機能がトラフィックの監視および分析に制限される場合があります。この状況に対処するため、Wireshark は、EXEC モード CLI から、コア システム フィルタ一致基準の明示的な仕様をサポートします。この対処方法の欠点は、指定できる一致基準が、クラスマップがサポートす

る対象の限定的なサブセットである (MAC、IP 送信元アドレスおよび宛先アドレス、イーサネット タイプ、IP プロトコル、および TCP/UDP の発信元および宛先ポートなど) ことです。

コンフィギュレーションモードを使用する場合は ACL を定義するか、クラス マップでそこへキャプチャ ポイントを参照させることができます。明示的かつ ACL ベースの一致基準がクラス マップとポリシー マップの作成に内部的に使用されます。

注：ACL およびクラス マップの設定はシステムの一部であり、Wireshark 機能の側面ではありません。

Display Filter

表示フィルタを使用すると、.pcap ファイルからデコードして表示するときに表示するパケットの集合をさらに絞り込むように Wireshark に指示できます。

アクション (Actions)

Wireshark はライブ トラフィックまたは前の既存 .pcap ファイルで呼び出すことができます。ライブ トラフィックに対して起動されたとき、その表示フィルタを通過するパケットに対して次の 4 種類の処理を実行できます。

- デコード、分析、保存のためにメモリ内バッファへキャプチャ
- .pcap ファイルへ保存
- デコードおよび表示
- 保存および表示

.pcap ファイルのみに対して起動された場合は、デコードと表示の処理だけが適用できます。

キャプチャ パケットのメモリ内のバッファへのストレージ

パケットは、メモリ内のキャプチャ バッファに格納して、後でデコード、分析、または .pcap ファイルへ保存できます。

キャプチャ バッファは線形モードまたは循環モードを選択できます。線形モードでは、バッファが上限に達すると、新しいパケットが廃棄されます。循環モードでは、バッファが上限に達すると、新しいパケットを格納するために最も古いパケットが廃棄されます。必要に応じてバッファをクリアすることもできますが、このモードは、ネットワーク トラフィックのデバッグに主に使用されます。ただし、これを削除せずに、バッファの内容をクリアだけすることはできません。これを有効にするためには、現行のキャプチャを停止し、キャプチャをもう一度再起動します。



(注) パケットをバッファ内に保存する複数のキャプチャがある場合、メモリ ロスを避けるため、新しいキャプチャを開始する前にバッファをクリアしてください。

.pcap ファイルにキャプチャされたパケットのストレージ



(注) Wireshark がスタック内のスイッチで使用される場合は、パケットキャプチャをアクティブスイッチに接続されたフラッシュまたは USB フラッシュ デバイスにのみ保存できます。

たとえば、flash1 がアクティブなスイッチに接続されており、flash2 がセカンダリ スwitchに接続されている場合、flash1 にのみパケット キャプチャを保存できます。

アクティブ スwitchに接続されたフラッシュまたは USB フラッシュ デバイス以外のデバイスにパケット キャプチャを保存しようとする、エラーが発生する場合があります。

Wireshark は .pcap ファイルにキャプチャされたパケットを保存できます。キャプチャ ファイルは次のストレージデバイスに配置可能です。

- オンボードフラッシュ ストレージ (flash:)
- USB ドライブ(usbflash0:)



(注) サポートされていないデバイスまたはアクティブなスイッチに接続されていないデバイスにパケット キャプチャを保存しようとする、エラーが発生する可能性があります。

Wireshark のキャプチャポイントを設定する場合は、ファイル名を関連付けることができます。キャプチャポイントをアクティブにすると、Wireshark は指定された名前で作成したファイルを作成し、パケットを書き込みます。キャプチャポイントの作成時にファイルがすでに存在する場合、Wireshark はファイルを上書きできるかどうかについて問い合わせます。キャプチャポイントの有効化時にファイルがすでに存在する場合、Wireshark は既存のファイルを上書きします。特定のファイル名には1つのキャプチャポイントのみ関連付けることができます。

Wireshark が書き込んでいるファイルシステムが一杯になると、Wireshark はファイルの一部のデータで失敗します。そのため、キャプチャセッションを開始する前に、ファイルシステムに十分な領域があることを確認する必要があります。

パケット全体ではなくセグメントのみを保持して、必要な記憶域を減らすことができます。通常、最初の 64 または 128 バイトを超える詳細は不要です。デフォルトの動作は、パケット全体の保存です。

ファイルシステムを処理し、ファイルシステムへの書き込みを行う際、パケットのドロップの発生を避けるため、Wireshark ではオプションでメモリ バッファを使用してパケットの到着時に一時的に保持できます。メモリ バッファのサイズは、キャプチャポイントが .pcap ファイルに関連付けられる際に指定できます。

パケットのデコードおよび表示

Wireshark はコンソールにパケットをデコードして表示できます。この機能は、ライブ トラフィックに適用されるキャプチャポイントと前の既存 .pcap ファイルに適用されるキャプチャポイントで使用可能です。



(注) パケットをデコードして表示すると、CPU への負荷が高くなる場合があります。

Wireshark は、幅広い種類のパケット形式に対してパケット詳細をデコードおよび表示できます。詳細は、**monitor capture name start** コマンドを以下のキーワード オプション付きですることにより表示されます。これにより、表示およびデコード モードが開始します。

- **brief** : パケットごとに 1 行表示します (デフォルト)。
- **detailed** : プロトコルがサポートされているすべてのパケットのすべてのフィールドをデコードして表示します。詳細モードでは、他の 2 種類のモードよりも多くの CPU が必要です。
- **(hexadecimal) dump** : パケットデータの 16 進ダンプおよび各パケットの印刷可能文字としてパケットごとに 1 行表示します。

capture コマンドをデコードおよび表示オプション付きで入力すると、Wireshark 出力が Cisco IOS に返され、変更なしでコンソールに表示されます。

ライブトラフィックの表示

Wireshark はコア システムからパケットのコピーを受信します。Wireshark は、表示フィルタを適用して、不要なパケットを破棄し、残りのパケットをデコードおよび表示します。

.pcap ファイルの表示

Wireshark は、以前に保存された .pcap ファイルからのパケットをデコードして表示し、選択的にパケットを表示するように表示フィルタに指示できます。

パケットのストレージおよび表示

機能的には、このモードは以前の 2 種類のモードの組み合わせです。Wireshark は指定された .pcap ファイルにパケットを保存し、これらをコンソールにデコードおよび表示します。ここではコア フィルタだけが該当します。

Wireshark キャプチャ ポイントのアクティブ化および非アクティブ化

Wireshark のキャプチャ ポイントが、接続ポイント、フィルタ、アクション、およびその他のオプションで定義された場合、Wireshark をアクティブにする必要があります。キャプチャ ポイントがアクティブになるまで、実際にパケットをキャプチャしません。

キャプチャポイントがアクティブになる前に、一部の機能性チェックが実行されます。キャプチャポイントは、コア システム フィルタと接続ポイントのどちらも定義されていない場合はアクティブにできません。これらの要件を満たしていないキャプチャポイントをアクティブ化しようとする、エラーが生成されます。

表示フィルタを、必要に応じて指定します。

Wireshark のキャプチャ ポイントはアクティブになると、複数の方法で非アクティブにできません。 .pcap ファイルにパケットを格納するだけのキャプチャ ポイントは手動で停止することも、また時間制限またはパケット制限付きで設定することもでき、その後でキャプチャ ポイントは自動的に停止します。

Wireshark のキャプチャ ポイントがアクティブになると、固定レート ポリサーがハードウェアに自動的に適用され、CPU が Wireshark によって指示されたパケットでフラッシングしないようになります。 レートポリサーの短所は、リソースが使用可能な場合でも、確立されたレートを超えて連続するパケットをキャプチャできないことです。

パケットキャプチャ設定レートは、1 秒あたり 1000 パケット (pps) です。 1000 pps の制限は、すべての接続ポイントの合計に適用されます。たとえば、3つの接続ポイントにキャプチャセッションがあれば、3つの接続ポイントすべてのレートの合計が 1000 pps にポリシングされます。



(注) ポリサーは、コントロールプレーンパケットキャプチャではサポートされていません。コントロールプレーンキャプチャ ポイントを有効化するときは、CPU があふれないよう慎重に行う必要があります。

Wireshark 機能

ここでは、Wireshark 機能が 環境でどのように動作するかについて説明します。

- ポートセキュリティおよび Wireshark が入力キャプチャに適用された場合でも、ポートセキュリティによってドロップされたパケットは Wireshark でキャプチャされます。ポートセキュリティが入力キャプチャに適用され、Wireshark が出力キャプチャに適用された場合、ポートセキュリティによってドロップされたパケットは Wireshark ではキャプチャされません。
- ダイナミック ARP インスペクション (DAI) によってドロップされたパケットは Wireshark ではキャプチャされません。
- STP ブロック ステートのポートが接続ポイントとして使用され、コア フィルタが一致する場合、Wireshark は、パケットがスイッチにドロップされる場合でもポートに入ってくるパケットをキャプチャします。
- 分類ベースのセキュリティ機能：入力分類ベースのセキュリティ機能によってドロップされたパケット (ACL および IPSG など) は同じ層の接続ポイントに接続する Wireshark キャプチャポイントでは検出されません。一方、出力分類ベースのセキュリティ機能によってドロップされたパケットは、同じ層の接続ポイントに接続されている Wireshark のキャプチャポイントでキャッチされます。論理モデルは、Wireshark の接続ポイントが、入力側のセキュリティ機能のルックアップ後、および出力側のセキュリティ機能のルックアップ前に発生することです。

入力では、パケットはレイヤ 2 ポート、VLAN、およびレイヤ 3 ポート/SVI を介して送信されます。出力では、パケットはレイヤ 3 ポート/SVI、VLAN、およびレイヤ 2 ポートを介して送信されます。接続ポイントがパケットがドロップされるポイントの前にある場

合、Wireshark はパケットをキャプチャします。これ以外の場合は、Wireshark はパケットをキャプチャしません。たとえば、入力方向のレイヤ2接続ポイントに接続される Wireshark のキャプチャポリシーはレイヤ3分類ベースのセキュリティ機能によってドロップされたパケットをキャプチャします。対照的に、出力方向のレイヤ3接続ポイントに接続する Wireshark のキャプチャポリシーは、レイヤ2分類ベースのセキュリティ機能によりドロップされたパケットをキャプチャします。

- ルーテッドポートおよびスイッチ仮想インターフェイス (SVIs) : SVI の出力から送信されるパケットは CPU で生成されるため、Wireshark は SVI の出力をキャプチャできません。これらのパケットをキャプチャするには、コントロールプレーンを接続ポイントとして含めます。
- VLAN : Cisco IOS リリース 16.1 以降、VLAN が Wireshark の接続ポイントとして使用されている場合、パケット キャプチャは、入力方向と出力方向の両方の L2 と L3 でサポートされます。
- リダイレクション機能 : 入力方向では、レイヤ3 (PBR および WCCP など) でリダイレクトされる機能トラフィックは、レイヤ3 の Wireshark の接続ポイントよりも論理的に後です。Wireshark は、後で別のレイヤ3 インターフェイスにリダイレクトされる可能性がある場合でも、これらのパケットをキャプチャします。対照的に、レイヤ3 によってリダイレクトされる出力機能 (出力 WCCP など) は論理的にレイヤ3接続ポイントの前にあり、Wireshark ではキャプチャされません。
- SPAN : Wireshark は、SPAN 宛先として設定されたインターフェイスでパケットをキャプチャできません。
- SPAN : Wireshark は、入力方向の SPAN 送信元として設定されたインターフェイスでパケットをキャプチャできます。出力方向でも使用できる可能性があります。
- ACL が適用されていない場合、最大 1000 の VLAN からパケットを一度にキャプチャできます。ACL が適用されている場合、Wireshark の使用できるハードウェア領域はより少なくなります。結果として、パケット キャプチャに一度に使用できる VLAN の最大数は低くなります。1000 以上の VLAN トンネルを一度に使用したり、ACL を多数使用すると予測されない結果が生じる可能性があります。たとえば、モビリティがダウンする可能性があります。



(注) 過剰な CPU 使用につながり、予測されないハードウェア動作の原因となる可能性があるため、過剰な数の接続ポイントを一度にキャプチャしないことを強くお勧めします。

Wireshark のガイドライン

- Wireshark でのパケット キャプチャ中に、ハードウェア転送が同時に発生します。
- Wireshark のキャプチャプロセスを開始する前に、CPU 使用率が妥当であり、十分なメモリ (少なくとも 200 MB) が使用可能であることを確認します。

- ストレージファイルにパケットを保存する予定の場合、Wireshark キャプチャプロセスを開始する前に十分なスペースが利用可能であることを確認してください。
- Wireshark のキャプチャ中の CPU 使用率は、設定された基準に一致するパケットの数と、一致したパケット用のアクション（ストア、デコードして表示、あるいはこの両方）によって異なります。
- 高 CPU 使用率および他の不要な条件を避けるため、可能な限りキャプチャを最小限に抑えてください（パケット、期間による制限）。
- パケット転送はハードウェアで通常実行されるため、パケットは、ソフトウェア処理のために CPU にコピーされません。Wireshark のパケットキャプチャの場合、パケットは CPU にコピーされ、配信されて、これが CPU 使用率の増加につながります。

CPU 使用率を高くしないようにするには、次の手順を実行します。

- 関連ポートだけに接続します。
 - 一致条件を表すにはクラス マップを使用し、二次的にアクセス リストを使用してください。いずれも実行可能でない場合は、明示的な、インラインフィルタを使用します。
 - フィルタ規則に正しく準拠させます。緩和されたのではなく制限的な ACL で、トラフィック タイプを（IPv4 のみなどに）制限して、不要なトラフィックを引き出します。
- パケット キャプチャを短い期間または小さなパケット番号に常に制限します。capture コマンドのパラメータにより、次を指定することができます。
 - Capture duration
 - キャプチャされたパケットの数
 - ファイルサイズ (File size)
 - パケットのセグメント サイズ
 - コアフィルタと一致するトラフィックが非常に少ないことが判明している場合は、制限なしでキャプチャセッションを実行します。
 - 次の場合に高い CPU（またはメモリ）使用率になる可能性があります。
 - キャプチャセッションをイネーブルにし長期間不在のままにして、予期しないトラフィックのバーストが起きた場合。
 - リング ファイルまたはキャプチャ バッファを使用してキャプチャセッションを起動して、長期間不在のままにすると、パフォーマンスまたはシステムヘルスの問題が引き起こされます。
 - キャプチャセッション中に、のパフォーマンスやヘルスに影響する可能性のある Wireshark による高い CPU 使用率およびメモリ消費がないか監視します。こうした状況が発生した場合、Wireshark セッションをすぐに停止します。

- 大きなファイルの .pcap ファイルからのパケットをデコードして表示することは避けてください。代わりに、PC に .pcap ファイルを転送し PC 上で Wireshark を実行します。
- Wireshark インスタンスは最大 8 個まで定義できます。 .pcap ファイルまたはキャプチャバッファからパケットをデコードして表示するアクティブな **show** コマンドは、1 個のインスタンスとしてカウントされます。ただし、アクティブにできるインスタンスは 1 つだけです。
- 実行中のキャプチャに関連付けられた ACL が変更された場合は常に、ACL 変更を有効にするにはキャプチャを再起動する必要があります。キャプチャを再起動しないと、変更前の元の ACL が継続して使用されます。
- パケット損失を防ぐには、次の点を考慮します。
 - ライブ パケットをキャプチャしている間は、CPU に負荷のかかる操作であるデコードと表示ではなく（特に **detailed** モードの場合）、保存のみを使用します（**display** オプションを指定しない場合）。
 - パケットをバッファ内に保存する複数のキャプチャがある場合、メモリ ロスを避けるため、新しいキャプチャを開始する前にバッファをクリアしてください。
 - デフォルト バッファ サイズを使用し、パケットが失われている場合、バッファ サイズを増加してパケットの喪失を防ぐことができます。
 - フラッシュ ディスクへの書き込みは、CPU に負荷のかかる操作であるため、キャプチャ レートが不十分な場合、バッファ キャプチャの使用をお勧めします。
 - Wireshark キャプチャセッションは 1000 pps のレートで常にストリーミングモードで動作します。
 - ストリーミング キャプチャ モードは約 1000 pps をサポートし、ロックステップモードは約 2Mbps（256 バイトパケットで測定）をサポートします。一致するトラフィック レートがこの値を超えると、パケット損失が発生する可能性があります。
- ストリーミング キャプチャ モードのレートは 1000 pps です。
- コンソール ウィンドウのライブ パケットをデコードして表示する場合は、Wireshark セッションが短いキャプチャ期間によって抑制されていることを確認します。

**警告**

期間制限がより長いまたはキャプチャ期間がない（**term len 0** コマンドを使用して **auto-more** サポートのない端末を使用した）Wireshark セッションでは、コンソールまたは端末が使用できなくなる場合があります。

- 高 CPU 使用率につながるライブ トラフィックのキャプチャに Wireshark を使用している場合、QoS ポリシーを一時的に適用して、キャプチャ プロセスが終了するまで実際のトラフィックを制限することを考慮してください。
- すべての Wireshark 関連のコマンドは EXEC モードで、コンフィギュレーション コマンドは、Wireshark にありません。

Wireshark CLI でアクセス リストまたはクラス マップを使用する必要がある場合は、コンフィギュレーション コマンドでアクセス リストおよびクラス マップを定義する必要があります。

- 特定の順序はキャプチャ ポイントを定義する場合には適用されません。CLI で許可されている任意の順序でキャプチャ ポイント パラメータを定義できます。Wireshark CLI では、単一行のパラメータ数に制限はありません。これはキャプチャ ポイントを定義するために必要なコマンドの数を制限します。
- 接続ポイントを除くすべてのパラメータは、単一の値を取ります。通常、コマンドを再入力することにより、値を新しいものに置き換えることができます。ユーザの確認後にシステムが新しい値を受け入れ、古い値を上書きします。コマンドの **no** 形式は、新しい値の入力には必要はありませんが、パラメータの削除には必要です。
- Wireshark では 1 つ以上の接続ポイントを指定することができます。複数の接続ポイントを追加するには、新しい接続ポイントでコマンドを再入力します。接続ポイントを削除するには、コマンドの **no** 使用します。接続ポイントとしてインターフェイス範囲を指定できます。たとえば、**monitor capture mycap interface gi 3/1 in** **monitor capture mycap interface GigabitEthernet1/0/1 in** を入力します。ここで、**interface gi 3/1 GigabitEthernet1/0/1** は接続ポイントです。

またインターフェイス **GigabitEthernet1/0/2** にも接続する必要がある場合、次のように、別の行で指定します。

monitor capture mycap interface GigabitEthernet1/0/2 in

- キャプチャがアクティブなときは、キャプチャに対する変更を行うことはできません。
- 実行する処理は、いずれのパラメータが必須であるかを決定します。Wireshark CLI では、**start** コマンドを入力する前に任意のパラメータを指定または変更することができます。**start** コマンドを入力すると、すべての必須パラメータが入力されたと判断した後にのみ Wireshark が開始します。
- キャプチャ ポイントの作成時にファイルがすでに存在する場合、Wireshark はファイルを上書きできるかどうかについて問い合わせます。キャプチャポイントの有効化時にファイルがすでに存在する場合、Wireshark は既存のファイルを上書きします。
- コア フィルタは明示的なフィルタ、アクセス リスト、またはクラス マップにできます。これらのタイプの新しいフィルタを指定すると、既存のものを置き換えます。



(注) コア フィルタは、CAPWAP トンネル インターフェイスをキャプチャポイントの接続ポイントとして使用している場合を除き、必須です。

- 明示的な **stop** コマンドを使用するか、**automore** モードに **q** を入力して、Wireshark のセッションを終了します。セッションは、期間やパケットキャプチャの制限などの停止の条件が満たされたときに、自動的に終了します。

- ドロップされたパケットはキャプチャの最後に表示されません。ただし、ドロップされたサイズ超過のパケット数のみが表示されます。

デフォルトの Wireshark の設定

次の表は、デフォルトの Wireshark の設定を示しています。

機能	デフォルト設定
時間 (Duration)	No limit
パケット (Packets)	No limit
パケット長	制限なし (フルパケット)
ファイルサイズ (File size)	No limit
リング ファイル ストレージ	なし
バッファのストレージ モード	線形

組み込みパケット キャプチャについて

組み込みパケット キャプチャの概要

組み込みパケットキャプチャ (EPC) は、パケットのトレースとトラブルシューティングに役立つ組み込みシステム管理機能を提供します。この機能を使用すると、ネットワーク管理者は、シスコ デバイスを出入りするか通過するデータ パケットをキャプチャできます。ネットワーク管理者は、キャプチャバッファサイズとタイプ (循環またはリニア) およびキャプチャする各パケットの最大バイト数を定義する場合があります。パケットキャプチャレートは、詳細な管理制御を使用してスロットリングできます。たとえば、アクセス コントロール リストを使用してキャプチャ対象パケットをフィルタリングするオプションや、最大パケットキャプチャレートまたはサンプリング間隔の指定などの詳細な定義を行うオプションが利用できません。

組み込みパケット キャプチャの利点

- デバイスで IPv4 および IPv6 パケットをキャプチャでき、MAC フィルタを使用したり、MAC アドレスをマッチさせたりして、非 IP パケットもキャプチャ可能。
- パケット キャプチャ ポイントをイネーブルにする拡張可能なインフラストラクチャキャプチャポイントは、パケットがキャプチャされ、バッファと関連付けられるトラフィック トランジット ポイントです。
- 外部ツールを使用した分析に適したパケットキャプチャファイル (PCAP) 形式でパケット キャプチャをエクスポートする機能。
- さまざまな詳細レベルでキャプチャされたデータ パケットをデコードする方法。

パケット データ キャプチャ

パケット データ キャプチャは、バッファに格納されるデータ パケットのキャプチャです。パケット データ キャプチャは、一意の名前とパラメータを入力することによって定義します。

こうしたキャプチャでは、次のアクションを実行できます。

- インターフェイスでのキャプチャのアクティブ化。
- キャプチャ ポイントへのアクセス コントロール リスト (ACL) やクラス マップの適用。



(注) Network Based Application Recognition (NBAR) と MAC スタイルのクラス マップは、サポートされていません。

- キャプチャの破棄。
- サイズやタイプなどのバッファ ストレージ パラメータの指定。サイズの範囲は 1 ~ 100 MB です。デフォルトのバッファは線形です。もう 1 つのバッファ オプションは循環です。
- プロトコル、IP アドレス、ポート アドレスに関する情報を含む一致基準の指定。

パケット キャプチャの設定

Wireshark の設定方法

Wireshark を設定するには、次の基本的な手順を実行します。

1. キャプチャ ポイントを定義します。
2. キャプチャ ポイントのパラメータを追加または変更します。
3. キャプチャ ポイントをアクティブ化または非アクティブ化します。
4. キャプチャ ポイントを今後使用しない場合は削除します。

キャプチャ ポイントの定義

この手順の例では、非常にシンプルなキャプチャ ポイントを定義します。必要に応じて、**monitor capture** コマンドの 1 つのインスタンスを使用してキャプチャ ポイントとそのすべてのパラメータを定義できます。



(注) 接続ポイント、キャプチャの方向、およびコア フィルタが機能するキャプチャ ポイントを持つよう定義する必要があります。

コア フィルタを定義する必要がないのは、CAPWAP トンネリング インターフェイスを使用してワイヤレス キャプチャ ポイントを定義する場合です。この場合、コア フィルタは定義しません。これは使用できません。

キャプチャ ポイントを定義するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <pre>Device> enable</pre>	<p>特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。</p>
ステップ 2	<p>monitor capture {<i>capture-name</i><i>interface-type</i><i>interface-id</i>} {interface control-plane} {in out both}</p> <p>例 :</p> <pre>Device# monitor capture mycap interface GigabitEthernet1/0/1 in</pre>	<p>キャプチャ ポイントを定義し、キャプチャ ポイントが関連付けられている接続ポイントを指定し、キャプチャの方向を指定します。</p> <p>キーワードの意味は次のとおりです。</p> <ul style="list-style-type: none"> • <i>capture-name</i> : 定義するキャプチャ ポイントの名前を指定します (例では mycap が使用されています)。キャプチャ名の長さは8文字以下にしてください。英数字、アンダースコア (_) のみが許可されます • (任意) interface <i>interface-type</i> <i>interface-id</i> : キャプチャ ポイントが関連付けられる接続ポイントを指定します (例では GigabitEthernet1/0/1 が使用されています)。

	コマンドまたはアクション	目的
		<p>(注) オプションで、このコマンドインスタンス1つでこのキャプチャポイントの複数の接続ポイントおよびパラメータすべてを定義できます。これらのパラメータについては、キャプチャポイントパラメータの変更に関する手順で説明されています。範囲のサポートは、接続ポイントを追加および削除するためにも使用できます。</p> <p><i>interface-type</i> には次のいずれかを使用します。</p> <ul style="list-style-type: none"> • GigabitEthernet : 接続ポイントを GigabitEthernet として指定します。 • vlan : 接続ポイントを VLAN として指定します。 <p>(注) このインターフェイスを接続ポイントとして使用する場合は、入力キャプチャのみが可能です。</p> <ul style="list-style-type: none"> • capwap : 接続ポイントを CAPWAP トンネルとして指定します。 <p>(注) このインターフェイスを接続ポイントとして使用すると、コアフィルタは使用できません。</p> <ul style="list-style-type: none"> • (任意) control-plane : 接続ポイントとしてコントロールプレーンを指定します。

	コマンドまたはアクション	目的
<p>ステップ 3</p>	<p>monitor capture {<i>capture-name</i>} [match {any ipv4 any any ipv6} any any}]</p> <p>例 :</p> <pre>Device# monitor capture mycap interface GigabitEthernet1/0/1 in match any</pre>	<p>• in out both : キャプチャの方向を指定します。</p> <p>コアシステムのフィルタを定義します。</p> <p>(注) コア フィルタが使用できなくなるため、CAPWAP のトンネリング インターフェイスを接続ポイントとして使用する場合はこの手順を実行しないでください。</p> <p>キーワードの意味は次のとおりです。</p> <ul style="list-style-type: none"> • capture-name : 定義するキャプチャポイントの名前を指定します (例では mycap が使用されています)。 • match : フィルタを指定します。定義されている最初のフィルタはコアフィルタです。 <p>(注) キャプチャ ポイントは、コアシステムフィルタと接続ポイントのどちらも定義されていない場合はアクティブにできません。これらの要件を満たしていないキャプチャ ポイントをアクティブ化しようとすると、エラーが生成されます。</p> <ul style="list-style-type: none"> • ipv4 : IP バージョン 4 のフィルタを指定します。 • ipv6 : IP バージョン 6 のフィルタを指定します。
<p>ステップ 4</p>	<p>show monitor capture {<i>capture-name</i>} [parameter]</p> <p>例 :</p> <pre>Device# show monitor capture mycap parameter monitor capture mycap interface GigabitEthernet1/0/1 in monitor capture mycap match any</pre>	<p>ステップ 2 で定義したキャプチャ ポイント パラメータを表示し、キャプチャポイントを定義したことを確認します。</p>

	コマンドまたはアクション	目的
ステップ 5	show capwap summary 例： Device# show capwap summary	ワイヤレス キャプチャの接続ポイントとして使用できる CAPWAP トンネルを表示します。 (注) このコマンドは、ワイヤレスキャプチャを実行するために CAPWAP トンネルを接続ポイントとして使用している場合にのみ使用します。例の項の CAPWAP の例を参照してください。
ステップ 6	show running-config 例： Device# show running-config	入力を確認します。
ステップ 7	copy running-config startup-config 例： Device# copy running-config startup-config	(任意) コンフィギュレーションファイルに設定を保存します。

例

CAPWAP 接続ポイントでキャプチャ ポイントを定義するには次を実行します。

```

Device# show capwap summary

CAPWAP Tunnels General Statistics:
  Number of Capwap Data Tunnels      = 1
  Number of Capwap Mobility Tunnels   = 0
  Number of Capwap Multicast Tunnels  = 0

Name  APName                               Type PhyPortIf Mode      McastIf
-----
Ca0   AP442b.03a9.6715                       data Gi3/0/6   unicast  -

Name  SrcIP          SrcPort  DestIP          DstPort  DtlsEn  MTU    Xact
-----
Ca0   10.10.14.32    5247     10.10.14.2     38514    No      1449   0

Device# monitor capture mycap interface capwap 0 both
Device# monitor capture mycap file location flash:mycap.pcap
Device# monitor capture mycap file buffer-size 1
Device# monitor capture mycap start

*Aug 20 11:02:21.983: %BUFCAP-6-ENABLE: Capture Point mycap enabled.on
    
```

```

Device# show monitor capture mycap parameter
  monitor capture mycap interface capwap 0 in
  monitor capture mycap interface capwap 0 out
  monitor capture mycap file location flash:mycap.pcap buffer-size 1
Device#
Device# show monitor capture mycap

Status Information for Capture mycap
Target Type:
Interface: CAPWAP,
  Ingress:
0
  Egress:
0
  Status : Active
Filter Details:
  Capture all packets
Buffer Details:
  Buffer Type: LINEAR (default)
File Details:
  Associated file name: flash:mycap.pcap
  Size of buffer(in MB): 1
Limit Details:
  Number of Packets to capture: 0 (no limit)
  Packet Capture duration: 0 (no limit)
  Packet Size to capture: 0 (no limit)
  Packets per second: 0 (no limit)
  Packet sampling rate: 0 (no sampling)
Device#
Device# show monitor capture file flash:mycap.pcap
  1  0.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....
  2  0.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....
  3  2.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....
  4  2.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....
  5  3.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....
  6  4.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....
  7  4.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....
  8  5.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....
  9  5.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....
 10  6.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....
 11  8.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....
 12  9.225986 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
 13  9.225986 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
 14  9.225986 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
 15  9.231998 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
 16  9.231998 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
 17  9.231998 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
 18  9.236987 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
 19 10.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....
 20 10.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....

```

```

21 12.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....
22 12.239993 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
23 12.244997 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
24 12.244997 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
25 12.250994 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
26 12.256990 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
27 12.262987 10.10.14.2 -> 10.10.14.32 DTLSv1.0 Application Data
28 12.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....
29 12.802012 10.10.14.3 -> 10.10.14.255 NBNS Name query NB WPAD.<00>
30 13.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=.....

```

次のタスク

さらなる接続ポイントを追加して、キャプチャポイントのパラメータを変更し、アクティブ化できます。または、キャプチャポイントをそのまま使用したい場合はすぐにアクティブ化することもできます。



(注) このトピックで説明されているメソッドを使用してキャプチャポイントのパラメータを変更することはできません。

ユーザが誤ったキャプチャ名、または無効/存在しない接続ポイントを入力すると、スイッチは、「Capture Name should be less than or equal to 8 characters.Only alphanumeric characters and underscore () is permitted」および「% Invalid input detected at '^' marker」のようなエラーを表示します。

キャプチャポイントパラメータの追加または変更

パラメータの値を指定する手順は、順番にリストされますが、任意の順序で実行できます。1行、2行、または複数行で指定できます。複数指定が可能な接続ポイントを除き、同じオプションを再定義することで、任意の値をより最近の値に置き換えることができます。すでに指定された特定のパラメータが変更されている場合は、インタラクティブに確認する必要があります。

キャプチャポイントのパラメータを変更するには、次の手順を実行します。

始める前に

以下の手順を実行する前にキャプチャポイントを定義する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例 : Device> enable	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	monitor capture { <i>capture-name</i> mac-match-string} match {any mac ipv4 {any host protocol}{any host} ipv6 {any host protocol}{any host}} 例 : Device# monitor capture mycap match ipv4 any any	明示的に、または ACL を介して、または クラス マップを介して定義されたコアシステムフィルタ (ipv4 any any) を定義します。
ステップ 3	monitor capture { <i>capture-name</i> secondssizenum} limit {[duration] [packet-length] [packets]} 例 : Device# monitor capture mycap limit duration 60 packet-len 400	秒単位のセッション制限 (60)、キャプチャされたパケット、または Wireshark によって保持されるパケットセグメント長 (400) を指定します。
ステップ 4	monitor capture { <i>capture-name</i> } file { <i>location filename</i> } 例 : Device# monitor capture mycap file location flash:mycap.pcap	キャプチャ ポイントがパケットを表示するだけでなくキャプチャできるようにする場合は、ファイルのアソシエーションを指定します。 (注) すでにファイルが存在する場合、それが上書きが可能かどうかを確認する必要があります。
ステップ 5	monitor capture { <i>capture-name</i> size} file { <i>buffer-size</i> } 例 : Device# monitor capture mycap file buffer-size 100	トラフィック バーストの処理に Wireshark で使用されるメモリ バッファのサイズを指定します。
ステップ 6	show monitor capture { <i>capture-name</i> } [parameter] 例 : Device# show monitor capture mycap parameter monitor capture mycap interface	以前に定義したキャプチャ ポイント パラメータを表示します。

	コマンドまたはアクション	目的
	<pre>GigabitEthernet1/0/1 in monitor capture mycap match ipv4 any any monitor capture mycap limit duration 60 packet-len 400 monitor capture point mycap file location bootdisk:mycap.pcap monitor capture mycap file buffer-size 100</pre>	
ステップ7	<p>end</p> <p>例 :</p> <p>Device(config)# end</p>	特権 EXEC モードに戻ります。

例

パラメータの変更

キャプチャ ファイルの関連付けまたは関連付け解除

```
Device# monitor capture point mycap file location flash:mycap.pcap
Device# no monitor capture mycap file
```

パケットバーストの処理にメモリバッファサイズを指定する

```
Device# monitor capture mycap buffer size 100
```

IPv4 と IPv6 の両方に一致するように、明示的なコア システム フィルタを定義する

```
Device# monitor capture mycap match any
```

次のタスク

キャプチャポイントに必要なパラメータがすべて含まれている場合はアクティブ化します。

キャプチャポイントパラメータの削除

順番に表示されていますが、パラメータを削除する手順は任意の順序で実行できます。1行、2行、または複数行で削除できます。複数可能な接続ポイントを除いて、任意のパラメータを削除できます。

キャプチャポイントのパラメータを削除するには、次の手順を実行します。

始める前に

キャプチャポイントパラメータは、以下の手順を使用して削除する前に定義する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <pre>Device> enable</pre>	<p>特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。</p>
ステップ 2	<p>no monitor capture {capture-name} match</p> <p>例 :</p> <pre>Device# no monitor capture mycap match</pre>	<p>キャプチャポイント (mycap) で定義されているすべてのフィルタを削除します。</p>
ステップ 3	<p>no monitor capture {capture-name} limit [duration] [packet-length] [packets]</p> <p>例 :</p> <pre>Device# no monitor capture mycap limit duration packet-len Device# no monitor capture mycap limit</pre>	<p>Wireshark によって保持されるセッション タイム制限およびパケット セグメント長を削除します。その他の指定された制限はそのままになります。</p> <p>Wireshark のすべての制限をクリアします。</p>
ステップ 4	<p>no monitor capture {capture-name} file [location] [buffer-size]</p> <p>例 :</p> <pre>Device# no monitor capture mycap file Device# no monitor capture mycap file location</pre>	<p>ファイルの関連付けを削除します。キャプチャポイントはパケットをキャプチャしなくなります。表示だけが実行されます。</p> <p>ファイル位置の関連付けを削除します。ファイル位置はキャプチャポイントとは関連付けられなくなります。ただし、他の定義されたファイル関連付けはこのアクションによっては影響を受けません。</p>
ステップ 5	<p>show monitor capture {capture-name} [parameter]</p> <p>例 :</p> <pre>Device# show monitor capture mycap parameter monitor capture mycap interface GigabitEthernet1/0/1 in</pre>	<p>パラメータの削除操作後にまだ定義されているキャプチャポイントパラメータを表示します。このコマンドは、キャプチャポイントと関連付けられるパラメータを確認するために手順の任意の地点で実行できます。</p>
ステップ 6	<p>end</p> <p>例 :</p> <pre>Device(config)# end</pre>	<p>特権 EXEC モードに戻ります。</p>

次のタスク

キャプチャポイントに必要なパラメータがすべて含まれている場合はアクティブ化します。



- (注) キャプチャポイントがアクティブなときにパラメータが削除されると、スイッチは「キャプチャがアクティブです (Capture is active)」というエラーを表示します。

キャプチャポイントの削除

キャプチャポイントを削除するには、次の手順を実行します。

始める前に

キャプチャポイントは、以下の手順を使用して削除する前に定義する必要があります。削除する前に、キャプチャポイントを停止する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	no monitor capture {capture-name} 例： Device# no monitor capture mycap	指定されたキャプチャポイント (mycap) を削除します。
ステップ 3	show monitor capture {capture-name} [parameter] 例： Device# show monitor capture mycap parameter Capture mycap does not exist	指定されたキャプチャポイントは削除されたため存在しないことを示すメッセージを表示します。
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 running-config 例：	入力を確認します。

	コマンドまたはアクション	目的
	Device# show running-config	
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。

次のタスク

削除したものと同名前の新規キャプチャ ポイントを定義できます。これらの手順は通常、キャプチャ ポイントの定義をやり直したい場合に実行します。

キャプチャ ポイントをアクティブまたは非アクティブにする

キャプチャ ポイントをアクティブまたは非アクティブにするには、次の手順を実行します。

始める前に

接続ポイントおよびコア システム フィルタが定義され、関連付けられたファイル名がすでに存在する場合でも、キャプチャ ポイントはアクティブ化することができます。このようなケースでは、既存のファイルは上書きされます。

関連するファイル名のないキャプチャ ポイントは、表示するためだけにアクティブにできます。ファイル名が指定されていないと、パケットはバッファに保管されます。ライブ表示 (キャプチャ時の表示) は、ファイルおよびバッファ モードの両方で使用できます。

表示フィルタを指定しない場合、パケットはライブ表示されず、コアシステムフィルタによってキャプチャされたすべてのパケットが表示されます。デフォルトの表示モードは **brief** です。



(注) CAPWAP のトンネリング インターフェイスを接続ポイントとして使用すると、コア フィルタは使用されないため、この場合は定義する必要はありません。

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。プロンプトが表示されたら、パスワードを入力します。

	コマンドまたはアクション	目的
ステップ 2	monitor capture {capture-name} start [display [display-filter <i>filter-string</i>]] [brief detailed dump] 例： Device# monitor capture mycap start display display-filter "stp"	キャプチャポイントをアクティブ化し、「stp」を含むパケットだけが表示されるように表示をフィルタします。
ステップ 3	monitor capture {capture-name} stop 例： Device# monitor capture name stop	キャプチャポイントを非アクティブにします。
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 running-config 例： Device# show running-config	入力を確認します。
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。

次のタスク

キャプチャポイントをアクティブおよび非アクティブにする際に、いくつかのエラーが発生する可能性があります。次に、発生する可能性のあるエラーのいくつかの例を示します。

アクティブ化する際に接続ポイントが不明

```
Switch#monitor capture mycap match any
Switch#monitor capture mycap start
No Target is attached to capture failed to disable provision featurefailed to remove
policyfailed to disable provision featurefailed to remove policyfailed to disable provision
featurefailed to remove policy
Capture statistics collected at software (Buffer):
Capture duration - 0 seconds
Packets received - 0
Packets dropped - 0
Packets oversized - 0

Unable to activate Capture.
```

```
Switch# unable to get action unable to get action unable to get action
Switch#monitor capture mycap interface g1/0/1 both
Switch#monitor capture mycap start
Switch#
*Nov 5 12:33:43.906: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
```

アクティブ化する際にフィルタが不明

```
Switch#monitor capture mycap int g1/0/1 both
Switch#monitor capture mycap start
Filter not attached to capture
Capture statistics collected at software (Buffer):
  Capture duration - 0 seconds
  Packets received - 0
  Packets dropped - 0
  Packets oversized - 0
```

```
Unable to activate Capture.
Switch#monitor capture mycap match any
Switch#monitor capture mycap start
Switch#
*Nov 5 12:35:37.200: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
```

キャプチャポイントがすでにアクティブ化されているのに、別のキャプチャポイントをアクティブ化しようとする

```
Switch#monitor capture mycap start
PD start invoked while previous run is active Failed to start capture : Wireshark operation failure
Unable to activate Capture.
Switch#show monitor capture
```

```
Status Information for Capture test
Target Type:
Interface: GigabitEthernet1/0/13, Direction: both
Interface: GigabitEthernet1/0/14, Direction: both
Status : Active
Filter Details:
Capture all packets
Buffer Details:
Buffer Type: LINEAR (default)
Buffer Size (in MB): 10
File Details:
Associated file name: flash:cchh.pcap
Limit Details:
Number of Packets to capture: 0 (no limit)
Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Maximum number of packets to capture per second: 1000
Packet sampling rate: 0 (no sampling)
```

```
Status Information for Capture mycap
Target Type:
Interface: GigabitEthernet1/0/1, Direction: both
Status : Inactive
Filter Details:
Capture all packets
Buffer Details:
Buffer Type: LINEAR (default)
Buffer Size (in MB): 10
File Details:
File not associated
Limit Details:
Number of Packets to capture: 0 (no limit)
```

```

Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Maximum number of packets to capture per second: 1000
Packet sampling rate: 0 (no sampling)
Switch#monitor capture test stop
Capture statistics collected at software (Buffer & Wireshark):
  Capture duration - 157 seconds
  Packets received - 0
  Packets dropped - 0
  Packets oversized - 0

Switch#
*Nov 5 13:18:17.406: %BUFCAP-6-DISABLE: Capture Point test disabled.
Switch#monitor capture mycap start
Switch#
*Nov 5 13:18:22.664: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
Switch#
    
```

キャプチャポイントバッファのクリア

次の手順に従ってバッファコンテンツをクリアするか、外部ファイルにストレージとして保存します。



(注) パケットをバッファ内に保存する複数のキャプチャがある場合、メモリロスを避けるため、新しいキャプチャを開始する前にバッファをクリアしてください。アクティブなキャプチャポイントのバッファをクリアしないでください。

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。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	monitor capture {capture-name} [clear export filename] 例： Device# monitor capture mycap clear	clear : 完全にバッファを削除します。 (注) clear コマンドを実行すると、 <ul style="list-style-type: none"> • Network Essentials ライセンスでは、このコマンドはバッファを削除せずにバッファの内容をクリアします。 • 他のすべてのライセンスでは、このコマンドはバッファ自体を削除します。

	コマンドまたはアクション	目的
		export : バッファでキャプチャされたパケットを保存し、バッファを削除します。
ステップ 3	end 例 : Device(config)# end	特権 EXEC モードに戻ります。
ステップ 4	show running-config 例 : Device# show running-config	入力を確認します。
ステップ 5	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。

例 : キャプチャポイントバッファの処理

キャプチャのファイルへのエクスポート

```
Device# monitor capture mycap export flash:mycap.pcap
```

Storage configured as File for this capture

キャプチャポイントバッファのクリア

```
Device# monitor capture mycap clear
```

Capture configured with file options

次のタスク



(注) Network Essentials 以外のライセンスでキャプチャポイントのバッファをクリアしようとする
と、スイッチは「Failed to clear capture buffer : Capture Buffer BUSY」エラーを表示します。

組み込みパケットキャプチャの実装方法

パケットデータキャプチャの管理



(注) アクティブなキャプチャポイントのエクスポートは、Network Essentials のみでサポートされています。他のすべてのライセンスでは、まずキャプチャを停止してからエクスポートをする必要があります。

バッファモードでパケットデータキャプチャを管理するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	monitor capture capture-name access-list access-list-name 例： Device# monitor capture mycap access- list v4acl	アクセスリストをパケットキャプチャのコアフィルタとして指定し、モニタキャプチャを設定します。
ステップ 3	monitor capture capture-name limit duration seconds 例： Device# monitor capture mycap limit duration 1000	モニタキャプチャの制限を設定します。
ステップ 4	monitor capture capture-name interface interface-name both 例： Device# monitor capture mycap interface GigabitEthernet 0/0/1 both	接続ポイントおよびパケットフロー方向を指定して、モニタキャプチャを設定します。
ステップ 5	monitor capture capture-name buffer circular size bytes 例：	パケットデータをキャプチャするようにバッファを設定します。

	コマンドまたはアクション	目的
	Device# monitor capture mycap buffer circular size 10	
ステップ 6	monitor capture capture-name start 例 : Device# monitor capture mycap start	トラフィック トレース ポイントでパケットデータのバッファへのキャプチャを開始します。
ステップ 7	monitor capture capture-name stop 例 : Device# monitor capture mycap stop	トラフィック トレース ポイントでパケットデータのキャプチャを停止します。
ステップ 8	monitor capture capture-name export file-location/file-name 例 : Device# monitor capture mycap export tftp://10.1.88.9/mycap.pcap	分析のためにキャプチャされたデータをエクスポートします。
ステップ 9	end 例 : Device# end	特権 EXEC モードに戻ります。

キャプチャされたデータのモニタリングとメンテナンス

キャプチャされたパケットデータのモニタリングとメンテナンスを行うには、次の作業を実行します。キャプチャ バッファの詳細とキャプチャ ポイントの詳細を表示します。

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。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	show monitor capture capture-buffer-name buffer dump 例 : Device# show monitor capture mycap buffer dump	(任意) キャプチャ パケットの 16 進数 ダンプおよびそのメタデータを表示します。

	コマンドまたはアクション	目的
ステップ 3	show monitor capture capture-buffer-name parameter 例： Device# show monitor capture mycap parameter	(任意) キャプチャを指定するために使用されたコマンドのリストを表示します。
ステップ 4	debug epc capture-point 例： Device# debug epc capture-point	(任意) パケットキャプチャポイントのデバッグをイネーブルにします。
ステップ 5	debug epc provision 例： Device# debug epc provision	(任意) パケットキャプチャプロビジョニングのデバッグをイネーブルにします。
ステップ 6	end 例： Device(config)# end	特権 EXEC モードに戻ります。

パケットキャプチャのモニタリング

Wireshark の設定例

例：.pcap ファイルからの概要出力の表示

次のように入力して、.pcap ファイルからの出力を表示できます。

```
Device# show monitor capture file flash:mycap.pcap brief
Starting the packet display ..... Press Ctrl + Shift + 6 to exit

  1 0.000000000 10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x002e,
seq=0/0, ttl=254
  2 0.000051000 10.10.10.1 -> 10.10.10.2  ICMP 114 Echo (ping) reply   id=0x002e,
seq=0/0, ttl=255 (request in 1)
  3 0.000908000 10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x002e,
seq=1/256, ttl=254
  4 0.001782000 10.10.10.1 -> 10.10.10.2  ICMP 114 Echo (ping) reply   id=0x002e,
seq=1/256, ttl=255 (request in 3)
  5 0.002961000 10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x002e,
seq=2/512, ttl=254
  6 0.003676000 10.10.10.1 -> 10.10.10.2  ICMP 114 Echo (ping) reply   id=0x002e,
seq=2/512, ttl=255 (request in 5)
```

```

 7 0.004835000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x002e,
seq=3/768, ttl=254
 8 0.005579000  10.10.10.1 -> 10.10.10.2  ICMP 114 Echo (ping) reply   id=0x002e,
seq=3/768, ttl=255 (request in 7)
 9 0.006850000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x002e,
seq=4/1024, ttl=254
10 0.007586000  10.10.10.1 -> 10.10.10.2  ICMP 114 Echo (ping) reply   id=0x002e,
seq=4/1024, ttl=255 (request in 9)
11 0.008768000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x002e,
seq=5/1280, ttl=254
12 0.009497000  10.10.10.1 -> 10.10.10.2  ICMP 114 Echo (ping) reply   id=0x002e,
seq=5/1280, ttl=255 (request in 11)
13 0.010695000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x002e,
seq=6/1536, ttl=254
14 0.011427000  10.10.10.1 -> 10.10.10.2  ICMP 114 Echo (ping) reply   id=0x002e,
seq=6/1536, ttl=255 (request in 13)
15 0.012728000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x002e,
seq=7/1792, ttl=254
16 0.013458000  10.10.10.1 -> 10.10.10.2  ICMP 114 Echo (ping) reply   id=0x002e,
seq=7/1792, ttl=255 (request in 15)
17 0.014652000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x002e,
seq=8/2048, ttl=254
18 0.015394000  10.10.10.1 -> 10.10.10.2  ICMP 114 Echo (ping) reply   id=0x002e,
seq=8/2048, ttl=255 (request in 17)
19 0.016682000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x002e,
seq=9/2304, ttl=254
20 0.017439000  10.10.10.1 -> 10.10.10.2  ICMP 114 Echo (ping) reply   id=0x002e,
seq=9/2304, ttl=255 (request in 19)
21 0.018655000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x002e,
seq=10/2560, ttl=254
22 0.019385000  10.10.10.1 -> 10.10.10.2  ICMP 114 Echo (ping) reply   id=0x002e,
seq=10/2560, ttl=255 (request in 21)
23 0.020575000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x002e,
seq=11/2816, ttl=254
--More<

```

例：.pcap ファイルからの詳細出力の表示

次のように入力して、.pcap ファイルの出力詳細を表示できます。

```

Device# show monitor capture file flash:mycap.pcap detailed
Starting the packet display ..... Press Ctrl + Shift + 6 to exit

Frame 1: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0
  Interface id: 0
  Encapsulation type: Ethernet (1)
  Arrival Time: Nov  6, 2015 11:44:48.322497000 UTC
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1446810288.322497000 seconds
  [Time delta from previous captured frame: 0.000000000 seconds]
  [Time delta from previous displayed frame: 0.000000000 seconds]
  [Time since reference or first frame: 0.000000000 seconds]
  Frame Number: 1
  Frame Length: 114 bytes (912 bits)
  Capture Length: 114 bytes (912 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ip:icmp:data]
Ethernet II, Src: Cisco_f3:63:46 (00:e1:6d:f3:63:46), Dst: Cisco_31:f1:c6
(00:e1:6d:31:f1:c6)
  Destination: Cisco_31:f1:c6 (00:e1:6d:31:f1:c6)

```

例 : .pcap ファイルからパケット ダンプ出力の表示

```

Address: Cisco_31:f1:c6 (00:e1:6d:31:f1:c6)
.... ..0. .... .. = LG bit: Globally unique address (factory default)

.... ..0 .... .. = IG bit: Individual address (unicast)
Source: Cisco_f3:63:46 (00:e1:6d:f3:63:46)
Address: Cisco_f3:63:46 (00:e1:6d:f3:63:46)
.... ..0. .... .. = LG bit: Globally unique address (factory default)

.... ..0 .... .. = IG bit: Individual address (unicast)
Type: IP (0x0800)
Internet Protocol Version 4, Src: 10.10.10.2 (10.10.10.2), Dst: 10.10.10.1 (10.10.10.1)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not
ECN-Capable Transport))
    0000 00.. = Differentiated Services Codepoint: Default (0x00)
    .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport)
(0x00)
Total Length: 100
Identification: 0x04ba (1210)
Flags: 0x00
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
Fragment offset: 0
Time to live: 254
Protocol: ICMP (1)
Header checksum: 0x8fc8 [validation disabled]
    [Good: False]
    [Bad: False]
Source: 10.10.10.2 (10.10.10.2)
Destination: 10.10.10.1 (10.10.10.1)
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0xe4db [correct]
Identifier (BE): 46 (0x002e)
Identifier (LE): 11776 (0x2e00)
Sequence number (BE): 0 (0x0000)
Sequence number (LE): 0 (0x0000)
Data (72 bytes)

0000 00 00 00 00 09 c9 8f 77 ab cd ab cd ab cd ab cd .....w.....
0010 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0020 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0030 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0040 ab cd ab cd ab cd ab cd .....
    Data: 0000000009c98f77abcdabcdabcdabcdabcdabcdabcd...
    [Length: 72]

Frame 2: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0
Interface id: 0

```

例 : .pcap ファイルからパケット ダンプ出力の表示

次のように入力して、パケット ダンプの出力を表示できます。

```

Device# show monitor capture file flash:mycap.pcap dump
Starting the packet display ..... Press Ctrl + Shift + 6 to exit

0000 00 e1 6d 31 f1 c6 00 e1 6d f3 63 46 08 00 45 00 ..m1....m.cF...E.
0010 00 64 04 ba 00 00 fe 01 8f c8 0a 0a 0a 02 0a 0a .d.....

```

```

0020 0a 01 08 00 e4 db 00 2e 00 00 00 00 00 09 c9 .....
0030 8f 77 ab cd ab cd ab cd ab cd ab cd ab cd ab cd .w.....
0040 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0050 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0060 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0070 ab cd ..

0000 00 e1 6d 31 f1 80 00 e1 6d 31 f1 80 08 00 45 00 ..m1....m1....E.
0010 00 64 04 ba 00 00 ff 01 8e c8 0a 0a 0a 01 0a 0a .d.....
0020 0a 02 00 00 ec db 00 2e 00 00 00 00 00 09 c9 .....
0030 8f 77 ab cd ab cd ab cd ab cd ab cd ab cd ab cd .w.....
0040 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0050 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0060 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0070 ab cd ..

0000 00 e1 6d 31 f1 c6 00 e1 6d f3 63 46 08 00 45 00 ..m1....m.cF..E.
0010 00 64 04 bb 00 00 fe 01 8f c7 0a 0a 0a 02 0a 0a .d.....
0020 0a 01 08 00 e4 d7 00 2e 00 01 00 00 00 09 c9 .....
0030 8f 7a ab cd ab cd ab cd ab cd ab cd ab cd ab cd .z.....
0040 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....

```

例：表示フィルタを使用した .pcap ファイルからのパケットの表示

次のように入力して、出力された .pcap ファイルのパケットを表示できます。

```

Device# show monitor capture file flash:mycap.pcap display-filter "ip.src == 10.10.10.2"
brief
Starting the packet display ..... Press Ctrl + Shift + 6 to exit

  1 0.000000000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,
seq=0/0, ttl=254
  3 0.000908000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,
seq=1/256, ttl=254
  5 0.002961000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,
seq=2/512, ttl=254
  7 0.004835000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,
seq=3/768, ttl=254
  9 0.006850000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,
seq=4/1024, ttl=254
 11 0.008768000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,
seq=5/1280, ttl=254
 13 0.010695000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,
seq=6/1536, ttl=254
 15 0.012728000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,
seq=7/1792, ttl=254
 17 0.014652000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,
seq=8/2048, ttl=254
 19 0.016682000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,
seq=9/2304, ttl=254
 21 0.018655000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,
seq=10/2560, ttl=254
 23 0.020575000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x002e,
seq=11/2816, ttl=254

```

例：.pcap ファイルにキャプチャされたパケットの数を表示

次のように入力して、.pcap ファイルにキャプチャされたパケットの数を表示できます。

例：.pcap ファイルから単一パケット ダンプの表示

```
Device# show monitor capture file flash:mycap.pcap packet-count
File name:          /flash/mycap.pcap
Number of packets:  50
```

例：.pcap ファイルから単一パケット ダンプの表示

次のように入力して、.pcap ファイルから単一のパケット ダンプを表示できます。

```
Device# show monitor capture file flash:mycap.pcap packet-number 10 dump
Starting the packet display ..... Press Ctrl + Shift + 6 to exit

0000 00 e1 6d 31 f1 80 00 e1 6d 31 f1 80 08 00 45 00  ..m1....m1....E.
0010 00 64 04 be 00 00 ff 01 8e c4 0a 0a 0a 01 0a 0a  .d.....
0020 0a 02 00 00 ec ce 00 2e 00 04 00 00 00 00 09 c9  .....
0030 8f 80 ab cd ab cd ab cd ab cd ab cd ab cd ab cd  .....
0040 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd  .....
0050 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd  .....
0060 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd  .....
0070 ab cd
```

例：.pcap ファイルにキャプチャされたパケットの統計情報を表示

次のように入力して、.pcap ファイルにキャプチャされたパケットの統計情報を表示できます。

```
Device# show monitor capture file flash:mycap.pcap statistics "h225,counter"
===== H225 Message and Reason Counter =====
RAS-Messages:
Call Signalling:
=====
```

例：単純なキャプチャおよび表示

次の例は、レイヤ3 インターフェイス ギガビットイーサネット 1/0/1 でトラフィックをモニタする方法を示しています。

ステップ 1: 次のように入力して関連トラフィックで一致するキャプチャ ポイントを定義します。

```
Device# monitor capture mycap interface GigabitEthernet1/0/3 in
Device# monitor capture mycap match ipv4 any any
Device# monitor capture mycap limit duration 60 packets 50
Device# monitor capture mycap buffer size 100
```

CPU 使用率の上昇を避けるため、制限として最も低いパケット数および時間が設定されています。

ステップ 2: 次のように入力してキャプチャ ポイントが正確に定義されていることを確認します。

```
Device# show monitor capture mycap parameter
      monitor capture mycap interface GigabitEthernet1/0/3 in
      monitor capture mycap match ipv4 any any
      monitor capture mycap buffer size 100
      monitor capture mycap limit packets 50 duration 60
```

```
Device# show monitor capture mycap
Status Information for Capture mycap
Target Type:
```

```

Interface: GigabitEthernet1/0/3, Direction: in
Status : Inactive
Filter Details:
  IPv4
    Source IP: any
    Destination IP: any
    Protocol: any
Buffer Details:
  Buffer Type: LINEAR (default)
  Buffer Size (in MB): 100
File Details:
  File not associated
Limit Details:
  Number of Packets to capture: 50
  Packet Capture duration: 60
  Packet Size to capture: 0 (no limit)
  Packet sampling rate: 0 (no sampling)
    
```

ステップ 3：キャプチャプロセスを開始し、結果を表示します。

```

Device# monitor capture mycap start display
Starting the packet display ..... Press Ctrl + Shift + 6 to exit

  1  0.000000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0030,
seq=0/0, ttl=254
  2  0.003682  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0030,
seq=1/256, ttl=254
  3  0.006586  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0030,
seq=2/512, ttl=254
  4  0.008941  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0030,
seq=3/768, ttl=254
  5  0.011138  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0030,
seq=4/1024, ttl=254
  6  0.014099  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0030,
seq=5/1280, ttl=254
  7  0.016868  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0030,
seq=6/1536, ttl=254
  8  0.019210  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0030,
seq=7/1792, ttl=254
  9  0.024785  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0030,
seq=8/2048, ttl=254
--More--
    
```

ステップ 4：次のように入力して、キャプチャポイントを削除します。

```
Device# no monitor capture mycap
```



(注) 制限が設定してあり、その制限に達するとキャプチャは自動的に停止するため、この特定のケースでは、**stop** コマンドは必要ありません。

pcap の統計情報に使用する構文の詳細については、「その他の参考資料」セクションを参照してください。

例：単純なキャプチャおよび保存

次の例は、フィルタにパケットをキャプチャする方法を示しています。

ステップ 1：次のように入力して、関連トラフィックで一致するキャプチャ ポイントを定義し、それをファイルに関連付けます。

```
Device# monitor capture mycap interface GigabitEthernet1/0/3 in
Device# monitor capture mycap match ipv4 any any
Device# monitor capture mycap limit duration 60 packets 50
Device# monitor capture mycap file location flash:mycap.pcap
```

ステップ 2：次のように入力してキャプチャ ポイントが正確に定義されていることを確認します。

```
Device# show monitor capture mycap parameter
monitor capture mycap interface GigabitEthernet1/0/3 in
monitor capture mycap match ipv4 any any
monitor capture mycap file location flash:mycap.pcap
monitor capture mycap limit packets 50 duration 60
```

```
Device# show monitor capture mycap
```

```
Status Information for Capture mycap
Target Type:
  Interface: GigabitEthernet1/0/3, Direction: in
  Status : Inactive
Filter Details:
  IPv4
  Source IP: any
  Destination IP: any
  Protocol: any
Buffer Details:
  Buffer Type: LINEAR (default)
File Details:
  Associated file name: flash:mycap.pcap
Limit Details:
  Number of Packets to capture: 50
  Packet Capture duration: 60
  Packet Size to capture: 0 (no limit)
  Packet sampling rate: 0 (no sampling)
```

ステップ 3：次のように入力してパケットを開始します。

```
Device# monitor capture mycap start
```

ステップ 4：次のように入力して実行中のエクステンドキャプチャ統計情報を表示します。

```
Device# show monitor capture mycap capture-statistics
Capture statistics collected at software:
  Capture duration - 15 seconds
  Packets received - 40
  Packets dropped - 0
  Packets oversized - 0
  Packets errored - 0
  Packets sent - 40
  Bytes received - 7280
  Bytes dropped - 0
  Bytes oversized - 0
  Bytes errored - 0
```



```
Bytes sent -> 4560
```

ステップ5：十分な時間の経過後に、次のように入力してキャプチャを停止します。

```
Device# monitor capture mycap stop
Capture statistics collected at software (Buffer & Wireshark):
  Capture duration - 20 seconds
  Packets received - 50
  Packets dropped - 0
  Packets oversized - 0
```



(注) あるいは、時間の経過またはパケットカウントが一致した後に、キャプチャ操作を自動的に停止させることもできます。

mycap.pcap ファイルには、キャプチャしたパケットが含まれます。

ステップ6：次のように入力して停止後のエクステンデッドキャプチャの統計情報を表示します。

```
Device# show monitor capture mycap capture-statistics
Capture statistics collected at software:
  Capture duration - 20 seconds
  Packets received - 50
  Packets dropped - 0
  Packets oversized - 0
  Packets errored - 0
  Packets sent - 50
  Bytes received - 8190
  Bytes dropped - 0
  Bytes oversized - 0
  Bytes errored - 0
  Bytes sent -> 5130
```

ステップ7：次のように入力してパケットを表示します。

```
Device# show monitor capture file flash:mycap.pcap
Starting the packet display ..... Press Ctrl + Shift + 6 to exit

  1 0.000000000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0031,
seq=0/0, ttl=254
  2 0.002555000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0031,
seq=1/256, ttl=254
  3 0.006199000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0031,
seq=2/512, ttl=254
  4 0.009199000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0031,
seq=3/768, ttl=254
  5 0.011647000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0031,
seq=4/1024, ttl=254
  6 0.014168000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0031,
seq=5/1280, ttl=254
  7 0.016737000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0031,
seq=6/1536, ttl=254
  8 0.019403000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0031,
seq=7/1792, ttl=254
  9 0.022151000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0031,
seq=8/2048, ttl=254
```

例：バッファのキャプチャの使用

```

10 0.024722000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0031,
seq=9/2304, ttl=254
11 0.026890000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0031,
seq=10/2560, ttl=254
12 0.028862000 10.10.10.2 -> 10.10.10.1 ICMP 114 Echo (ping) request id=0x0031,
seq=11/2816, ttl=254
--More--

```

pcap の統計情報に使用する構文の詳細については、「その他の参考資料」セクションを参照してください。

ステップ 8：次のように入力して、キャプチャ ポイントを削除します。

```
Device# no monitor capture mycap
```

例：バッファのキャプチャの使用

次に、バッファのキャプチャを使用する例を示します。

ステップ 1：次のように入力してバッファ キャプチャ オプションでキャプチャ セッションを起動します。

```

Device# monitor capture mycap interface GigabitEthernet1/0/3 in
Device# monitor capture mycap match ipv4 any any
Device# monitor capture mycap buffer circular size 1
Device# monitor capture mycap start

```

ステップ 2：次のように入力してキャプチャがアクティブであるかどうかを決定します。

```

Device# show monitor capture mycap
Status Information for Capture mycap
Target Type:
  Interface: GigabitEthernet1/0/3, Direction: in
  Status : Active
Filter Details:
  IPv4
  Source IP: any
  Destination IP: any
  Protocol: any
Buffer Details:
  Buffer Type: CIRCULAR
  Buffer Size (in MB): 1
File Details:
  File not associated
Limit Details:
  Number of Packets to capture: 0 (no limit)
  Packet Capture duration: 0 (no limit)
  Packet Size to capture: 0 (no limit)
  Maximum number of packets to capture per second: 1000
  Packet sampling rate: 0 (no sampling)

```

ステップ 3：次のように入力してランタイム時に拡張キャプチャの統計情報を表示します。

```

Device# show monitor capture mycap capture-statistics
Capture statistics collected at software:
  Capture duration - 88 seconds
  Packets received - 1000

```

```
Packets dropped - 0
Packets oversized - 0
Packets errored - 0
Packets sent - 1000
Bytes received - 182000
Bytes dropped - 0
Bytes oversized - 0
Bytes errored - 0
Bytes sent - 114000
```

ステップ4：次のように入力してキャプチャを停止します。

```
Device# monitor capture mycap stop
Capture statistics collected at software (Buffer):
  Capture duration - 2185 seconds
  Packets received - 51500
  Packets dropped - 0
  Packets oversized - 0
```

ステップ5：次のように入力して停止後の拡張キャプチャの統計情報を表示します。

```
Device# show monitor capture mycap capture-statistics
Capture statistics collected at software:
  Capture duration - 156 seconds
  Packets received - 2000
  Packets dropped - 0
  Packets oversized - 0
  Packets errored - 0
  Packets sent - 2000
  Bytes received - 364000
  Bytes dropped - 0
  Bytes oversized - 0
  Bytes errored - 0
  Bytes sent - 228000
```

ステップ6：次のように入力してキャプチャがアクティブであるかどうかを決定します。

```
Device# show monitor capture mycap
Status Information for Capture mycap
  Target Type:
    Interface: GigabitEthernet1/0/3, Direction: in
    Status : Inactive
  Filter Details:
    IPv4
    Source IP: any
    Destination IP: any
    Protocol: any
  Buffer Details:
    Buffer Type: CIRCULAR
    Buffer Size (in MB): 1
  File Details:
    File not associated
  Limit Details:
    Number of Packets to capture: 0 (no limit)
    Packet Capture duration: 0 (no limit)
    Packet Size to capture: 0 (no limit)
    Maximum number of packets to capture per second: 1000
    Packet sampling rate: 0 (no sampling)
```

ステップ7：次のように入力してバッファのパケットを表示します。

```
Device# show monitor capture mycap buffer brief
Starting the packet display ..... Press Ctrl + Shift + 6 to exit

  1  0.000000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0038,
seq=40057/31132,  ttl=254
  2  0.000030  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0038,
seq=40058/31388,  ttl=254
  3  0.000052  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0038,
seq=40059/31644,  ttl=254
  4  0.000073  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0038,
seq=40060/31900,  ttl=254
  5  0.000094  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0038,
seq=40061/32156,  ttl=254
  6  0.000115  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0038,
seq=40062/32412,  ttl=254
  7  0.000137  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0038,
seq=40063/32668,  ttl=254
  8  0.000158  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0038,
seq=40064/32924,  ttl=254
  9  0.000179  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0038,
seq=40065/33180,  ttl=254
 10  0.000200  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0038,
seq=40066/33436,  ttl=254
 11  0.000221  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0038,
seq=40067/33692,  ttl=254
 12  0.000243  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0038,
seq=40068/33948,  ttl=254
--More--
```

パケットがバッファに入ったことに注意してください。

ステップ 8：他の表示モードでパケットを表示します。

```
Device# show monitor capture mycap buffer detailed
Starting the packet display ..... Press Ctrl + Shift + 6 to exit

Frame 1: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0
Interface id: 0
Encapsulation type: Ethernet (1)
Arrival Time: Nov  6, 2015 18:10:06.297972000 UTC
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1446833406.297972000 seconds
[Time delta from previous captured frame: 0.000000000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 0.000000000 seconds]
Frame Number: 1
Frame Length: 114 bytes (912 bits)
Capture Length: 114 bytes (912 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ip:icmp:data]
Ethernet II, Src: Cisco_f3:63:46 (00:e1:6d:f3:63:46), Dst: Cisco_31:f1:c6
(00:e1:6d:31:f1:c6)
  Destination: Cisco_31:f1:c6 (00:e1:6d:31:f1:c6)
    Address: Cisco_31:f1:c6 (00:e1:6d:31:f1:c6)
      .... ..0. .... .. = LG bit: Globally unique address (factory default)

      .... ...0 .... .. = IG bit: Individual address (unicast)
  Source: Cisco_f3:63:46 (00:e1:6d:f3:63:46)
    Address: Cisco_f3:63:46 (00:e1:6d:f3:63:46)
      .... ..0. .... .. = LG bit: Globally unique address (factory default)

      .... ...0 .... .. = IG bit: Individual address (unicast)
```

```

Type: IP (0x0800)
Internet Protocol Version 4, Src: 10.10.10.2 (10.10.10.2), Dst: 10.10.10.1 (10.10.10.1)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    0000 00.. = Differentiated Services Codepoint: Default (0x00)
    .... 0000 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)
  Total Length: 100
  Identification: 0xabdd (43997)
  Flags: 0x00
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
  Fragment offset: 0
  Time to live: 254
  Protocol: ICMP (1)
  Header checksum: 0xe8a4 [validation disabled]
    [Good: False]
    [Bad: False]
  Source: 10.10.10.2 (10.10.10.2)
  Destination: 10.10.10.1 (10.10.10.1)
Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0xa620 [correct]
  Identifier (BE): 56 (0x0038)
  Identifier (LE): 14336 (0x3800)
  Sequence number (BE): 40057 (0x9c79)
  Sequence number (LE): 31132 (0x799c)
  Data (72 bytes)

```

```

0000 00 00 00 00 0b 15 30 63 ab cd ab cd ab cd ab cd .....0c.....
0010 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0020 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0030 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0040 ab cd ab cd ab cd ab cd .....
      Data: 0000000000b153063abcdabcdabcdabcdabcdabcdabcd...
      [Length: 72]

```

Frame 2: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0

Device# **show monitor capture mycap buffer dump**

Starting the packet display Press Ctrl + Shift + 6 to exit

```

0000 00 e1 6d 31 f1 c6 00 e1 6d f3 63 46 08 00 45 00 ..m1....m.cF..E.
0010 00 64 ab dd 00 00 fe 01 e8 a4 0a 0a 0a 02 0a 0a .d.....
0020 0a 01 08 00 a6 20 00 38 9c 79 00 00 00 00 0b 15 .....8.y.....
0030 30 63 ab cd ab cd ab cd ab cd ab cd ab cd ab cd 0c.....
0040 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0050 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0060 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0070 ab cd ..

```

```

0000 00 e1 6d 31 f1 c6 00 e1 6d f3 63 46 08 00 45 00 ..m1....m.cF..E.
0010 00 64 ab de 00 00 fe 01 e8 a3 0a 0a 0a 02 0a 0a .d.....
0020 0a 01 08 00 a6 1d 00 38 9c 7a 00 00 00 00 0b 15 .....8.z.....
0030 30 65 ab cd ab cd ab cd ab cd ab cd ab cd ab cd 0e.....
0040 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0050 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0060 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....

```

```
0070 ab cd
```

ステップ9：次のように入力してバッファをクリアします。

```
Device# monitor capture mycap clear
```



(注) 注：バッファをクリアすると、その内容とともにバッファが削除されます。



(注) バッファの内容を表示する必要がある場合は、`show` コマンドの後に `clear` コマンドを実行します。

ステップ10：トラフィックを再開し、10秒待ってから次のように入力してバッファコンテンツを表示します。



(注) キャプチャがアクティブなときに、バッファから `show` の実行をすることはできません。バッファから `show` を実行する前に、キャプチャを停止する必要があります。しかし、ファイルおよびバッファモードの両方においてキャプチャがアクティブなときに `pcap` ファイルで `show` の実行ができます。ファイルモードでは、キャプチャがアクティブなときに、現在のキャプチャセッションの `pcap` ファイルでパケットを表示することもできます。

```
Device# monitor capture mycap start
Switch# show monitor capture mycap

Status Information for Capture mycap
Target Type:
  Interface: GigabitEthernet1/0/3, Direction: in
  Status : Active
Filter Details:
  IPv4
  Source IP: any
  Destination IP: any
  Protocol: any
Buffer Details:
  Buffer Type: CIRCULAR
  Buffer Size (in MB): 1
File Details:
  File not associated
Limit Details:
  Number of Packets to capture: 0 (no limit)
  Packet Capture duration: 0 (no limit)
  Packet Size to capture: 0 (no limit)
  Maximum number of packets to capture per second: 1000
  Packet sampling rate: 0 (no sampling)
```

ステップ11：次のように入力して、パケットキャプチャを停止し、バッファの内容を表示します。

```
Device# monitor capture mycap stop
Capture statistics collected at software (Buffer):
  Capture duration - 111 seconds
  Packets received - 5000
  Packets dropped - 0
  Packets oversized - 0
```

ステップ 12：次のように入力してキャプチャがアクティブであるかどうかを決定します。

```
Device# show monitor capture mycap
Status Information for Capture mycap
  Target Type:
    Interface: GigabitEthernet1/0/3, Direction: in
  Status : Inactive
  Filter Details:
    IPv4
    Source IP: any
    Destination IP: any
    Protocol: any
  Buffer Details:
    Buffer Type: CIRCULAR
    Buffer Size (in MB): 1
  File Details:
    File not associated
  Limit Details:
    Number of Packets to capture: 0 (no limit)
    Packet Capture duration: 0 (no limit)
    Packet Size to capture: 0 (no limit)
    Maximum number of packets to capture per second: 1000
    Packet sampling rate: 0 (no sampling)
```

ステップ 13：次のように入力してバッファのパケットを表示します。

```
Device# show monitor capture mycap buffer brief
Starting the packet display ..... Press Ctrl + Shift + 6 to exit

  1 0.000000000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0039,
seq=0/0, ttl=254
  2 0.000030000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0039,
seq=1/256, ttl=254
  3 0.000051000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0039,
seq=2/512, ttl=254
  4 0.000072000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0039,
seq=3/768, ttl=254
  5 0.000093000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0039,
seq=4/1024, ttl=254
  6 0.000114000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0039,
seq=5/1280, ttl=254
  7 0.000136000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0039,
seq=6/1536, ttl=254
  8 0.000157000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0039,
seq=7/1792, ttl=254
  9 0.000178000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0039,
seq=8/2048, ttl=254
 10 0.000199000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0039,
seq=9/2304, ttl=254
 11 0.000220000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0039,
seq=10/2560, ttl=254
 12 0.000241000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request id=0x0039,
seq=11/2816, ttl=254
--More<
```

例：出力方向のパケットの簡単なキャプチャおよび保存

ステップ 14：次のように入力して、内部 flash: storage デバイス内の mycap1.pcap ファイルにバッファ コンテンツを保存します。

```
Device# monitor capture mycap export flash:mycap.pcap
Exported Successfully
```



(注) 現在のエクスポート実装では、コマンドを実行すると、エクスポートは「開始」されますが、ユーザにプロンプトを返す場合には完了しません。そこで、ファイルでパケットの表示を実行する前に、Wireshark からコンソールにメッセージが表示されるのを待機する必要があります。

ステップ 15：次のように入力してファイルからキャプチャ パケットを表示します。

```
Device# show monitor capture file flash:mycap.pcap
Starting the packet display ..... Press Ctrl + Shift + 6 to exit

  1 0.000000000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0039,
    seq=0/0, ttl=254
  2 0.000030000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0039,
    seq=1/256, ttl=254
  3 0.000051000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0039,
    seq=2/512, ttl=254
  4 0.000072000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0039,
    seq=3/768, ttl=254
  5 0.000093000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0039,
    seq=4/1024, ttl=254
  6 0.000114000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0039,
    seq=5/1280, ttl=254
  7 0.000136000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0039,
    seq=6/1536, ttl=254
  8 0.000157000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0039,
    seq=7/1792, ttl=254
  9 0.000178000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0039,
    seq=8/2048, ttl=254
 10 0.000199000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0039,
    seq=9/2304, ttl=254
 11 0.000220000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0039,
    seq=10/2560, ttl=254
 12 0.000241000  10.10.10.2 -> 10.10.10.1  ICMP 114 Echo (ping) request  id=0x0039,
    seq=11/2816, ttl=254
--More--
```

ステップ 16：次のように入力して、キャプチャ ポイントを削除します。

```
Device# no monitor capture mycap
```

例：出力方向のパケットの簡単なキャプチャおよび保存

次の例は、フィルタにパケットをキャプチャする方法を示しています。

ステップ 1：次のように入力して、関連トラフィックで一致するキャプチャ ポイントを定義し、それをファイルに関連付けます。


```
Device# monitor capture mycap interface Gigabit 1/0/1 out match ipv4 any any
Device# monitor capture mycap limit duration 60 packets 100
Device# monitor capture mycap file location flash:mycap.pcap buffer-size 90
```

ステップ 2：次のように入力してキャプチャポイントが正確に定義されていることを確認します。

```
Device# show monitor capture mycap parameter
  monitor capture mycap interface GigabitEthernet1/0/1 out
  monitor capture mycap match ipv4 any any
  monitor capture mycap file location flash:mycap.pcap buffer-size 90
  monitor capture mycap limit packets 100 duration 60
```

```
Device# show monitor capture mycap
```

```
Status Information for Capture mycap
Target Type:
  Interface: GigabitEthernet1/0/1, Direction: out
Status : Inactive
Filter Details:
  IPv4
  Source IP: any
  Destination IP: any
  Protocol: any
Buffer Details:
  Buffer Type: LINEAR (default)
File Details:
  Associated file name: flash:mycap.pcap
  Size of buffer(in MB): 90
Limit Details:
  Number of Packets to capture: 100
  Packet Capture duration: 60
  Packet Size to capture: 0 (no limit)
  Packets per second: 0 (no limit)
  Packet sampling rate: 0 (no sampling)
```

ステップ 3：次のように入力してパケットを開始します。

```
Device# monitor capture mycap start
A file by the same capture file name already exists, overwrite?[confirm]
Turning on lock-step mode
```

```
Device#
*Oct 14 09:35:32.661: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
```



(注) 時間の経過またはパケットカウントが一致した後に、キャプチャ操作を自動的に停止させてください。出力に次のメッセージが表示された場合は、キャプチャ処理が停止していることを意味します。

```
*Oct 14 09:36:34.632: %BUFCAP-6-DISABLE_ASYNC: Capture Point mycap disabled. Reason : Wireshark Session Ended
```

mycap.pcap ファイルには、キャプチャしたパケットが含まれます。

ステップ 4：次のように入力してパケットを表示します。

```
Device# show monitor capture file flash:mycap.pcap
Starting the packet display ..... Press Ctrl + Shift + 6 to exit

0.000000  10.1.1.30 -> 20.1.1.2      UDP Source port: 20001  Destination port: 20002
```

```

1.000000  10.1.1.31 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
2.000000  10.1.1.32 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
3.000000  10.1.1.33 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
4.000000  10.1.1.34 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
5.000000  10.1.1.35 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
6.000000  10.1.1.36 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
7.000000  10.1.1.37 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
8.000000  10.1.1.38 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
9.000000  10.1.1.39 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002

```

ステップ5: 次のように入力してキャプチャポイントを削除します。

```
Device# no monitor capture mycap
```

組み込みパケットキャプチャの設定例

例: パケットデータキャプチャの管理

次の例では、パケットデータキャプチャを管理する方法を示します。

```

Device> enable
Device# monitor capture mycap start
Device# monitor capture mycap access-list v4acl
Device# monitor capture mycap limit duration 1000
Device# monitor capture mycap interface GigabitEthernet 0/0/1 both
Device# monitor capture mycap buffer circular size 10
Device# monitor capture mycap start
Device# monitor capture mycap export tftp://10.1.88.9/mycap.pcap
Device# monitor capture mycap stop
Device# end

```

例: キャプチャされたデータのモニタリングとメンテナンス

次の例は、ASCII形式でパケットをダンプする方法を示しています。

```

Device# show monitor capture mycap buffer dump
Starting the packet display ..... Press Ctrl + Shift + 6 to exit

0
0000: 01005E00 00020000 0C07AC1D 080045C0 ..^.....E.
0010: 00300000 00000111 CFDC091D 0002E000 .0.....
0020: 000207C1 07C1001C 802A0000 10030AFA .....*.....
0030: 1D006369 73636F00 0000091D 0001 ..example.....
1
0000: 01005E00 0002001B 2BF69280 080046C0 ..^.....+.....F.
0010: 00200000 00000102 44170000 0000E000 . .....D.....
0020: 00019404 00001700 E8FF0000 0000 .....
2
0000: 01005E00 0002001B 2BF68680 080045C0 ..^.....+.....E.
0010: 00300000 00000111 CFDB091D 0003E000 .0.....
0020: 000207C1 07C1001C 88B50000 08030A6E .....n
0030: 1D006369 73636F00 0000091D 0001 ..example.....
3
0000: 01005E00 000A001C 0F2EDC00 080045C0 ..^.....E.
0010: 003C0000 00000258 CE7F091D 0004E000 .<.....X.....
0020: 000A0205 F3000000 00000000 00000000 .....
0030: 00000000 00D10001 000C0100 01000000 .....
0040: 000F0004 00080501 0300

```

次の例は、mycap という名前のキャプチャの設定に使用するコマンドのリストを表示する方法を示しています。

```
Device# show monitor capture mycap parameter
monitor capture mycap interface GigabitEthernet 1/0/1 both
monitor capture mycap match any
monitor capture mycap buffer size 10
monitor capture mycap limit pps 1000
```

次の例は、キャプチャ ポイントをデバッグする方法を示しています。

```
Device# debug epc capture-point
EPC capture point operations debugging is on

Device# monitor capture mycap start
*Jun 4 14:17:15.463: EPC CP: Starting the capture cap1
*Jun 4 14:17:15.463: EPC CP: (brief=3, detailed=4, dump=5) = 0
*Jun 4 14:17:15.463: EPC CP: final check before activation
*Jun 4 14:17:15.463: EPC CP: setting up c3pl infra
*Jun 4 14:17:15.463: EPC CP: Setup c3pl acl-class-policy
*Jun 4 14:17:15.463: EPC CP: Creating a class
*Jun 4 14:17:15.464: EPC CP: Creating a class : Successful
*Jun 4 14:17:15.464: EPC CP: class-map Created
*Jun 4 14:17:15.464: EPC CP: creating policy-name epc_policy_cap1
*Jun 4 14:17:15.464: EPC CP: Creating Policy epc_policy_cap1 of type 49 and client type
21
*Jun 4 14:17:15.464: EPC CP: Storing a Policy
*Jun 4 14:17:15.464: EPC CP: calling ppm_store_policy with epc_policy
*Jun 4 14:17:15.464: EPC CP: Creating Policy : Successful
*Jun 4 14:17:15.464: EPC CP: policy-map created
*Jun 4 14:17:15.464: EPC CP: creating filter for ANY
*Jun 4 14:17:15.464: EPC CP: Adding acl to class : Successful
*Jun 4 14:17:15.464: EPC CP: Setup c3pl class to policy
*Jun 4 14:17:15.464: EPC CP: Attaching Class to Policy
*Jun 4 14:17:15.464: EPC CP: Attaching epc_class_cap1 to epc_policy_cap1
*Jun 4 14:17:15.464: EPC CP: Attaching Class to Policy : Successful
*Jun 4 14:17:15.464: EPC CP: setting up c3pl qos
*Jun 4 14:17:15.464: EPC CP: DBG> Set packet rate limit to 1000
*Jun 4 14:17:15.464: EPC CP: creating action for policy_map epc_policy_cap1 class_map
epc_class_cap1
*Jun 4 14:17:15.464: EPC CP: DBG> Set packet rate limit to 1000
*Jun 4 14:17:15.464: EPC CP: Activating Interface GigabitEthernet1/0/1 direction both
*Jun 4 14:17:15.464: EPC CP: Id attached 0
*Jun 4 14:17:15.464: EPC CP: inserting into active lists
*Jun 4 14:17:15.464: EPC CP: Id attached 0
*Jun 4 14:17:15.465: EPC CP: inserting into active lists
*Jun 4 14:17:15.465: EPC CP: Activating Vlan
*Jun 4 14:17:15.465: EPC CP: Deleting all temp interfaces
*Jun 4 14:17:15.465: %BUFCAP-6-ENABLE: Capture Point cap1 enabled.
*Jun 4 14:17:15.465: EPC CP: Active Capture 1

Device# monitor capture mycap1 stop
*Jun 4 14:17:31.963: EPC CP: Stopping the capture cap1
*Jun 4 14:17:31.963: EPC CP: Warning: unable to unbind capture cap1
*Jun 4 14:17:31.963: EPC CP: Deactivating policy-map
*Jun 4 14:17:31.963: EPC CP: Policy epc_policy_cap1
*Jun 4 14:17:31.964: EPC CP: Deactivating policy-map Successful
*Jun 4 14:17:31.964: EPC CP: removing provision feature
*Jun 4 14:17:31.964: EPC CP: Found action for policy-map epc_policy_cap1 class-map
epc_class_cap1
*Jun 4 14:17:31.964: EPC CP: cleaning up c3pl infra
*Jun 4 14:17:31.964: EPC CP: Removing Class epc_class_cap1 from Policy
*Jun 4 14:17:31.964: EPC CP: Removing Class from epc_policy_cap1
*Jun 4 14:17:31.964: EPC CP: Successfully removed
```

```
*Jun 4 14:17:31.964: EPC CP: Removing acl mac from class
*Jun 4 14:17:31.964: EPC CP: Removing acl from class : Successful
*Jun 4 14:17:31.964: EPC CP: Removing all policies
*Jun 4 14:17:31.964: EPC CP: Removing Policy epc_policy_cap1
*Jun 4 14:17:31.964: EPC CP: Removing Policy : Successful
*Jun 4 14:17:31.964: EPC CP: Removing class epc_class_cap1
*Jun 4 14:17:31.965: EPC CP: Removing class : Successful
*Jun 4 14:17:31.965: %BUFCAP-6-DISABLE: Capture Point cap1 disabled.
*Jun 4 14:17:31.965: EPC CP: Active Capture 0
```

次の例は、組み込みパケットキャプチャ（EPC）のプロビジョニングをデバッグする方法を示しています。

```
Device# debug epc provision
EPC provisioning debugging is on
```

```
Device# monitor capture mycap start
*Jun 4 14:17:54.991: EPC PROV: No action found for policy-map epc_policy_cap1 class-map
epc_class_cap1
*Jun 4 14:17:54.991: EPC PROV:
*Jun 4 14:17:54.991: Attempting to install service policy epc_policy_cap1
*Jun 4 14:17:54.992: EPC PROV: Attached service policy to epc idb subblock
*Jun 4 14:17:54.992: EPC PROV: Successful. Create feature object
*Jun 4 14:17:54.992: EPC PROV:
*Jun 4 14:17:54.992: Attempting to install service policy epc_policy_cap1
*Jun 4 14:17:54.992: EPC PROV: Successful. Create feature object
*Jun 4 14:17:54.992: %BUFCAP-6-ENABLE: Capture Point cap1 enabled.
```

```
Device# monitor capture mycap stop
*Jun 4 14:18:02.503: EPC PROV: Successful. Remove feature object
*Jun 4 14:18:02.504: EPC PROV: Successful. Remove feature object
*Jun 4 14:18:02.504: EPC PROV: Destroyed epc idb subblock
*Jun 4 14:18:02.504: EPC PROV: Found action for policy-map epc_policy_cap1 class-map
epc_class_cap1
*Jun 4 14:18:02.504: EPC PROV: Deleting EPC action
*Jun 4 14:18:02.504: EPC PROV: Successful. CLASS_REMOVE, policy-map epc_policy_cap1,
class epc_class_cap1
*Jun 4 14:18:02.504: %BUFCAP-6-DISABLE: Capture Point cap1 disabled.
```

その他の参考資料

関連資料

関連項目	参照先
表示フィルタ	表示フィルタの構文については、以下を参照して下さい。 『Display Filter Reference』
pcap ファイル統計情報	pcap ファイル統計情報の表示に使用する構文については、以下で「-z」オプションの詳細を参照してください。 『Tshark Command Reference』

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	役職 (Title)
なし	-

MIB

MIB	MIB 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

テクニカル サポート

説明	リンク
<p>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。</p> <p>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、Product Alert Tool (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。</p> <p>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。</p>	http://www.cisco.com/support

