eStreamer アプリケーション プロトコルについて

Firepower システム Event Streamer（eStreamer）は、メッセージ指向のプロトコルを使用して、イベントおよびホスト プロファイル情報をクライアント アプリケーションにストリーミングします。クライアントは、Management Center からイベント データとホスト プロファイル データを要求でき、管理対象デバイスからは侵入イベント データのみを要求できます。クライアント アプリケーションは、送信されるデータを指定する要求メッセージを送信することでデータ ストリームを開始し、ストリーミング開始後に Management Center または管理対象デバイスからのメッセージ フローを制御します。

このドキュメントでは、Management Center または管理対象デバイス上の eStreamer サービスを eStreamer サーバまたは eStreamer と呼ぶことがあります。

以下の項では、eStreamer サービスに接続するための要件を説明し、eStreamer プロトコルで使用されるコマンドとデータ形式について紹介します。

- 接続の仕様（2-1 ページ）では、eStreamer サービスとクライアントとの間の通信フローについて説明し、クライアントがそのサービスとどのようにやりとりするかについて説明します。
- eStreamer 通信段階について（2-2 ページ）では、クライアント アプリケーションがデータ要求を eStreamer サーバに送信し、eStreamer が要求された情報をクライアントに配信するための通信プロトコルについて説明します。
- eStreamer メッセージ タイプについて（2-6 ページ）では、eStreamer プロトコルで使用されるメッセージ タイプについて説明し、侵入イベント データ、検出イベント データ、メタデータ、およびホスト データをクライアントに返すために eStreamer によって使用されるデータ パケットの基本構造について説明します。また、eStreamer メッセージを解釈できるクライアントの作成に役立つその他の情報を提供します。

接続の仕様

eStreamer サービス：

- SSL 接続を介する TCP を使用した通信（クライアント アプリケーションは SSL ベースの認証をサポートしている必要があります）。
- ポート 8302 で接続要求を受け入れます。
- クライアントがすべての通信セッションを開始するまで待機します。
- すべてのメッセージ フィールドをネットワーク バイト順（ビッグ エンディアン）で書き込みます。
- UTF-8 でテキストをエンコードします。

eStreamer 通信段階について

クライアントと eStreamer サービスとの間には、次の 4 つの主要な通信段階があります。

1. クライアントは eStreamer サーバとの接続を確立し、接続が両方の当事者によって認証されます。

   詳細については、認証された接続の確立（2-2 ページ）を参照してください。

2. クライアントは eStreamer サービスからデータを要求し、ストリーミングされるデータのタイプを指定します。単一のイベント要求メッセージは、イベント メタデータを含む利用可能なイベント データの任意の組み合わせを指定できます。単一のホスト プロファイル要求では、単一のホストまたは複数のホストを指定できます。

   イベント データを要求するための 2 つの要求モードを使用できます。

   – イベント ストリーム要求: クライアントは、要求されたイベント タイプと各タイプのバージョンを指定する要求フラグを含むメッセージを送信し、eStreamer サーバは要求されたデータをストリーミングすることで応答します。

   – 拡張要求: クライアントは、イベント ストリーム要求と同じメッセージ形式で要求を送信しますが、拡張要求用のフラグを設定します。これにより、クライアントと eStreamer サーバ間のメッセージのやりとりが開始され、クライアントはイベント ストリーム要求では利用できない追加の情報とバージョンの組み合わせを要求します。

   データの要求の詳細については、eStreamer からのデータの要求（2-3 ページ）を参照してください。

3. eStreamer は要求されたデータ ストリームをクライアントに確立します。

   詳細については、eStreamer からのデータの受け取り（2-5 ページ）を参照してください。

4. 接続が終了します。
   詳細については、接続の終了（2-6 ページ）を参照してください。

認証された接続の確立

クライアントが eStreamer からデータを要求できるようになるには、クライアントは eStreamer サービスとの SSL 対応 TCP 接続を開始する必要があります。クライアントは、Management Center または管理対象デバイス上の設定済みの管理インターフェイスで要求できます。クライアント接続は管理インターフェイスのトラフィック チャネル構成を強制しないため、接続用のインターフェイスを選択する場合は構成を無視できます。クライアントが接続を開始すると、eStreamer サーバが応答し、クライアントとの SSL ハンドシェイクを開始します。SSL ハンドシェイクの一部として、eStreamer サーバはクライアントの認証証明書を要求し、証明書が有効である（eStreamer サーバで内部認証局（内部 CA）によって署名されている）ことを確認します。

（注）　シスコ は、クライアントが eStreamer サーバによって提示された証明書が信頼できる認証局によって署名されていることを確認するように要求することを推奨しています。これは PKCS＃12 ファイルに含まれる内部 CA 証明書で、シスコ では、新しい eStreamer クライアントを Management Center または管理対象デバイスに登録するときに提供しています。詳細については、eStreamer クライアントの認証の追加（6-3 ページ）を参照してください。

SSL セッションが確立された後、eStreamer サーバは証明書の追加の接続後検証を実行します。この検証では、クライアント接続が証明書で指定されたホストから始まり、証明書のサブジェクト名に適切な値が含まれているか確認されます。いずれかの接続後のチェックが失敗すると、eStreamer サーバは接続を閉じます。必要に応じて、クライアント ホスト名のチェックを実行しないように eStreamer サービスを設定できます（詳細については、eStreamer サービスのオプション（6-5 ページ）を参照）。

クライアントは接続後の検証を実行する必要はありませんが、シスコ では、クライアントがこの検証手順を実行することを推奨しています。認証証明書には、証明書のサブジェクト名に次のフィールド値が含まれています。

表 2-1　証明書のサブジェクト名フィールド

フィールド	値
title	eStreamer
generationQualifier	server

接続後の検証が終了すると、eStreamer サーバはクライアントからのデータ要求を待ちます。

eStreamer からのデータの要求

クライアントが実行する、データ要求の管理におけるタスクの概略は次のとおりです。

- 要求セッションの初期化：セッションの確立（2-3 ページ）を参照してください。
- eStreamer イベント アーカイブからのイベントの要求：イベント ストリーム要求と拡張要求を使用したイベント ストリーミングの開始（2-4 ページ）。
- ホスト データの要求：ホスト データの要求（2-5 ページ）を参照してください。
- 要求の変更：要求の変更（2-5 ページ）を参照してください。

セッションの確立

クライアントは、eStreamer サービスに最初のイベント ストリーム要求を送信することによってセッションを確立します。

この最初のメッセージでは、データ要求フラグを含めるか、または後続のメッセージでデータ要求を送信することができます。この最初のイベント ストリーム要求メッセージ自体は、イベント データ用であれ、ホスト データ用であれ、すべての eStreamer 要求の前提条件です。イベント ストリーム要求メッセージの使用方法については、イベント ストリーム要求メッセージの形式（2-11 ページ）を参照してください。

（注）　eStreamer クライアントは、Management Center または管理対象デバイス上の設定済みの管理インターフェイスで要求できます。クライアント接続は管理インターフェイスのトラフィック チャネル構成を強制しないため、接続用のインターフェイスを選択する場合は構成を無視できます。

イベント ストリーム要求と拡張要求を使用したイベント ストリーミングの開始

eStreamer サービスでは、イベント ストリーミング用の 2 つの要求モードが提供されます。モードを組み合わせた要求も可能です。どちらのモードでも、クライアントはイベント ストリーム要求メッセージで要求を開始しますが、要求フラグ ビットは別々に設定します。イベント ストリームのメッセージ形式に関する詳細については、イベント ストリーム要求メッセージの形式 (2-11 ページ)を参照してください。

eStreamer はイベント ストリーム要求メッセージを受信すると、次のようにクライアント要求を処理します。

- 要求メッセージが要求フラグ フィールドにビット 30 を設定していない場合、eStreamer は要求フラグ フィールド内の他のセット ビットによって要求されたイベントのストリーミングを開始します。詳細については、イベント ストリーム要求の送信 (2-4 ページ)を参照してください。
- イベント ストリーム要求でビット 30 が設定されている場合、eStreamer は拡張要求処理を行います。このビットが設定されている場合は、拡張要求フラグを送信する必要があります。詳細については、拡張要求の送信 (2-4 ページ)を参照してください。eStreamer は重複する要求をすべて解決することに注意してください。複数のフラグまたは複数の拡張要求のいずれかによって同じデータの複数のバージョンを要求する場合は、最新のバージョンが使用されます。たとえば、eStreamer が検出イベント バージョン 1 および 6 のフラグ要求と、バージョン 3 の拡張要求を受信すると、バージョン 6 が送信されます。

イベント ストリーム要求の送信

イベント ストリーム要求は単純なプロセスを使用します。

- クライアントは、開始日時と、データ ストリームに含めるイベントとそのバージョン レベルを指定する要求フラグ フィールドを含む要求メッセージを eStreamer サービスに送信します。
- eStreamer は、指定された時刻にイベントのストリーミングを開始します。ストリーミング プロトコルについては、eStreamer からのデータの受け取り (2-5 ページ)を参照してください。

クライアントのイベント ストリーム要求メッセージの形式と内容については、イベント ストリーム要求メッセージの形式 (2-11 ページ)を参照してください。

クライアントが要求できるイベントのタイプとイベントのバージョンについては、表 2-6 (2-13 ページ)を参照してください。

拡張要求の送信

イベント ストリーム要求メッセージの要求フラグ フィールドにビット 30 を設定すると、拡張要求が開始され、サーバとのネゴシエーションが開始されます。このビットが設定されている場合は、拡張要求フラグを送信する必要があります。拡張要求で使用可能なイベント タイプについては、表 2-22 (2-40 ページ)を参照してください。

拡張要求の手順は次のとおりです。

- クライアントは、イベント ストリーミング要求メッセージを、要求フラグ ビット 30 を 1 に設定 (拡張要求を示す) して eStreamer に送信します。メッセージ形式の詳細については、イベント ストリーム要求メッセージの形式 (2-11 ページ)を参照してください。
- eStreamer は、クライアントが使用可能なサービスのリストをアドバタイズするストリーミング情報メッセージで応答します。ストリーミング情報メッセージの詳細については、ストリーミング情報メッセージの形式 (2-34 ページ)を参照してください。

- クライアントは、使用したいサービスを示すストリーミング要求メッセージと、そのサービスから使用可能なイベントのタイプとバージョンの要求リストを返します。要求リストは、標準イベント ストリーム要求を行う場合の要求フラグ フィールドの設定ビットに対応します。ストリーミング要求メッセージを使用してイベントを要求する方法の詳細については、「拡張要求メッセージの例」セクション（ 2-42 ページ）を参照してください。
- eStreamer は、クライアントのストリーミング要求メッセージを処理し、メッセージで指定された時刻にデータのストリーミングを開始します。ストリーミング プロトコルについては、eStreamer からのデータの受け取り（2-5 ページ）を参照してください。

ホスト データの要求

セッションを確立すると、ホスト データの要求をいつでも送信できます。eStreamer は、要求されたホストの情報を Firepower システム ネットワーク マップから生成します。

要求の変更

確立されたセッションの要求パラメータを変更するには、クライアントは切断して新しいセッションを要求する必要があります。

eStreamer からのデータの受け取り

（注）　eStreamer サーバは、送信したイベントの履歴を保持しません。クライアント アプリケーションは重複したイベントがないかチェックする必要があります。イベントの重複は、いくつかの理由で不注意に発生する可能性があります。たとえば、新しいストリーミング セッションを開始するときに、新しいセッションの開始点としてクライアントによって指定された時間に複数のメッセージがあり、前のセッションで送信されたものもあれば、送信されていないものもある可能性があります。eStreamer は、指定された要求基準を満たすすべてのメッセージを送信します。アプリケーションは、結果の重複を検出する必要があります。

非アクティブの期間中、eStreamer はクライアントに定期的なヌル メッセージを送信して、接続を開いたままにします。クライアントまたは中間ホストからエラー メッセージを受信すると、接続を終了します。

eStreamer は、要求モードに応じて、要求されたデータをクライアントに異なる方法で送信します。

イベント ストリーム要求

クライアントがイベント ストリーム要求を送信すると、eStreamer はメッセージごとにデータ メッセージを返します。クライアントの確認応答を待つことなく、複数のメッセージを連続して送信することができます。特定の時点で、中断し、クライアントの応答を待ちます。クライアント オペレーティング システムは、受信したデータをバッファリングし、クライアントが独自のペースで処理できるようにします。

クライアント要求にメタデータの要求が含まれている場合、eStreamer は最初にメタデータを送信します。クライアントは、後続のイベント レコードを処理するときに使用できるように、それをメモリに保存する必要があります。

拡張要求

クライアントが拡張要求を送信すると、eStreamer はメッセージをキューに入れてバンドルで送信します。eStreamer は、クライアントの確認応答を待つことなく、複数のバンドルを連続して送信することができます。特定の時点で、中断し、クライアントの応答を待ちます。クライアント オペレーティング システムは、受信したデータをバッファリングし、クライアントが独自のペースで読み取ることができるようにします。

クライアントは各バンドルをメッセージごとに解凍し、レコードとブロックの長さを使用して各メッセージを解析します。各メッセージ ヘッダーのメッセージ全体の長さを使用して、各メッセージの終わりに達した時点を計算し、バンドル全体の長さを使用して、バンドルの終わりに達した時点を知ることができます。バンドルを正しく解析するためにそのコンテンツのインデックスは必要ありません。

メッセージのバンドリング メカニズムについては、メッセージ バンドルの形式（2-43 ページ）を参照してください。

クライアントが追加のフロー制御に使用できるヌル メッセージについては、ヌル メッセージの形式（2-8 ページ）を参照してください。

接続の終了

eStreamer サーバは、接続を閉じる前にエラー メッセージの送信を試行します。エラー メッセージについては、エラー メッセージの形式（2-9 ページ）を参照してください。

eStreamer サーバは、次の理由でクライアント接続を閉じる可能性があります。

- メッセージを送信するとエラーが発生する。これには、非アクティブの期間中に eStreamer が送信するイベント データ メッセージとヌル キープアライブ メッセージの両方が含まれます。
- クライアント要求の処理中にエラーが発生する。
- クライアント認証が失敗する（エラー メッセージは送信されません）。
- eStreamer サービスがシャットダウンしている（エラー メッセージは送信されません）。

クライアントはいつでも eStreamer サーバへの接続を閉じることができ、エラー メッセージ形式を使用して理由を eStreamer サーバに通知することを試行する必要があります。

eStreamer メッセージ タイプについて

eStreamer アプリケーション プロトコルは、標準メッセージ ヘッダーと、メッセージのペイロードを含むレコード データが続く様々なサブヘッダー フィールドを含む単純なメッセージ形式を使用します。メッセージ ヘッダーはすべての eStreamer メッセージ タイプで同じです。詳細については、eStreamer メッセージ ヘッダー（2-8 ページ）を参照してください。

表 2-2　eStreamer メッセージ タイプ

メッセージ タイプ	名前	説明
0	ヌル メッセージ	eStreamer サーバとクライアントの両方が、データ フローを制御するためのヌル メッセージを送信します。詳細については、ヌル メッセージの形式 (2-8 ページ) を参照してください。
1	エラー メッセージ	eStreamer サーバとクライアントの両方がエラー メッセージを使用して、接続が閉じた理由を示します。詳細については、エラー メッセージの形式 (2-9 ページ) を参照してください。
2	イベント ストリーム要求	クライアントは、このメッセージ タイプを eStreamer サービスに送信して、新しいストリーミング セッションを開始し、データを要求します。詳細については、イベント ストリーム要求メッセージの形式 (2-11 ページ) を参照してください。
4	イベント データ	eStreamer サービスは、このメッセージ タイプを使用して、イベント データとメタデータをクライアントに送信します。詳細については、イベント データ メッセージの形式 (2-18 ページ) を参照してください。
5	ホスト データ要求	クライアントはこのメッセージ タイプを eStreamer サービスに送信し、ホスト データを要求します。セッションは、すでにイベント ストリーム要求メッセージを介して開始されていなければなりません。詳細については、ホスト要求メッセージの形式 (2-27 ページ) を参照してください。
6	単一ホスト データ	eStreamer サービスは、このメッセージ タイプを使用して、クライアントが要求した単一のホスト データを送信します。詳細については、ホスト データおよびマルチ ホスト データ メッセージの形式 (2-33 ページ) を参照してください。
7	複数のホスト データ	eStreamer サービスは、このメッセージ タイプを使用して、クライアントが要求した複数のホスト データを送信します。詳細については、ホスト データおよびマルチ ホスト データ メッセージの形式 (2-33 ページ) を参照してください。
2049	ストリーミング要求	クライアントは、このメッセージ タイプを拡張要求で使用して、希望するストリーム情報メッセージからアドバタイズされたイベントを指定します。詳細については、拡張要求メッセージの例 (2-42 ページ) を参照してください。
2051	ストリーミング情報	eStreamer サービスは、このメッセージ タイプを拡張要求で使用して、クライアントが使用可能なサービスのリストをアドバタイズします。詳細については、ストリーミング情報メッセージの形式 (2-34 ページ) を参照してください。
4002	メッセージ バンドル	eStreamer サービスは、このメッセージ タイプを使用して、クライアントにストリーミングするメッセージをパッケージ化します。詳細については、メッセージ バンドルの形式 (2-43 ページ) を参照してください。

eStreamer メッセージ ヘッダー

すべての eStreamer メッセージは、次の図に示すメッセージ ヘッダーで始まります。次の表では、フィールドについて説明しています。

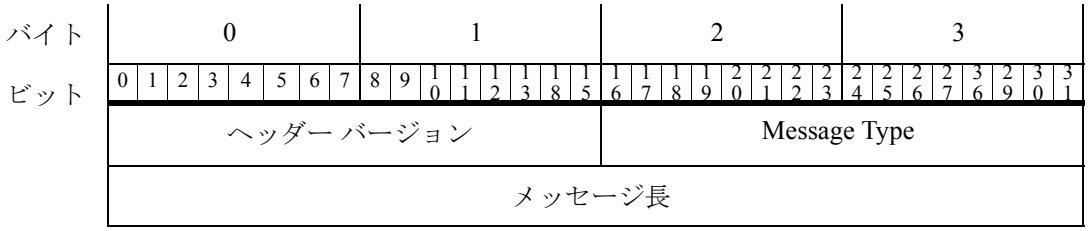


表 2-3　標準の eStreamer メッセージ ヘッダー フィールド

フィールド	データ タイプ	説明
ヘッダー バージョン	uint16	メッセージで使用されるヘッダーのバージョンを示します。eStreamer の現在のバージョンの場合、この値は常に 1 となります。
Message Type	uint16	送信されるメッセージのタイプを示します。現在の値のリストについては、表 2-2 (2-7 ページ) を参照してください。
メッセージ長	uint32	後続のコンテンツの長さを示し、メッセージ ヘッダー自体のバイトを除外します。ヘッダーがありデータのないメッセージのメッセージ長はゼロです。

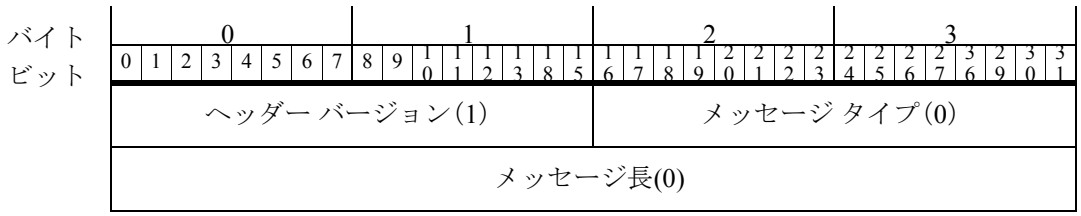
ヌル メッセージの形式

クライアント アプリケーションと eStreamer サービスの両方がヌル メッセージを送信します。ヌル メッセージのタイプは 0 で、メッセージ ヘッダーの後ろにデータはありません。

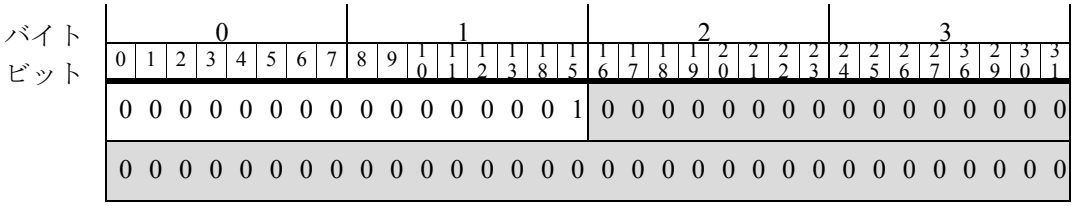
クライアントは、追加のデータを受け入れる準備ができていることを示すために、ヌル メッセージを eStreamer サーバに送信します。eStreamer サービスは、データが送信されていないときに接続のアクティブ状態を維持するために、ヌル メッセージをクライアントに送信します。ヌル メッセージのメッセージ長の値は、常に 0 に設定されています。

ヒント　本書のデータ構造図では、(1) や (115) のようなカッコ内の整数は、定数フィールド値を表します。たとえば、ヘッダー バージョン (1) は、議論中のデータ構造のフィールドが常に 1 の値を持つことを意味します。

ヌル メッセージの形式を以下に示します。メッセージ内のゼロ以外の値のみがヘッダー バージョンです。



バイナリ形式のヌル メッセージの例を次に示します。ゼロ以外の値だけが、ヘッダー バージョン値 1 を示す 2 番目のバイトに存在することに注目してください。メッセージのタイプと長さのフィールド(網掛け)の値はそれぞれ 0 です。



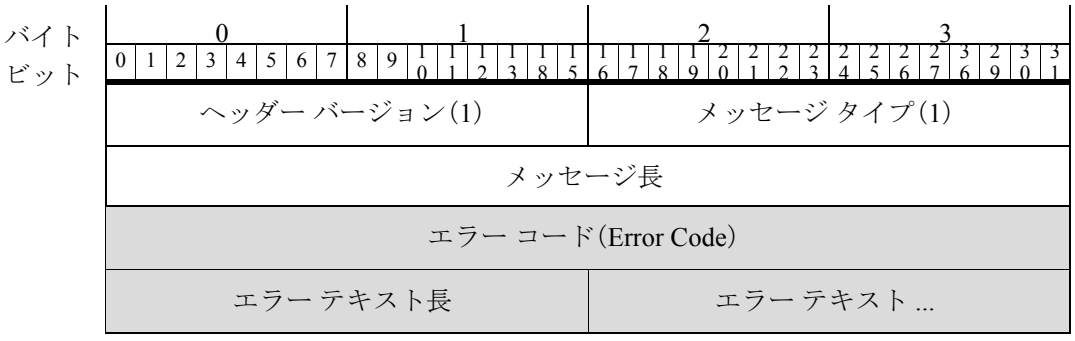
ヒント　このガイドの例は、どのビットが設定されているかを明確に示すためにバイナリ形式で表示されています。これは、イベント要求メッセージ フィールドやイベント影響フィールドなど、一部のメッセージにとって重要です。

エラー メッセージの形式

クライアント アプリケーションと eStreamer サービスの両方でエラー メッセージが使用されます。エラー メッセージのメッセージタイプは 1 で、ヘッダー、エラー コード、エラー テキスト長、および実際のエラー テキストが含まれています。エラー テキストには、0 ～ 65,535 バイトを含めることができます。

クライアント アプリケーションのカスタム エラー メッセージを作成する場合、シスコ は、エラー コードとして -1 を使用することを推奨します。

次の図は、基本的なエラー メッセージの形式を示しています。網掛けのフィールドは、エラー メッセージに固有のフィールドです。

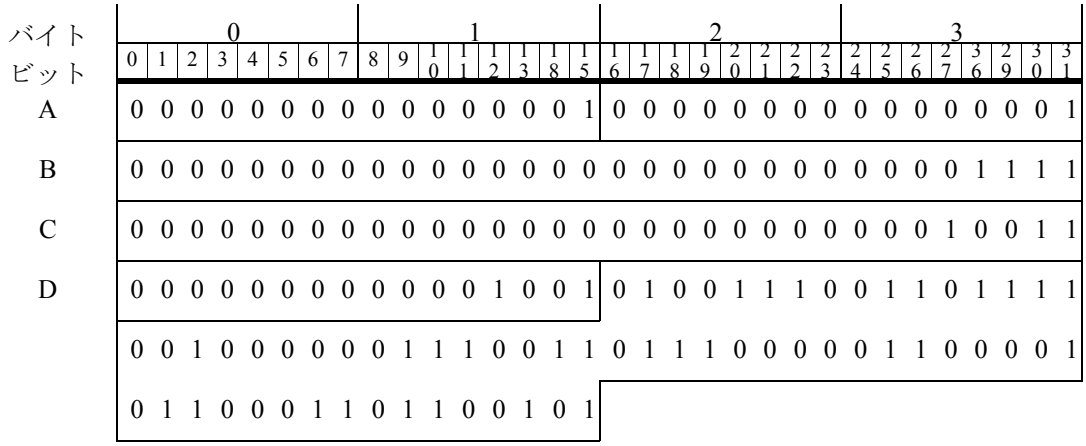


次の表では、エラー コード メッセージの各フィールドについて説明します。

表 2-4　エラー メッセージのフィールド

フィールド	データ タイプ	説明
エラー コード（Error Code）	int32	エラーを表す数値。
エラー テキスト長	uint16	エラー テキスト フィールドに含まれるバイト数。
エラー テキスト	変数（variable）	エラー メッセージ。最大 65,535 バイト。

次の図に、エラー メッセージの例を示します。



上記の例では、次の情報が表示されます。

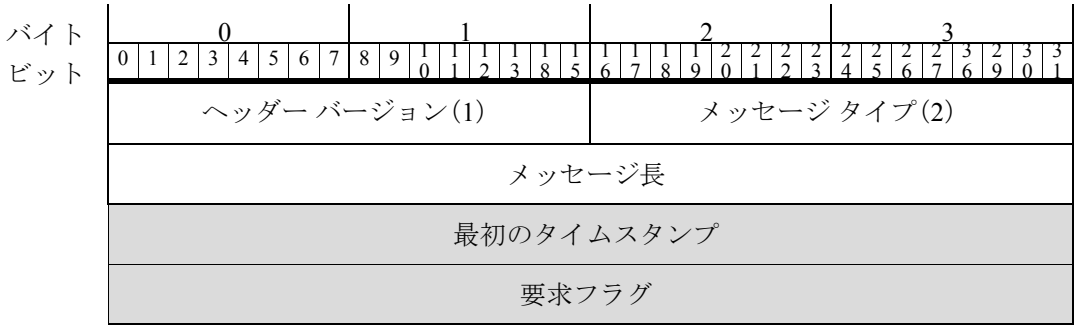
文字	説明
A	最初の 2 バイトは、標準ヘッダー値 1 を示します。2 番目の 2 バイトは値 1 を示し、送信がエラー メッセージであることを示します。
B	この行は、それに続くメッセージ データの量を示します。この例では、15 バイト（バイナリで 1111）のデータが続きます。
C	この行には、エラー コードが表示されます。この例では、メッセージに値 19（10011）が含まれています。したがって、エラー番号 19 がメッセージで送信されます。
D	この行には、エラー メッセージのバイト数（1001、または 9 バイト）が含まれ、エラー メッセージ自体が次の 9 バイトに続きます。エラー メッセージの値は、ASCII テキストに変換された場合、エラー コード 19 に付随するエラー メッセージである「スペースなし（No space）」と等しくなります。

イベント ストリーム要求メッセージの形式

eStreamer クライアントは、イベント ストリーム要求メッセージを使用して、ストリーミング セッションを開始します。要求メッセージには、開始時間と、eStreamer サービスが含むべきデータを指定するためのビット フラグ フィールドが含まれ、イベントの任意の組み合わせ、および侵入イベントの追加データやメタデータにすることができます。イベント ストリーム要求メッセージは、イベント ストリーム要求と拡張要求の両方を開始することができます。メッセージ タイプは 2 です。

ホスト プロファイル情報専用の要求を含む、すべてのデータ要求に対するイベント ストリーム要求メッセージを送信する必要があります。このような場合は、最初にイベント ストリーム要求メッセージを送信し、次にホスト要求メッセージ（タイプ 5）を送信してホスト データを指定します。

次の図に、イベント ストリーム要求メッセージの形式を示します。このメッセージは、標準ヘッダーを使用しています。網掛けのフィールドは要求メッセージに固有のフィールドで、次の表で説明します。



次の表では、イベント ストリーム要求メッセージの各フィールドについて説明します。

表 2-5　イベント ストリーム要求メッセージのフィールド

フィールド	データ タイプ	説明
最初のタイムスタンプ	uint32	セッションの開始を定義します。開始するタイミング： • クライアントが eStreamer に接続するときに開始するには、すべてのタイムスタンプ ビットを 1 に設定します。 • 使用可能な最も古いデータから開始するには、すべてのタイムスタンプ ビットをゼロに設定します。 • 特定の日時に開始するには、UNIX タイムスタンプ（1970 年 1 月 1 日以降の秒数）を指定します。 詳細については、以下の最初のタイムスタンプ（2-12 ページ）を参照してください。
要求フラグ	bits[32]	イベント ストリーム要求で返されるイベントとメタデータのタイプとバージョンを指定します。フラグの定義については、要求フラグ（2-12 ページ）を参照してください。 ビット 30 を設定すると、同じメッセージ内のイベント ストリーム要求と共存できる拡張要求が開始されます。

最初のタイムスタンプ

（注）　以下で説明するように、クライアント アプリケーションは、イベント ストリーム要求を送信するときに、[最初のタイムスタンプ（Initial Timestamp）] フィールドのアーカイブ タイムスタンプを使用する必要があります。これにより、誤ってイベントを除外しないようにします。デバイスは、送信遅延を伴う「ストア アンド フォワード」メカニズムを使用して、データを Management Center に送信します。検出したデバイスによって割り当てられた生成タイムスタンプによってイベントを要求した場合、遅延イベントが除外される可能性があります。

セッションを開始するときは、前のセッションの最後のレコードのアーカイブ タイムスタンプ（「サーバ タイムスタンプ」とも呼ばれる）から起動することを推奨します。これは技術的な要件ではありませんが、強く推奨されます。特定の状況下では、生成タイムスタンプを使用すると、意図せずに新しいストリーミング セッションからイベントを除外してしまう可能性があります。

ストリーミングされたイベントにアーカイブ タイムスタンプを含めるには、要求フラグ フィールドにビット 23 を設定する必要があります。

時間ベースのイベントだけがアーカイブ タイムスタンプを持つことに注意してください。ビット 23 が設定された拡張イベント ヘッダーが要求された場合、メタデータなどの eStreamer が生成するイベントのこのフィールドはゼロになります。

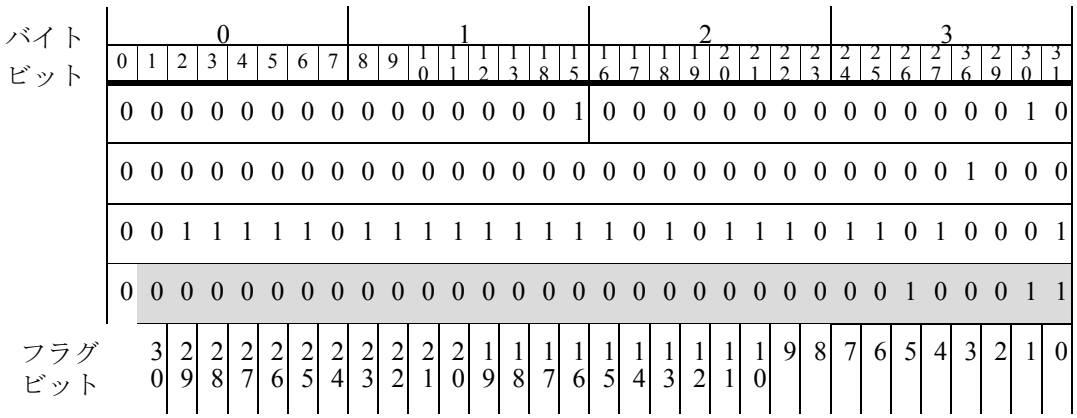
要求フラグ

eStreamer が送信するイベントのタイプを選択するには、イベント データ要求のフラグ フィールドにビット 0 ～ 29 を設定します。拡張要求モードをアクティブにするには、ビット 30 を設定します。ビット 30 を設定しても、データは直接要求されません。このビットが設定されている場合は、拡張要求フラグを送信する必要があります。クライアントは、イベント ストリーム要求メッセージの送信後のサーバ/クライアント メッセージ ダイアログ中にデータを要求します。拡張要求については、eStreamer からのデータの要求（2-3 ページ）を参照してください。

[要求フラグ（Request Flags）] フィールドのビット設定の定義については、表 2-6（2-13 ページ）を参照してください。異なるフラグは、異なるバージョンのイベント データを要求します。たとえば、4.10 形式ではなく Firepower システム 4.9 形式でデータを取得するには、異なるフラグ ビットを設定します。特定の製品バージョンのデータを要求するときに使用するフラグの固有情報については、表 2-7（2-16 ページ）を参照してください。

個々のメタデータ レコードではなく、バージョン別にメタデータを要求することに注意してください。サポートされている各メタデータのバージョンについては、要求フラグ（2-12 ページ）を参照してください。

次の図では、現在使用されているフラグ フィールドのビットを網掛けにしています。



各要求フラグ ビットについては、次の表を参照してください。

表 2-6　　要求フラグ

ビット フィールド	説明
ビット 0	侵入イベントに関連付けられたパケット データの送信を要求します。1 に設定すると、パケット データが侵入イベントとともに送信されます。0 に設定すると、パケット データは送信されません。
ビット 1	侵入、検出、相関、および接続イベントに関連するバージョン 1 メタデータの送信を要求します。1 に設定すると、バージョン 1 のメタデータがイベントとともに送信されます。0 に設定すると、バージョン 1 のメタデータは送信されません。 メタデータを使用して、イベントのコード化されたフィールドおよび数値フィールドを解決できます。eStreamer がメタデータをクライアントに送信する方法と、クライアントがメタデータを使用する方法に関する一般的な情報については、メタデータについて (2-44 ページ) を参照してください。
ビット 2	侵入イベントの送信を要求します。ビット 2、ビット 6、またはビット 2 および 6 の両方が 1 に設定されているが、拡張要求フラグであるビット 30 が 0 に設定されている場合、システムはこれをバージョン 4.x クライアントからの要求として解釈し、レコード タイプ 104/105 が送信されます。ビット 2、ビット 6、またはビット 2 と 6 の両方が 1 に設定され、ビット 30 が 1 に設定されているときにイベント タイプが指定されていない場合、システムはこれをバージョン 5.0-5.1 クライアントからの要求として解釈し、レコード タイプ 207/208 が送信されます。ビット 30 が 1 に設定され、特定のイベント タイプが要求された場合は、ビット 2 および 6 に関係なく、侵入イベントが送信されます。 レコード タイプの要求の詳細については、拡張要求の送信 (2-4 ページ) を参照してください。 ビット 2、ビット 6、ビット 30 がすべて 0 に設定されている場合、侵入イベントは送信されません。 ビット 6 は、ビット 2 と同じ方法で使用されます。いずれかのビットを設定して侵入イベントを要求することができます。これらのビットの 1 つを 0 に設定しても、他のビットは上書きされません。ビット 2 を 0 に設定してビット 6 を 1 に設定するか、またはビット 2 を 1 に設定してビット 6 を 0 に設定すると、侵入イベントの要求として解釈されます。
ビット 3	検出データ バージョン 1 (Management Center 3.2) の送信を要求します。0 に設定すると、検出データ バージョン 1 は送信されません。 検出イベントの詳細については、検出と接続データ構造の概要 (4-1 ページ) を参照してください。
ビット 4	相関データ バージョン 1 (Management Center 3.2) の送信を要求します。0 に設定すると、相関データ バージョン 1 は送信されません。

表 2-6　要求フラグ(続き)

ビット フィールド	説明
ビット 5	影響相関イベント(侵入影響アラート)の送信を要求します。1 に設定すると、侵入影響アラートが送信されます。0 に設定すると、侵入影響アラートは送信されません。 侵入影響アラートの詳細については、侵入の影響アラート データ 5.3 以上(3-18 ページ)を参照してください。
ビット 6	ビット 6 は、ビット 2 と同じ方法で使用されます。ビット 2(2-13 ページ)を参照してください。
ビット 7	検出データ バージョン 2(Management Center 4.0 ～ 4.1)の送信を要求します(1 に設定されている場合)。0 に設定すると、検出データ バージョン 2 は送信されません。
ビット 8	接続データ バージョン 1(Management Center 4.0 ～ 4.1)の送信を要求します(1 に設定されている場合)。0 に設定すると、接続データ バージョン 1 は送信されません。
ビット 9	相関データ バージョン 2(Management Center 4.0 ～ 4.1.x)の送信を要求します(1 に設定されている場合)。0 に設定すると、相関ポリシー データ バージョン 2 は送信されません。
ビット 10	検出データ バージョン 3(Management Center 4.5 ～ 4.6.1)の送信を要求します(1 に設定されている場合)。0 に設定すると、検出データ バージョン 3 は送信されません。 レガシー検出イベントの詳細については、レガシー ディスカバリ データ構造(B-93 ページ)を参照してください。
ビット 11	イベントの送信を無効にします。
ビット 12	接続データ バージョン 3(Management Center 4.5 ～ 4.6.1)の送信を要求します(1 に設定されている場合)。0 に設定すると、接続データ バージョン 3 は送信されません。
ビット 13	相関データ バージョン 3(Management Center 4.5 ～ 4.6.1)の送信を要求します。0 に設定すると、相関データ バージョン 3 は送信されません。
ビット 14	侵入、検出、相関、および接続イベントに関連するバージョン 2 メタデータの送信を要求します。1 に設定すると、バージョン 2 のメタデータがイベントとともに送信されます。0 に設定すると、バージョン 2 のメタデータは送信されません。 eStreamer がメタデータをクライアントに送信する方法と、クライアントがメタデータを使用する方法に関する一般的な情報については、メタデータについて(2-44 ページ)を参照してください。
ビット 15	侵入、相関、検出、および接続イベントに関連するバージョン 3 メタデータの送信を要求します。1 に設定すると、バージョン 3 のメタデータがイベントとともに送信されます。0 に設定すると、バージョン 3 のメタデータは送信されません。 eStreamer がメタデータをクライアントに送信する方法と、クライアントがメタデータを使用する方法に関する一般的な情報については、メタデータについて(2-44 ページ)を参照してください。
ビット 16	未使用(Unused)
ビット 17	検出データ バージョン 4(Management Center 4.7 ～ 4.8.x)の送信を要求します。0 に設定すると、検出データ バージョン 4 は送信されません。
ビット 18	接続データ バージョン 4(Management Center 4.7 ～ 4.9.0.x)の送信を要求します(1 に設定されている場合)。0 に設定すると、接続データ バージョン 4 は送信されません。詳細については、ユーザ レコード(4-20 ページ)を参照してください。
ビット 19	相関データ バージョン 4(Management Center 4.7)の送信を要求します。0 に設定すると、相関データ バージョン 4 は送信されません。 Management Center 4.7 形式で送信される相関イベントについては、レガシー相関イベントのデータ構造(B-274 ページ)を参照してください。

表 2-6　要求フラグ(続き)

ビット フィールド	説明
ビット 20	侵入、検出、ユーザ アクティビティ、相関、および接続イベントに関連するバージョン 4 メタデータの送信を要求します。1 に設定すると、バージョン 4 のメタデータがイベントとともに送信されます。0 に設定すると、バージョン 4 のメタデータは送信されません。 バージョン 4 のメタデータには、次のものが含まれます。 • 相関(コンプライアンス)ルールの情報 • 相関(コンプライアンス)ポリシーの情報 • フィンガープリント レコード • クライアント アプリケーション レコード • クライアント アプリケーション タイプのレコード • 脆弱性レコード • ホストの重要度レコード • ネットワーク プロトコル レコード • ホストの属性レコード • スキャン タイプのレコード • ユーザ レコード • サービス検出デバイス(バージョン 2)のレコード • イベント分類(バージョン 2)のレコード • 優先順位レコード • ルール情報(バージョン 2) • マルウェアの情報 ビット 22 を使用してビット 20 を要求すると、ユーザのメタデータも送信されます。 eStreamer がメタデータをクライアントに送信する方法と、クライアントがメタデータを使用する方法に関する一般的な情報については、メタデータについて(2-44 ページ)を参照してください。
ビット 21	バージョン 1 ユーザ イベントの送信を要求します。ユーザ イベントの詳細については、ユーザ レコード(4-20 ページ)を参照してください。
ビット 22	相関データ バージョン 5(Management Center 4.8.0.2 ～ 4.9.1)の送信を要求します。0 に設定すると、相関データ バージョン 5 は送信されません。 ビット 22 を使用してビット 20 を要求すると、ユーザのメタデータも送信されます。 レガシー相関(コンプライアンス)イベントの詳細については、レガシー相関イベントのデータ構造(B-274 ページ)を参照してください。
ビット 23	拡張イベント ヘッダーを要求します。1 に設定すると、イベントは、eStreamer サーバが処理するためにイベントがアーカイブされたときに適用されたタイムスタンプと、将来の使用のために予約された 4 バイトが付いて送信されます。このフィールドが 0 に設定されている場合、イベントは、レコード タイプとレコード長のみを含む標準のイベント ヘッダーが付いて送信されます。 イベント メッセージ ヘッダーについては、eStreamer メッセージ ヘッダー(2-8 ページ)を参照してください。

表 2-6　要求フラグ(続き)

ビット フィールド	説明
ビット 24	検出データ バージョン 5(Management Center 4.9.0.x)の送信を要求します。0 に設定すると、検出データ バージョン 5 は送信されません。 検出イベントの詳細については、検出と接続データ構造の概要(4-1 ページ)を参照してください。
ビット 25	検出データ バージョン 6(Management Center 4.9.1+)の送信を要求します。0 に設定すると、検出データ バージョン 6 は送信されません。 検出イベントの詳細については、検出と接続データ構造の概要(4-1 ページ)を参照してください。
ビット 26	接続データ バージョン 5(Management Center 4.9.1 ～ 4.10.x)の送信を要求します(1 に設定されている場合)。0 に設定すると、接続データ バージョン 5 は送信されません。詳細については、ユーザ レコード(4-20 ページ)を参照してください。
ビット 27	追加データ レコード内の侵入イベントに関連するイベント追加データを要求します。 イベント データの詳細については、表 3-11 侵入イベント追加データのデータ ブロック フィールド(3-29 ページ)を参照してください。
ビット 28	検出データ バージョン 7(Management Center 4.10.0+)の送信を要求します。0 に設定すると、検出データ バージョン 7 は送信されません。 検出イベントの詳細については、検出と接続データ構造の概要(4-1 ページ)を参照してください。
ビット 29	相関データ バージョン 6(Management Center 4.10 ～ 4.10.x)の送信を要求します。0 に設定すると、相関ポリシー データ バージョン 6 は送信されません。 ビット 29 を使用してビット 20 を要求すると、ユーザのメタデータも送信されます。 相関イベントの詳細については、製品の以前のバージョンを参照してください。
ビット 30	eStreamer への拡張要求を示します。このビットが設定されている場合は、拡張要求フラグを送信する必要があります。拡張要求については、拡張要求の送信(2-4 ページ)を参照してください。

特定のバージョンのデータを要求するために使用するフラグを決定するには、次の表を参照してください。バージョン 5.0 以降の場合は、ビット 30 の使用の詳細について、拡張要求の送信(2-4 ページ)を参照してください。

表 2-7　製品バージョン別のイベント要求フラグ

要求されたデータのタイプ	4.9.0.x	4.9.1.x	4.10.x	5.0+	5.1	5.1.1+
パケット データ	ビット 0	ビット 0	ビット 0	ビット 0	ビット 0	ビット 0
侵入イベント	ビット 2	ビット 2	ビット 2	ビット 2	ビット 2	ビット 30
メタデータ	ビット 20	ビット 20	ビット 20	ビット 20	ビット 20	ビット 20
検出イベント	ビット 24	ビット 25	ビット 28	ビット 30	ビット 30	ビット 30
相関イベント	ビット 22	ビット 22	ビット 29	ビット 30	ビット 30	ビット 30
イベント追加データ	—	—	ビット 27	ビット 27	ビット 27	ビット 27
影響イベント アラート	ビット 5	ビット 5	ビット 5	ビット 5	ビット 5	ビット 5

表 2-7　製品バージョン別のイベント要求フラグ(続き)

要求されたデータのタイプ	4.9.0.x	4.9.1.x	4.10.x	5.0+	5.1	5.1.1+
接続データ	ビット 18	ビット 26	ビット 26	ビット 30	ビット 30	ビット 30
ユーザ イベント	ビット 21	ビット 21	ビット 21	ビット 30	ビット 30	ビット 30
マルウェア イベント	—	—	—	—	—	ビット 30
ファイル イベント	—	—	—	—	—	ビット 30

注意　バージョン 5.x より前のすべてのイベント タイプでは、参照クライアントは、検出エンジン ID フィールドをセンサー ID とラベル付けします。

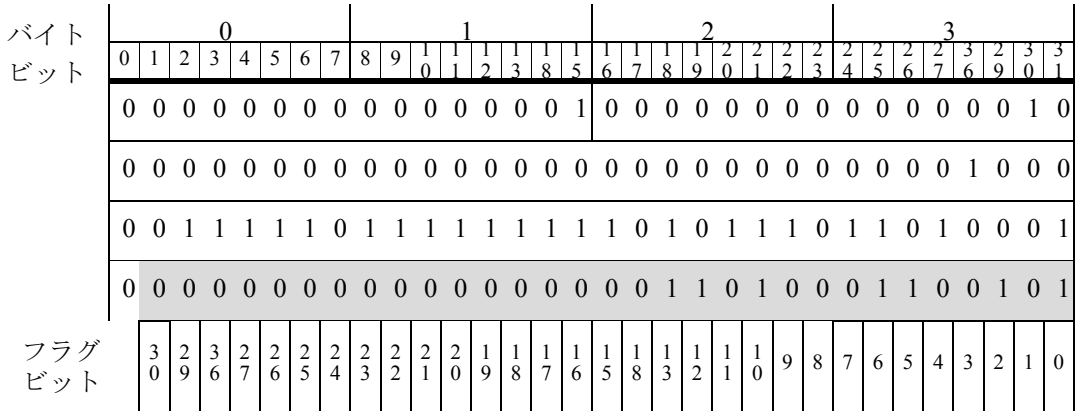
次の例では、バージョン 1 のメタデータとパケット フラグの両方を使用して、タイプ 7 (Firepower システム 3.2+ と互換性あり)の侵入イベントを要求しています。

バイト	0								1								2								3							
ビット	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	0	1	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
フラグビット	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

Firepower システム 3.2 と互換性のあるデータ(侵入イベント、パケット、メタデータ、影響アラート、ポリシー違反イベント、およびバージョン 2.0 イベントを含む)のみを要求するには、以下を使用します。

バイト	0								1								2								3							
ビット	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	36	29	30	31
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	0	1	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
フラグビット	30	29	36	27	26	25	24	23	22	21	20	19	18	17	16	15	18	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

侵入影響アラート、相関イベント、検出イベント、接続イベント、およびパケットとバージョン 3 メタデータを含むタイプ 7 の侵入イベントを Management Center 4.6.1+ 形式で要求するには、以下を使用します。



イベント データ メッセージの形式

eStreamer サービスは、イベント要求を受信すると、イベント データと関連するメタデータをクライアントに送信します。イベント データ メッセージのメッセージ タイプは 3 です。各メッセージには、イベント データまたはメタデータのいずれかを含む単一のデータ レコードが含まれています。

タイプ 3 のメッセージは、イベント データとメタデータのみを伝送することに注意してください。eStreamer は、タイプ 6（単一ホスト）とタイプ 7（マルチホスト）メッセージ内のホスト情報を送信します。ホスト メッセージ形式については、ホスト データおよびマルチ ホスト データ メッセージの形式（2-33 ページ）を参照してください。

イベント データ メッセージの構成について

eStreamer が送信するイベント データおよびメタデータ メッセージには、次のセクションが含まれています。

- eStreamer メッセージ ヘッダー：eStreamer メッセージ ヘッダー（2-8 ページ）で定義されている標準メッセージ ヘッダー。
- イベント固有のサブヘッダー：追加のイベントの詳細を記述し、後続のペイロード データの構造を決定するコードを含む、イベント タイプによって異なるフィールドのセット。
- データ レコード：固定長フィールドとデータ ブロック。

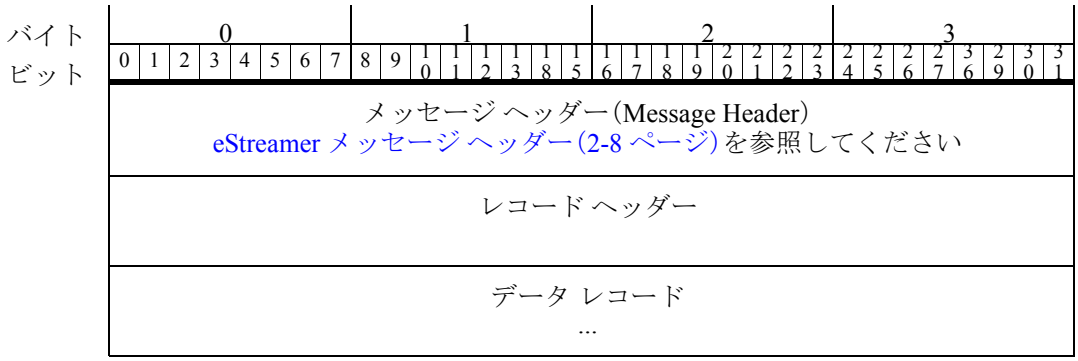
（注）　クライアントは、フィールド長に基づいてすべてのメッセージを展開する必要があります。

イベント タイプ別のイベント メッセージ形式については、以下を参照してください。

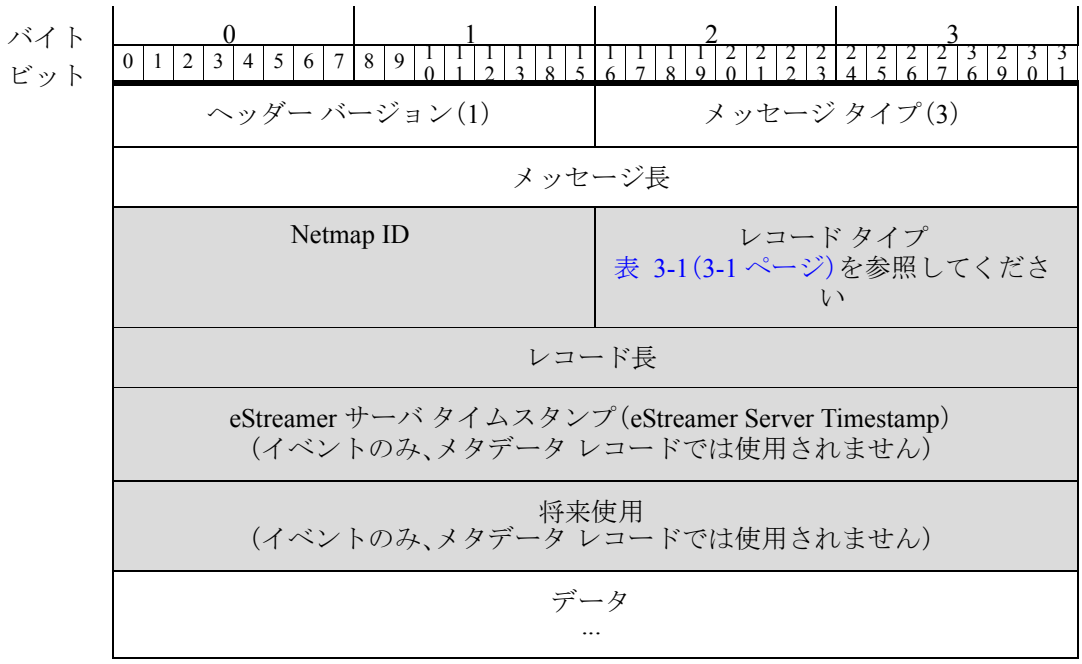
- 侵入イベント データ レコードとすべてのメタデータ レコードについては侵入イベントとメタデータ メッセージの形式（2-19 ページ）。これらのメッセージは固定長フィールドを持ちます。
- 検出イベントまたはユーザ イベント データを含むメッセージについては検出イベント メッセージの形式（2-21 ページ）。標準の eStreamer メッセージ ヘッダーおよび侵入イベント メッセージに類似したレコード ヘッダーに加えて、検出メッセージには、イベント タイプとサブタイプ フィールドが含まれた独特の検出イベント ヘッダーがあります。検出イベント メッセージ内のデータ レコードは、可変長フィールドとカプセル化されたブロックの複数の層を持つことができるシリーズ 1 ブロックにパッケージ化されます。
- 接続統計情報を含むメッセージについては接続イベント メッセージの形式（2-23 ページ）。それらの一般的な構造は、検出イベント メッセージと同じです。ただし、データ ブロック タイプは接続統計情報に固有のものです。
- 相関（コンプライアンス）イベント データを含むメッセージについては相関イベント メッセージの形式（2-23 ページ）。これらのメッセージのヘッダーは侵入イベント メッセージと同じですが、データ ブロックはシリーズ 1 ブロックです。
- 可変長フィールドおよび侵入イベントの追加データなどのネストされたデータ ブロックの複数の層を含む侵入関連レコード タイプを配信する一連のメッセージについてはイベント追加データ メッセージの形式（2-25 ページ）。このメッセージ シリーズの構造に関する一般的な情報については、イベント追加データ メッセージの形式（2-25 ページ）を参照してください。シリーズ 1 ブロックに類似しているが、個別に番号が付けられているこのシリーズのブロックの構造に関する情報については、データ ブロック ヘッダー（2-26 ページ）を参照してください。

侵入イベントとメタデータ メッセージの形式

次の図に、侵入イベントおよびメタデータ メッセージの一般的な構造を示します。



次の図に、侵入イベントおよびメタデータ メッセージ形式のレコード ヘッダー部分の詳細を示します。レコード ヘッダー フィールドは網掛けされています。その次にある表では、フィールドを定義しています。



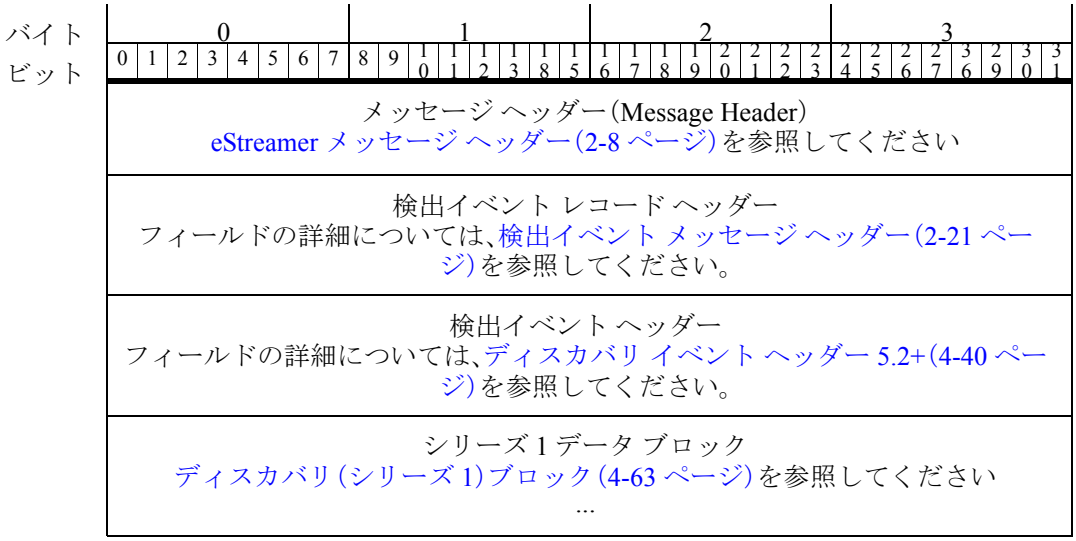
次の表に、侵入イベントおよびメタデータ メッセージのヘッダーの各フィールドについて説明します。

表 2-8　侵入イベントとメタデータ レコード ヘッダー フィールド

フィールド	データ タイプ	説明
Netmap ID	uint16	このフィールドの第 1 ビットは、ヘッダーがアーカイブ タイムスタンプを含む拡張ヘッダーであるかどうかを示すフラグです。残りの 15 ビットは、イベントが検出されたドメインの Netmap ID を含むオプションのフィールドです。このフィールドは、使用されていない場合は空のままです。Netmap ID は、メタデータで提供されるドメインにマップされます。
レコード タイプ	uint16	データ レコードのコンテンツ タイプを識別します。レコード タイプのリストについては、表 3-1 侵入イベントと一般的なメタデータのレコード タイプ(3-1 ページ)を参照してください。
レコード長	uint32	レコード ヘッダーの後のメッセージのコンテンツの長さ。レコード ヘッダーの 8 または 16 バイトは含まれません。(レコード長 + レコード ヘッダーの長さは、メッセージ長と等しくなります。)
eStreamer サーバ タイムスタンプ(eStreamer Server Timestamp)	uint32	イベントが eStreamer サーバによってアーカイブされたときに適用されるタイムスタンプを示します。アーカイブ タイムスタンプとも呼ばれます。 要求メッセージ フラグにビット 23 が設定されている場合にのみ表示されるフィールド。
将来使用	uint32	今後使用するために予約されています。 要求メッセージ フラグにビット 23 が設定されている場合にのみ表示されるフィールド。

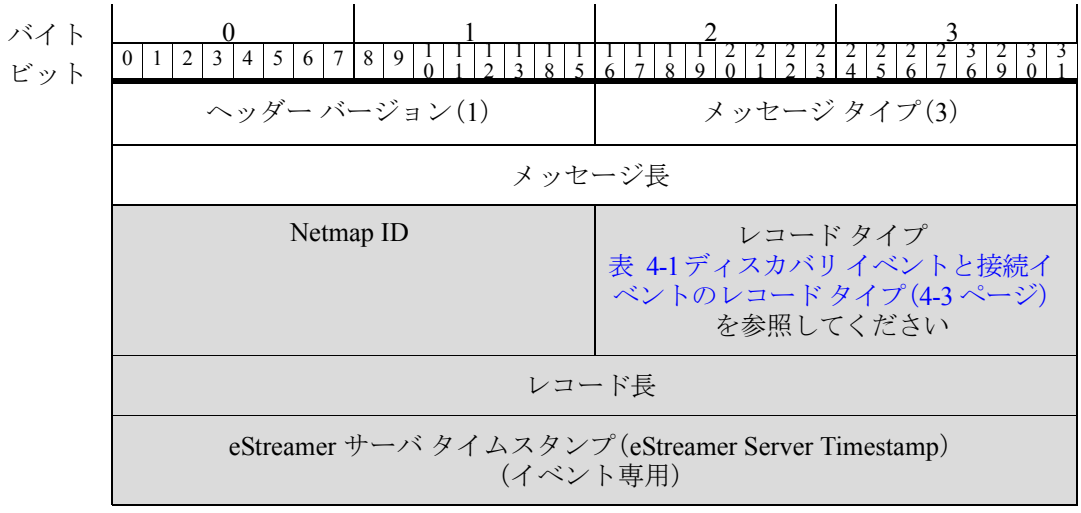
検出イベント メッセージの形式

次の図に、検出イベント メッセージの構造を示します。標準の eStreamer メッセージ ヘッダーとイベント レコード ヘッダーの後には、検出イベント メッセージとユーザ イベント メッセージでのみ使用される検出イベント ヘッダーが続きます。メッセージの検出イベント ヘッダー セクションには、検出イベント タイプおよびサブタイプ フィールドが含まれており、これらのフィールドが一緒になって後続のデータ ブロックへのキーを形成します。現在の検出イベント タイプおよびサブタイプについては、表 4-29 タイプ/サブタイプ別のディスカバリ イベントと接続イベント (4-42 ページ) を参照してください。



検出イベント メッセージ ヘッダー

次の図の網掛け部分は、検出イベント データ メッセージ形式のレコード ヘッダーのフィールドを示し、それに続くイベント ヘッダーの位置を示しています。次の表では、検出イベント メッセージ ヘッダーのフィールドを定義しています。



将来使用 （イベント専用）
検出イベント ヘッダー 表 4-28 ディスカバリ イベント ヘッダーのフィールド (4-41 ページ) を参照してください
シリーズ 1 データ ブロック ディスカバリ (シリーズ 1) ブロック (4-63 ページ) を参照してください ...

次の表では、検出イベント メッセージのレコード ヘッダーとイベント ヘッダーのフィールドについて説明します。

表 2-9　検出イベント メッセージ ヘッダーのフィールド

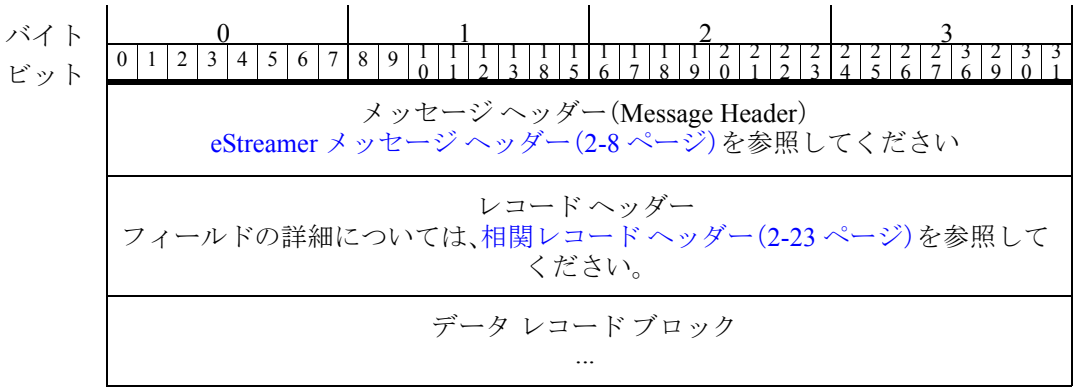
フィールド	データ タイプ	説明
Netmap ID	uint16	このフィールドの第 1 ビットは、ヘッダーがアーカイブ タイムスタンプを含む拡張ヘッダーであるかどうかを示すフラグです。残りの 15 ビットは、イベントが検出されたドメインの Netmap ID を含むオプションのフィールドです。このフィールドは、使用されていない場合は空のままです。Netmap ID は、メタデータで提供されるドメインにマップされます。
レコード タイプ	uint16	データ レコードのコンテンツ タイプを識別します。レコード タイプのリストについては、表 4-1 ディスカバリ イベントと接続イベントのレコード タイプ (4-3 ページ) を参照してください。
レコード長	uint32	レコード ヘッダーの後のメッセージのコンテンツの長さ。レコード ヘッダーの 8 または 16 バイトは含まれません。(レコード長 + レコード ヘッダーの長さは、メッセージ長と等しくなります。)
eStreamer サーバ タイムスタンプ (eStreamer Server Timestamp)	uint32	イベントが eStreamer サーバによってアーカイブされたときに適用されるタイムスタンプを示します。アーカイブ タイムスタンプとも呼ばれます。イベント ストリーム要求の要求フラグ フィールドにビット 23 が設定されている場合にのみ存在するフィールド。
将来使用	uint32	今後使用するために予約されています。要求メッセージ フラグにビット 23 が設定されている場合にのみ表示されるフィールド。
検出イベント ヘッダー	さまざま	イベント タイプとサブタイプを含む複数のフィールドが含まれており、これらが一緒になって後続のデータ構造への固有キーを形成します。検出イベント ヘッダーのフィールドの定義については、ディスカバリ イベント ヘッダー 5.2+ (4-40 ページ) を参照してください。

接続イベント メッセージの形式

接続統計情報を含むメッセージの構造は、検出イベント メッセージと同じです。一般的なメッセージ形式の情報については、検出イベント メッセージの形式（2-21 ページ）を参照してください。接続イベント メッセージは、それらが組み込むデータ ブロック タイプの点で区別されます。

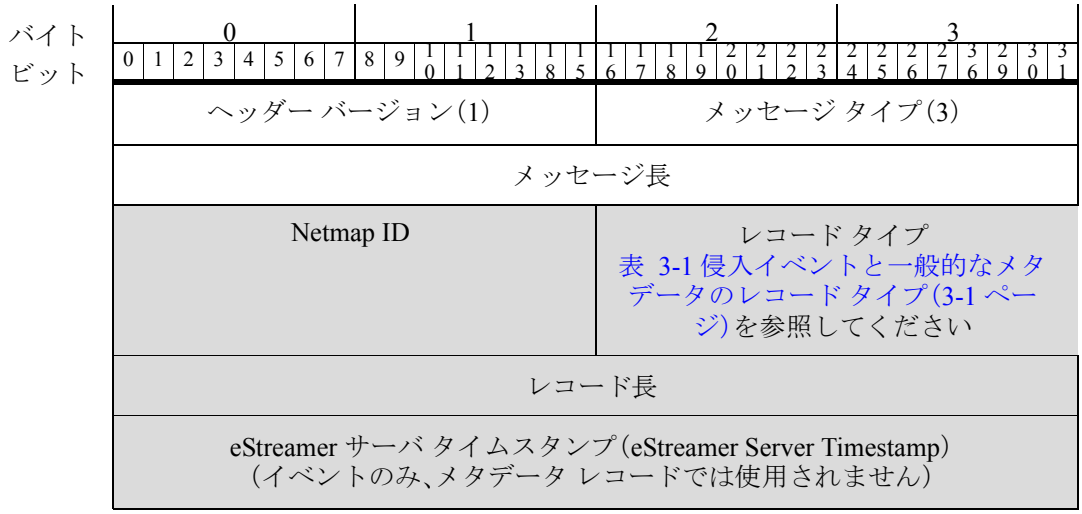
相関イベント メッセージの形式

次の図に、相関（コンプライアンス）イベント メッセージの一般的な構造を示します。標準の eStreamer メッセージ ヘッダーとレコード ヘッダーの直後には、メッセージのデータ レコード セクションのデータ ブロックが続きます。相関メッセージは、シリーズ 1 データ ブロックを使用します。



相関レコード ヘッダー

次の図の網掛け部分は、相関イベント メッセージのレコード ヘッダーのフィールドを示しています。相関メッセージはシリーズ 1 データ ブロックを使用することに注意してください。ただし、検出イベント メッセージに表示される検出ヘッダーは含まれていません。それらのヘッダー フィールドは、侵入イベント メッセージのヘッダー フィールドに似ています。次の図に続く表では、相関イベントのレコード ヘッダー フィールドを定義しています。



将来使用 （イベントのみ、メタデータ レコードでは使用されません）
データ レコード ブロック シリーズ 1 ブロックを使用します（ディスカバリ（シリーズ 1）ブロック（4-63 ページ）を参照）。 ...

次の表では、相関イベント メッセージのレコード ヘッダーの各フィールドについて説明します。

表 2-10　相関イベント メッセージ レコード ヘッダーのフィールド

フィールド	データ タイプ	説明
Netmap ID	uint16	このフィールドの第 1 ビットは、ヘッダーがアーカイブ タイムスタンプを含む拡張ヘッダーであるかどうかを示すフラグです。残りの 15 ビットは、イベントが検出されたドメインの Netmap ID を含むオプションのフィールドです。このフィールドは、使用されていない場合は空のままです。Netmap ID は、メタデータで提供されるドメインにマップされます。
レコード タイプ	uint16	データ レコードのコンテンツ タイプを識別します。侵入、相関、およびメタデータのレコード タイプのリストについては、表 3-1（3-1 ページ）を参照してください。
レコード長	uint32	レコード ヘッダーの後のメッセージのコンテンツの長さ。レコード ヘッダーの 8 または 16 バイトは含まれません。（レコード長 + レコード ヘッダーの長さは、メッセージ長と等しくなります。）
eStreamer サーバ タイムスタンプ（eStreamer Server Timestamp）	uint32	イベントが eStreamer サーバによってアーカイブされたときに適用されるタイムスタンプを示します。アーカイブ タイムスタンプとも呼ばれます。 要求メッセージ フラグにビット 23 が設定されている場合にのみ表示されるフィールド。 ホスト プロファイルやメタデータなど、Management Center によって生成されたデータの場合フィールドはゼロです。
将来使用	uint32	今後使用するために予約されています。 要求メッセージ フラグにビット 23 が設定されている場合にのみ表示されるフィールド。

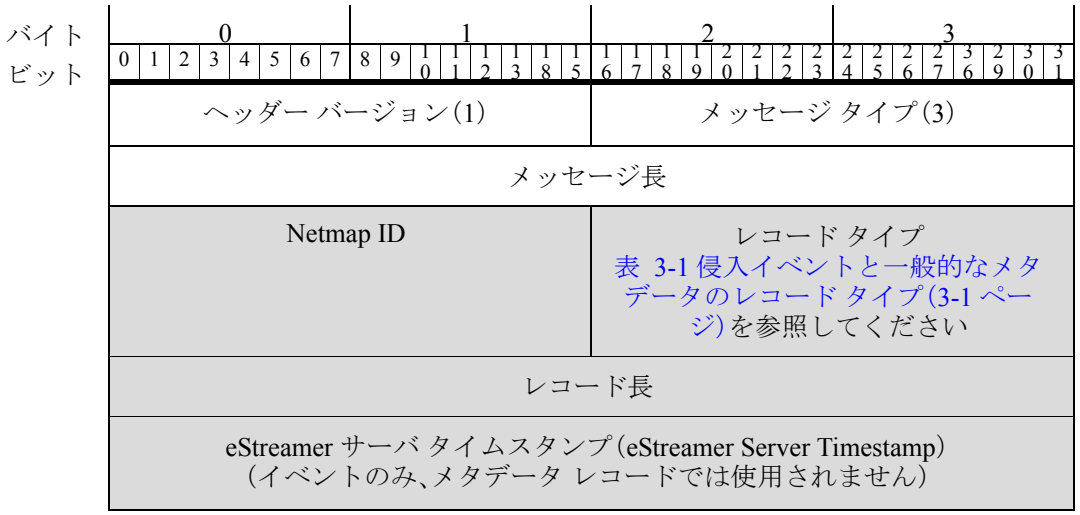
イベント追加データ メッセージの形式

次の図に、イベント追加データ メッセージの構造を示します。侵入イベント追加データ メッセージは、このメッセージ グループの例です。

イベント追加データ メッセージは、相関イベント メッセージと同じ形式で、レコード ヘッダーの直後にデータ ブロックがあります。相関メッセージとは異なり、シリーズ 1 データ ブロックではなくシリーズ 2 データ ブロックが使用され、個別のナンバリング シーケンスがあります。シリーズ 2 ブロックのタイプについては、シリーズ 2 のデータ ブロックの概要（3-58 ページ）を参照してください。

イベント追加データ メッセージのレコード ヘッダー

次の図の網掛け部分は、イベント追加データ メッセージのレコード ヘッダーのフィールドを示しています。その次にある表では、イベント追加データ メッセージのレコード ヘッダー フィールドを定義しています。



将来使用 （イベントのみ、メタデータ レコードでは使用されません）
データ レコード ブロック（Data Record Block） シリーズ 2 ブロックを使用します（シリーズ 2 のデータ ブロックの概要（3-58 ページ）を参照）。 ...

次の表では、イベント追加データ メッセージのレコード ヘッダーの各フィールドについて説明します。

表 2-11　イベント追加データ メッセージのレコード ヘッダー フィールド

フィールド	データ タイプ	説明
Netmap ID	uint16	このフィールドの第 1 ビットは、ヘッダーがアーカイブ タイムスタンプを含む拡張ヘッダーであるかどうかを示すフラグです。残りの 15 ビットは、イベントが検出されたドメインの Netmap ID を含むオプションのフィールドです。このフィールドは、使用されていない場合は空のままです。Netmap ID は、メタデータで提供されるドメインにマップされます。
レコード タイプ	uint16	データ レコードのコンテンツ タイプを識別します。イベント追加データ レコード タイプのリストについては、表 3-1 侵入イベントと一般的なメタデータのレコード タイプ（3-1 ページ）を参照してください。
レコード長	uint32	レコード ヘッダーの後のメッセージのコンテンツの長さ。レコード ヘッダーの 8 または 16 バイトは含まれません。（レコード長 + レコード ヘッダーの長さは、メッセージ長と等しくなります。）
eStreamer サーバ タイムスタンプ（eStreamer Server Timestamp）	uint32	イベントが eStreamer サーバによってアーカイブされたときに適用されるタイムスタンプを示します。アーカイブ タイムスタンプとも呼ばれます。 要求メッセージ フラグにビット 23 が設定されている場合にのみ表示されるフィールド。Management Center によって生成されたイベントの場合は、フィールドが存在しません。
将来使用	uint32	今後使用するために予約されています。 要求メッセージ フラグにビット 23 が設定されている場合にのみ表示されるフィールド。Management Center によって生成されたイベントの場合は、フィールドが存在しません。

データ ブロック ヘッダー

シリーズ 1 ブロックとシリーズ 2 ブロックは、構造は類似していますが、ナンバリングが異なります。これらのブロックは、検出、相関、接続、またはイベント追加データ メッセージのデータ部分のどこにでも置くことができます。これらのブロックは、複数のネスティング レベルで他のブロックをカプセル化します。

第 1 シリーズと第 2 シリーズの両方のデータ ブロックは、次の図に示すヘッダー構造で始まります。次の表に、ヘッダー フィールドに関する情報を示します。ヘッダーの直後には、データ ブロック タイプに関連付けられたデータ構造が続きます。

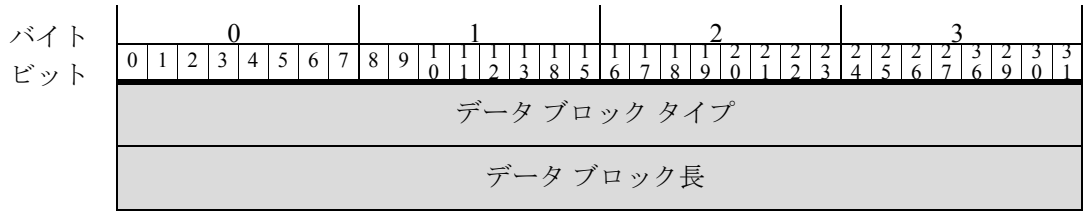


表 2-12

フィールド	データ タイプ	説明
データ ブロック タイプ	uint32	シリーズ 1 ブロックのタイプについては、ディスカバリ（シリーズ 1）ブロック（4-63 ページ）を参照してください。 シリーズ2ブロックのタイプについては、表 3-26 シリーズ 2 のブロック タイプ（3-58 ページ）を参照してください。
データ ブロック長	uint32	データ ブロックの長さ。2 つのデータ ブロック ヘッダー フィールドに 8 バイトを加えたデータのバイト数が含まれます。

ホスト要求メッセージの形式

ホスト プロファイルを受信するには、ホスト要求メッセージを送信します。IP アドレス範囲で定義された単一のホストまたは複数のホストのデータを要求できます。

イベント ストリーム要求メッセージを送信することによって、ホスト プロファイル情報の要求を含むすべてのデータ要求で最初にセッションを初期化することが必須であることに注意してください。ホスト データをストリーミングするだけのために設定するには、最初のイベント ストリーム要求メッセージで次のいずれかの要求フラグ設定を使用できます。

- 適切なバージョンのメタデータのビットを設定する（これは、ホスト データをストリーミングする場合に有益です）
- 要求フラグを設定しない
- ビット 11 を設定する（eStreamer のレガシー バージョンを使用する場合は、デフォルトのイベント ストリーミングを抑制するため）

最初のメッセージの後、ホスト要求メッセージ（タイプ 5）を使用してホストを指定します。

（注） デフォルトのイベント ストリーミングを使用するレガシー eStreamer バージョンの場合、ホスト プロファイル データのみをストリーミングする場合は、デフォルトのイベント メッセージを抑制する必要があります。最初に、要求フラグ フィールドのビット 11 を 1 に設定したイベント ストリーム要求メッセージをサーバに送信します。その後、ホスト要求メッセージを送信します。

次の図に、ホスト要求メッセージの形式を示します。網掛けのフィールドはホスト要求メッセージの形式に固有であり、次の表で定義されています。上記の 3 つのフィールドは、標準のメッセージ ヘッダーです。

次の表では、メッセージ フィールドについて説明します。

表 2-13　ホスト要求メッセージ フィールド

フィールド	データ タイプ	説明
データ タイプ	uint32	次のコードを使用して、単一のホストまたは複数のホストのデータを要求します。 • 0:単一ホストのバージョン 3.5 ～ 4.6。 • 1:複数のホストのバージョン 3.5 ～ 4.6(ブロック 34 を使用)。 • 2:単一ホストのバージョン 4.7 ～ 4.8(ブロック 47 を使用)。 • 3:複数のホストのバージョン 4.7 ～ 4.8(ブロック 47 を使用)。 • 4:単一ホストのバージョン 4.9 ～ 4.10(ブロック 92 を使用)。 • 5:複数のホストのバージョン 4.9 ～ 4.10(ブロック 92 を使用)。 • 6:単一ホストのバージョン 5.0.x データ(ブロック 111 を使用。フル ホスト プロファイル データ ブロック 5.0 ～ 5.0.2(B-291 ページ)を参照してください)。 • 7:複数ホストのバージョン 5.0.x データ(ブロック 111 を使用。フル ホスト プロファイル データ ブロック 5.0 ～ 5.0.2(B-291 ページ)を参照してください)。 • 8:複数ホストのバージョン 5.1.x データ(ブロック 111 を使用。フル ホスト プロファイル データ ブロック 5.1.1(B-301 ページ)を参照してください)。 • 9:複数ホストのバージョン 5.1.x データ(ブロック 111 を使用。フル ホスト プロファイル データ ブロック 5.1.1(B-301 ページ)を参照してください)。 • 10:ルール ドキュメンテーション データ(ブロック 27 を使用。ルール ドキュメンテーションのメッセージ形式(2-31 ページ)を参照してください)。 • 11:複数ホストのバージョン 5.2x データ(ブロック 111 を使用。フル ホスト プロファイル データ ブロック 5.2.x(B-312 ページ)を参照してください)。 • 12:複数ホストのバージョン 5.2.x データ(ブロック 111 を使用。フル ホスト プロファイル データ ブロック 5.2.x(B-312 ページ)を参照してください)。 • 13:複数ホストのバージョン 5.3+ データ(ブロック 111 を使用。全ホスト プロファイル データ ブロック 5.3+(5-1 ページ)を参照してください)。 • 14:複数ホストのバージョン 5.3+ データ(ブロック 111 を使用。全ホスト プロファイル データ ブロック 5.3+(5-1 ページ)を参照してください)。
フラグ	32 ビット フィールド	• 0x00000001:ホスト プロファイルの [注(Notes)] フィールドが(Firepower システム に格納されているホストに関するユーザ定義の情報を使用して)読み込まれます。 • 0x00000002:サービス ブロックの [バナー(Banner)] フィールドが(サービスについて検出された最初のパケットの最初の 256 バイトを使用して)読み込まれます。バナーはデフォルトでは無効になっており、設定されている場合にのみ使用できます。

表 2-13　ホスト要求メッセージ フィールド(続き)

フィールド	データ タイプ	説明
開始 IP アドレス	uint8[4]	データを返す必要があるホストの IP アドレス(要求が単一ホストに対する場合)、または IP アドレス範囲の開始アドレス(要求が複数のホストに対する場合)。IP アドレス オクテットでアドレスを指定します。
終了 IP アドレス	uint8[4]	IP アドレス範囲の終了アドレス(要求が複数のホストに対する場合)、または開始 IP アドレスの値(要求が単一ホストに対する場合)。

次の図に、レガシーのホスト要求メッセージの形式を示します。eStreamer は引き続きこの要求に応答します。現在の要求との唯一の違いは、IPv4 アドレス フィールドが小さいという点です。網掛けのフィールドはホスト要求メッセージの形式に固有であり、次の表で定義されています。上記の 3 つのフィールドは、標準のメッセージ ヘッダーです。

次の表では、メッセージ フィールドについて説明します。

表 2-14　ホスト要求メッセージ フィールド

フィールド	データ タイプ	説明
データ タイプ	uint32	次のコードを使用して、単一のホストまたは複数のホストのデータを要求します。 • 0:単一ホストのバージョン 3.5 ～ 4.6。 • 1:複数のホストのバージョン 3.5 ～ 4.6(ブロック 34 を使用)。 • 2:単一ホストのバージョン 4.7 ～ 4.8(ブロック 47 を使用)。 • 3:複数のホストのバージョン 4.7 ～ 4.8(ブロック 47 を使用)。 • 4:単一ホストのバージョン 4.9 ～ 4.10(ブロック 92 を使用)。 • 5:複数のホストのバージョン 4.9 ～ 4.10(ブロック 92 を使用)。 • 6:単一ホストのバージョン 5.0+ データ(ブロック 111 を使用、全ホスト プロファイル データ ブロック 5.3+(5-1 ページ)を参照)。 • 7:複数のホストのバージョン 5.0+ データ(ブロック 111 を使用、全ホスト プロファイル データ ブロック 5.3+(5-1 ページ)を参照)。
フラグ	32 ビット フィールド	• 0x00000001:ホスト プロファイルの [注(Notes)] フィールドが(Firepower システム に格納されているホストに関するユーザ定義の情報を使用して)読み込まれます。 • 0x00000002:サービス ブロックの [バナー(Banner)] フィールドが(サービスについて検出された最初のパケットの最初の 256 バイトを使用して)読み込まれます。バナーはデフォルトでは無効になっており、設定されている場合にのみ使用できます。
開始 IP アドレス	uint8[4]	データを返す必要があるホストの IP アドレス(要求が単一ホストに対する場合)、または IP アドレス範囲の開始アドレス(要求が複数のホストに対する場合)。IP アドレス オクテットでアドレスを指定します。
終了 IP アドレス	uint8[4]	IP アドレス範囲の終了アドレス(要求が複数のホストに対する場合)、または開始 IP アドレスの値(要求が単一ホストに対する場合)。

ルール ドキュメンテーションのメッセージ形式

ルール ドキュメンテーション プロファイルを受信するには、ルール ドキュメンテーション メッセージを送信します。ジェネレータ ID、署名 ID、およびリビジョンでこれらを要求します。

ルール ドキュメンテーション情報の要求を含むすべてのデータ要求では、イベント ストリーム要求メッセージを送信することで、最初にセッションを初期化しておく必要があります。ホスト データをストリーミングするだけのために設定するには、最初のイベント ストリーム要求メッセージで次のいずれかの要求フラグ設定を使用できます。

- 適切なバージョンのメタデータのビットを設定する(これは、ホスト データをストリーミングする場合に有益です)
- 要求フラグを設定しない
- ビット 11 を設定する(eStreamer のレガシー バージョンを使用する場合は、デフォルトのイベント ストリーミングを抑制するため)

最初のメッセージの後、ルール ドキュメンテーション メッセージ（タイプ 10）を使用してルールを指定します。

以下のグラフィックに、ルール ドキュメンテーション メッセージの形式を示します。網掛けされたフィールドは、ルール ドキュメンテーションのメッセージ形式に固有です。これを次の表で定義します。上記の 3 つのフィールドは、標準のメッセージ ヘッダーです。

次の表では、メッセージ フィールドについて説明します。

表 2-15　ルール ドキュメンテーション メッセージ フィールド

フィールド	データ タイプ	説明
データ タイプ	uint32	ルール ドキュメンテーション データ ブロックのデータを要求します。この値は常に 10 です。5.2 以上のルール ドキュメントのデータ ブロック (3-110 ページ) を参照してください。
フラグ	32 ビット フィールド	• 0x00000001: ルール ドキュメンテーション データ ブロックの [注記 (Notes)] フィールドに Firepower システム に格納されているホストに関するユーザ定義の情報が読み込まれます。 • 0x00000002: サービス ブロックの [バナー (Banner)] フィールドが (サービスについて検出された最初のパケットの最初の 256 バイトを使用して) 読み込まれます。バナーはデフォルトでは無効になっており、設定されている場合にのみ使用できます。
シグネチャ ID	uint32	要求したルールの ID 番号。
ジェネレータ ID	uint32	要求したルールの Firepower システム プリプロセッサの ID 番号。
ルール リビジョン	uint32	ルール リビジョン番号。
予約済み	uint8[20]	このフィールドは現在使用されていません。

ホスト データおよびマルチ ホスト データ メッセージの形式

eStreamer は、完全なホスト プロファイル データ ブロックをそれぞれ含む、ホスト データ メッセージを送信することによって、ホスト要求に応答します。eStreamer は、要求で指定された各ホストに対し 1 つのホスト データ メッセージを送信します。eStreamer は、タイプ 6 のメッセージを使用して単一のホスト プロファイルの要求に応答し、タイプ 7 のメッセージを使用して複数のホストの要求に応答します。タイプ 6 およびタイプ 7 のメッセージの形式は同一であり、メッセージ タイプのみが異なります。

ホスト データ メッセージには、レコード タイプ フィールドはありません。メッセージの構造は、メッセージ タイプと、メッセージに含まれる完全なホスト プロファイルのデータ ブロック タイプによって伝達されます。完全なホスト プロファイル データ ブロックは、一連のブロックのグループです。

次の図はホスト データ メッセージの形式を示しており、その次の表では網掛けフィールドを定義しています。

ホスト要求メッセージに固有のフィールドは次のとおりです。

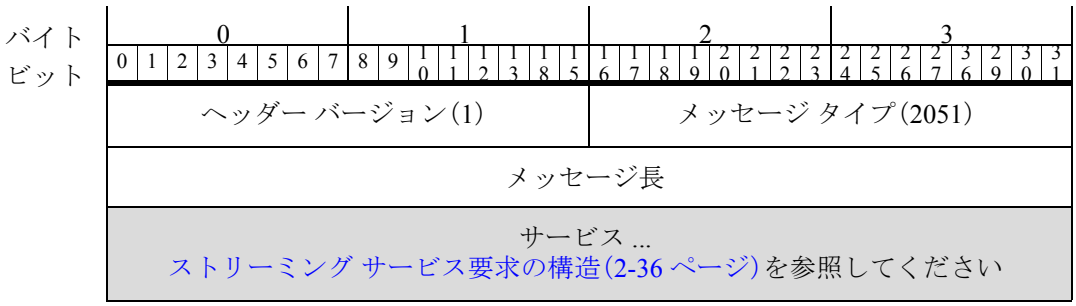
表 2-16

フィールド	データ タイプ	説明
完全なホスト プロファイル データ ブロック タイプ	uint32	メッセージに含まれる完全なホスト プロファイル データのブロック タイプを指定します。表 4-30 ホスト ディスカバリと接続データ ブロック タイプ(4-64 ページ)を参照してください。
長さ(Length)	uint32	メッセージ内の完全なホスト プロファイル データの長さ。
完全なホスト プロファイル データ ブロック(Full Host Profile Data Block)	変数(variable)	ホストのデータ。現在の完全なホスト プロファイル データ ブロックの定義へのリンクについては、表 4-30 ホスト ディスカバリと接続データ ブロック タイプ(4-64 ページ)を参照してください。

ストリーミング情報メッセージの形式

eStreamer サービスは、拡張要求の要求を受信すると、以下に説明するストリーミング情報メッセージをクライアントに送信します。このメッセージは、サーバの使用可能なサービスのリストをアドバタイズします。現在、関連する唯一のオプションは eStreamer サービス(6667)ですが、メッセージには他のサービスがリストされる場合があり、それらは無視する必要があります。アドバタイズされた各サービスは、ストリーミング サービス要求の構造(2-36 ページ)で説明するストリーミング サービス要求構造によって表されます。

次の図に、ストリーミング情報メッセージの形式を示します。網掛けのフィールドは、このメッセージ タイプに固有のものです。上記の 3 つのフィールドは、標準のメッセージ ヘッダーです。



ストリーミング情報メッセージのフィールドは次のとおりです。

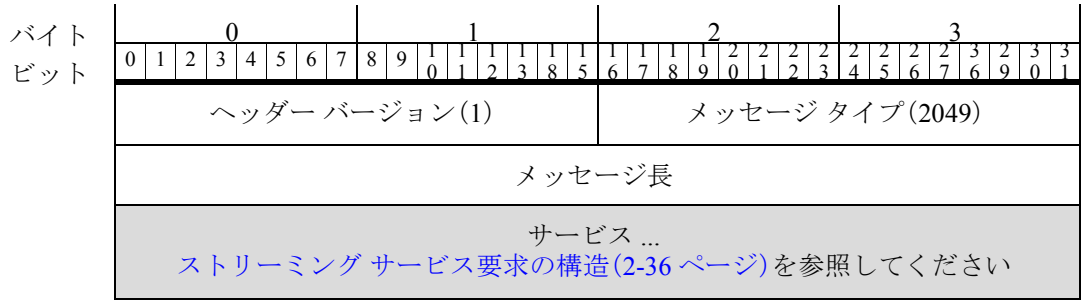
表 2-17　ストリーミング情報メッセージのフィールド

フィールド	データ タイプ	説明
ヘッダー バージョン	uint16	1 に設定します。
Message Type	uint16	eStreamer メッセージ タイプ。ストリーミング要求メッセージの場合は 2051 に設定します。
メッセージ長	uint32	メッセージ ヘッダーの後のメッセージのコンテンツの長さ。[ヘッダー バージョン(Header Version)]、[メッセージ タイプ(Message Type)]、および [メッセージ長(Message Length)] フィールドのバイトは含まれません。
サービス [] (Service[])	アレイ	使用できるサービスのリスト。ストリーミング サービス要求の構造(2-36 ページ)を参照してください。

ストリーミング要求メッセージの形式

クライアントは、ストリーミング要求メッセージを使用して、使用するストリーミング情報メッセージで eStreamer サービスに指定し、その後にストリーミングされるイベント タイプおよびバージョンの要求のセットを指定します。次の図はメッセージの構造を示し、次の表ではフィールドを定義しています。要求されたサービスは、ストリーミング サービス要求の構造(2-36 ページ)で説明するストリーミング サービス要求構造によって表されます。

次の図に、ストリーミング情報メッセージの形式を示します。網掛けのフィールドは、このメッセージ タイプに固有のものです。上記の 3 つのフィールドは、標準のメッセージ ヘッダーです。



ストリーミング要求メッセージのフィールドは次のとおりです。

表 2-18　ストリーミング要求メッセージのフィールド

フィールド	データ タイプ	説明
ヘッダー バージョン	uint16	1 に設定します。
Message Type	uint16	eStreamer メッセージ タイプ。ストリーミング要求メッセージの場合は 2049 に設定します。
メッセージ長	uint32	メッセージ ヘッダーの後のメッセージのコンテンツの長さ。[ヘッダー バージョン（Header Version）]、[メッセージ タイプ（Message Type）]、および [メッセージ長（Message Length）] フィールドのバイトは含まれません。
サービス [] （Service[]）	アレイ	要求されたサービス構造のリスト。ストリーミング サービス要求の構造（2-36 ページ）を参照してください。

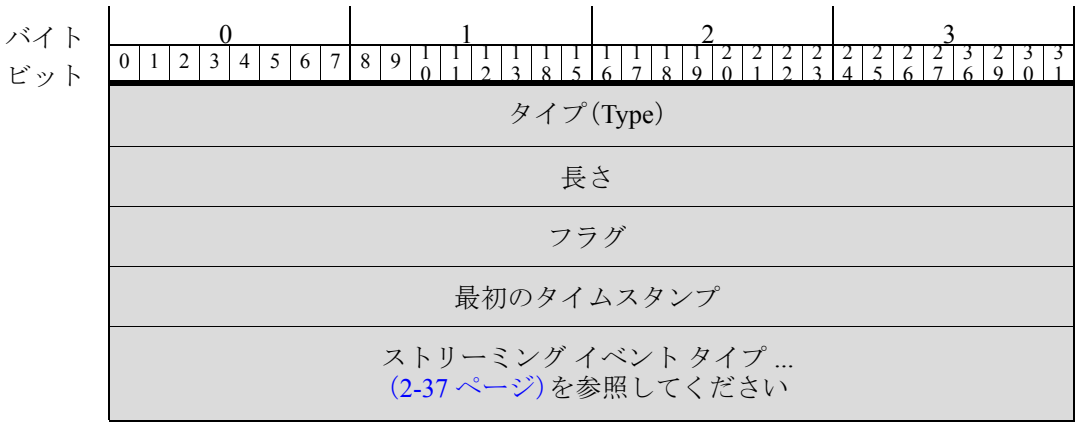
ストリーミング サービス要求の構造

eStreamer サービスは、アドバタイズする各サービスについて、ストリーミング情報メッセージで 1 つのストリーミング サービス要求のデータ構造を送信します。eStreamer サービスは、ストリーミング サービス要求の最後のフィールドを使用しません。このフィールドは、含まれる予定のイベント タイプのリストを規定します。

クライアントは、eStreamer からのストリーミング サービス要求構造を処理し、サーバに返す応答で同じ構造を使用します。クライアントがサーバに送信するストリーミング サービス要求には、最初に、eStreamer によってアドバタイズされるサービスに対する要求が含まれ、2 番目に、クライアントが受信する要求されたイベント タイプを指定するストリーミング イベント タイプ構造のリストが含まれます。

各ストリーミング イベント タイプ構造には、要求された各イベント タイプのイベント タイプとバージョンを指定する 2 つのフィールドが含まれています。ストリーミング イベント タイプの構造については、（2-37 ページ）を参照してください。

次の図に、ストリーミング サービス要求構造のフィールドを示します。その次にある表では、フィールドを定義しています。



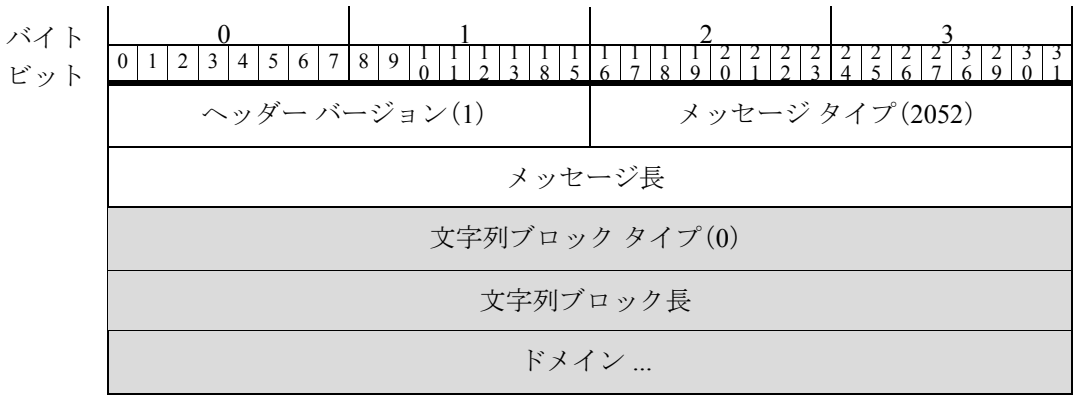
ストリーミング サービス要求構造のフィールドは次のとおりです。

表 2-19　ストリーミング サービス要求フィールド

フィールド	データ タイプ	説明
タイプ（Type）	uint32	[サービス ID（Service ID）]。 eStreamer サーバ メッセージでは、これによって利用可能なサービスがアドバタイズされます。 クライアント メッセージでは、要求されたサービスが指定されます。 現在の有効なオプション： • 6667（eStreamer サービスの場合）
長さ（Length）	uint32	サービス要求の長さ。タイプと長さを含むサービス要求の長さを表します。 長さには、メッセージ内のすべてのストリーミング イベント タイプのレコードと、終端レコードを含める必要があることに注意してください。
フラグ	uint32	eStreamer のストリーミング情報メッセージ：常に 0。 クライアントのストリーミング要求メッセージ：元のイベント ストリーム要求メッセージのフラグ設定を複製します。
最初のタイムスタンプ	uint32	eStreamer のストリーミング情報メッセージ：常に 0。 クライアントのストリーミング要求メッセージ：元のイベント ストリーム要求メッセージのタイムスタンプを複製します。
ストリーミング イベント タイプ	アレイ	eStreamer のストリーミング情報メッセージ： • 今後使用するために予約されています。0 の長さが含まれています。 クライアントのストリーミング要求メッセージ： • 各要求されたイベント タイプの 1 つのストリーミング イベント タイプ エントリ。(2-37 ページ) を参照してください。 • [イベント タイプ] と [バージョン（Version）] を両方とも 0 に設定して、0 のイベント タイプ エントリを含む要求リストを終了します。 (2-37 ページ) を参照してください。

ドメイン ストリーミング要求メッセージの形式

クライアントは、ドメイン ストリーミング要求メッセージを使用して、eStreamer の特定のドメインからのイベントを要求します。次の図はメッセージの構造を示し、次の表ではフィールドを定義しています。網掛けのフィールドは、このメッセージ タイプに固有のものです。上記の 3 つのフィールドは、標準のメッセージ ヘッダーです。



ドメイン ストリーミング要求メッセージのフィールドは次のとおりです。

表 2-20　ドメイン ストリーミング要求メッセージのフィールド

フィールド	データ タイプ	説明
ヘッダー バージョン	uint16	1 に設定します。
Message Type	uint16	eStreamer メッセージ タイプ。ドメイン ストリーミング要求メッセージの場合は 2052 に設定します。
メッセージ長	uint32	メッセージ ヘッダーの後のメッセージのコンテンツの長さ。[ヘッダー バージョン(Header Version)]、[メッセージ タイプ(Message Type)]、および [メッセージ長(Message Length)] フィールドのバイトは含まれません。
文字列ブロック タイプ	uint32	ドメインを含む文字列データ ブロックを開始します。この値は常に 0 です。
文字列ブロック長	uint32	ドメイン文字列データ ブロックに含まれるバイト数。ブロック タイプおよびヘッダー フィールドの 8 バイトにドメイン内のバイト数を加えたものです。
ドメイン	string	ストリーミング イベントの要求元のドメイン。空白のままにすると、サービスはクライアントがアクセスするすべてのドメインのイベントをストリーミングします。

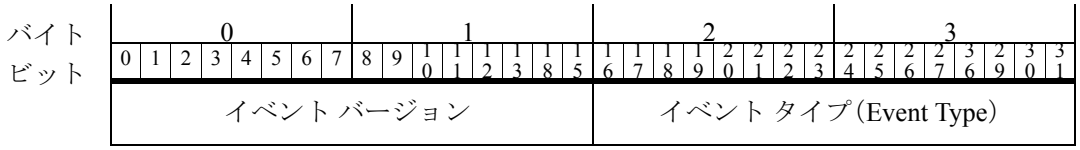
ストリーミング イベント タイプの構造

eStreamer クライアントは、ストリーミング イベント タイプ構造を使用して、イベントのバージョンとバージョンを指定します。各イベント バージョンとタイプの組み合わせは、イベント ストリームの要求です。

ストリーミング イベント タイプ構造のリストは、すべてのフィールドがゼロに設定された構造で終了する必要があります。具体的な場所は次のとおりです。

```
イベント バージョン = 0
イベント タイプ = 0
```

次の図に、ストリーミング イベント タイプ構造の形式を示します。



ストリーミング イベント タイプ構造のフィールドは次のとおりです。

表 2-21　ストリーミング イベント タイプのフィールド

フィールド	データ タイプ	説明
イベント バージョン	uint16	イベント タイプのバージョン番号。各イベント タイプでサポートされているバージョンのリストについては、表 2-22 拡張要求のイベント タイプとバージョン（2-40 ページ）を参照してください。
イベント タイプ（Event Type）	uint16	要求されたイベント タイプのコード。有効なイベント タイプとバージョン コードの現在のリストについては、表 2-22 拡張要求のイベント タイプとバージョン（2-40 ページ）を参照してください。 イベント タイプのリストは、ゼロのイベント タイプとゼロのイベント バージョンで終了する必要があります。

次の表に、クライアントが拡張要求で指定できるイベントのタイプとバージョンを示します。表には、各イベント タイプのバージョンに対応する Management Center のソフトウェア バージョンが示されています。たとえば、バージョン 4.8.0.2 ～ 4.9.1 で Management Center によってサポートされていた相関イベントを要求するには、イベント タイプ 31、バージョン 5 を要求する必要があります。イベントが異なるイベント タイプで記録されていた場合は、要求されたイベント タイプの形式に一致するようにアップグレードまたはダウングレードされます。

表 2-22　拡張要求のイベント タイプとバージョン

要求内容	使用するイベント バージョン番号	使用するイベント コード
侵入イベント	1:4.8.x 以前 2:4.9 ～ 4.10.x 3:5.0 ～ 5.1 4:5.1.1.x 5:5.2.x 6:5.3 7:5.3.1 8:5.4.x 9:6.0+	12
メタデータ	1:3.2 ～ 4.5.x 2:4.6.0.x 3:4.6.1 ～ 4.6.x 4:4.7+	21
相関およびコンプライアンスのホワイト リスト イベント	1:3.2 以前 2:4.0 ～ 4.4.x 3:4.5 ～ 4.6.1 4:4.7 ～ 4.8.0.1 5:4.8.0.2 ～ 4.9.1.x 6:4.10.0 ～ 4.10.x 7:5.0 ～ 5.0.2 8:5.1 ～ 5.3.x 9:5.4+	31
検出イベント	1:3.2 以前 2:3.0 ～ 3.4.x 3:3.5 ～ 4.6.x 4:4.7 ～ 4.8.x 5:4.9.0.x 6:4.9.1 ～ 4.9.x.x 7:4.10.0 ～ 4.10.x 8:5.0.x 9:5.1.x 10:5.2 ～ 5.3 11:5.3.1+	61

表 2-22　拡張要求のイベント タイプとバージョン(続き)

要求内容	使用するイベント バージョン番号	使用するイベントコード
接続イベント	1:4.0 ～ 4.1 3:4.5 ～ 4.6.1 4:4.7 ～ 4.9.0.x 5:4.9.1 ～ 4.10.x 6:5.0.x 7:5.1.0.x 8:5.1.1.x 9:5.2.x 10:5.3 11:5.3.1 12:5.4 13:5.4.0.1 ～ 5.4.0.2 14:6.0.x 15:6.1.x 16:6.2+	71
ユーザ イベント	1:4.7 ～ 4.10.x 2:5.0.x 3:5.1 ～ 5.1.x 4:5.2 5:6.0 6:6.1 7:6.2+	91
マルウェア イベント	1:5.1.0.x 2:5.1.1.x 3:5.2.x 4:5.3 5:5.3.1 6:5.4.x 7:6.0+	101
ファイル イベント	1:5.1.1 ～ 5.1.x 2:5.2.x 3:5.3 4:5.3.1 5:5.4.x 6:6.0+	111
影響相関イベント	1:5.2.x 以前 2:5.3+	131
リスト内の終了イベント タイプ	0	0

拡張要求メッセージの例

ストリーミング情報メッセージ

次の例では、サーバは 2 つのサービス、第 1 のタイプ 6667（eStreamer）と第 2 のタイプ 5000 をアドバタイズします。サーバからのストリーミング情報メッセージでは、[フラグ（flags）] フィールドと [最初のタイムスタンプ（initial timestamp）] フィールドはゼロであり、メッセージではイベント タイプは指定されていません。

表 2-23

ヘッダー バージョン:	1	/*always 1*/
メッセージ タイプ:	2051	/*streaming info msg*/
メッセージ長	32	/*bytes of msg content*/
サービス [1]. タイプ	6667	/*eStreamer service ID*/
サービス [1]. 長さ	8	
サービス [1]. フラグ	0	/*no flags from server*/
サービス [1]. 最初のタイムスタンプ	0	/*always 0*/
サービス [2]. タイプ	5000	/*service-2 ID*/
サービス [2]. 長さ	8	
サービス [2]. フラグ	0	/*no flags from server*/
サービス [2]. 最初のタイムスタンプ	0	/*always 0*/
ヘッダー バージョン:	1	/*always 1*/
メッセージ タイプ:	2051	/*streaming info msg*/

ストリーミング要求メッセージ

以下は、クライアントがサービス タイプ 6667（eStreamer）を要求し、接続イベントのバージョン 6（イベント タイプ 71）とメタデータのバージョン 4（イベント タイプ 21）の 2 つのイベント タイプを指定するストリーミング要求メッセージです。

表 2-24

ヘッダー バージョン:	1	/*always 1*/
メッセージ タイプ:	2049	/*stream request msg*/
メッセージ長	36	/*payload bytes*/
サービス [1]. タイプ	6667	/*eStreamer service ID*/
サービス [1]. 長さ	20	
サービス [1]. フラグ	30	/*original flags value*/
サービス [1]. 最初のタイムスタンプ	0	/*original timestamp*/
サービス [1]. イベント [1]. バージョン	6	/*version 6*/
サービス [1]. イベント [1]. タイプ	71	/*connection events*/
サービス [1]. イベント [2]. バージョン	4	/* version 4*/
サービス [1]. イベント [2]. タイプ	21	/*metadata*/
サービス [1]. イベント [3]. バージョン	0	/*terminate event list*/
サービス [1]. イベント [3]. タイプ	0	/*terminate event list*/

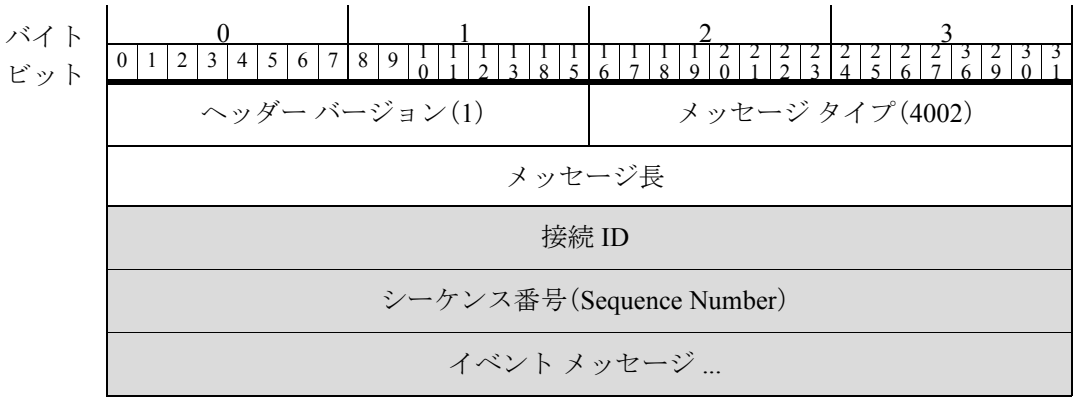
メッセージ バンドルの形式

クライアントが拡張要求を送信すると、eStreamer サーバはバンドル形式でメッセージを送信します。

クライアントはヌル メッセージで応答し、バンドル全体の受信の確認応答を行います。クライアントは、バンドル内の個々のメッセージの受信を確認応答するべきではありません。

メッセージ バンドルのメッセージ タイプは 4002 です。

次の図に、メッセージ バンドルの構造を示します。網掛けのフィールドは、バンドル メッセージ タイプに固有のものです。次の表に、フィールドとデータ構造の内容を示します。



メッセージ バンドル メッセージのフィールドは次のとおりです。

表 2-25　メッセージ バンドル メッセージのフィールド

フィールド	データ タイプ	説明
ヘッダー バージョン	uint16	常に 1 です。
Message Type	uint16	常に 4002 です。
メッセージ長	uint32	メッセージ ヘッダーの後のメッセージのコンテンツの長さ。バンドルの [ヘッダー バージョン（Header Version）]、[メッセージ タイプ（Message Type）]、および [メッセージ長（Message Length）] フィールドのバイトは含まれません。 クライアントがバンドルからメッセージをロードするとき、このフィールドの長さからメッセージのトータル長（ヘッダーを含む）を差し引くことができます。残りの部分が正数であれば、処理するメッセージがさらにあります。
接続 ID	uint32	サーバとの接続用の一意の識別子。
シーケンス番号（Sequence Number）	uint32	1 から始まり、eStreamer サーバによって送信された各バンドルに対して 1 ずつ増分します。
イベント メッセージ []	アレイ	バンドル内のサーバによってストリーミングされたイベント。各メッセージには、メッセージのバージョン番号（1）、要求された場合はアーカイブ タイムスタンプなど、フルセットのヘッダーがあります。

メタデータについて

eStreamer サーバは、要求されたイベント レコードとともにメタデータを提供できます。メタデータを受信するには、明示的に要求する必要があります。特定のバージョンのメタデータを要求する方法については、表 2-6 要求フラグ（2-13 ページ）を参照してください。メタデータは、イベント レコードのコードおよび数値識別子のコンテキスト情報を提供します。たとえば、侵入イベントには検出デバイスの内部識別子のみが含まれ、メタデータはデバイスの名前を提供します。

メタデータの伝送

要求メッセージがメタデータを指定する場合、eStreamer は関連するメタデータ レコードを送信してから、関連するイベント レコードを送信します。

eStreamer は、クライアントに送信したメタデータを追跡し、同じメタデータ レコードを再送しません。クライアントは、受信した各メタデータ レコードをキャッシュする必要があります。eStreamer は、あるセッションから次のセッションへのメタデータ送信の履歴を保持しないため、新しいセッションが開始され、要求メッセージがメタデータを指定すると、eStreamer は最初からメタデータのストリーミングを再スタートします。