



eStreamer の設定

クライアントアプリケーションを作成したら、ユーザはそれを eStreamer サーバに接続し、eStreamer サービスを開始して、データのやりとりを始めることができます。



コメント

eStreamer サーバとは、eStreamer サービスが実行されている Management Center または管理対象デバイス(バージョン 4.9 以降)です。

eStreamer とクライアントのインタラクションを管理するには、次のタスクを実行します。

1. eStreamer サーバで eStreamer を有効にします。
eStreamer サーバへのアクセス許可、クライアントの追加、および認証された接続を確立するための認証クレデンシャルの生成の詳細については、「[eStreamer サーバでの eStreamer の設定 \(6-1 ページ\)](#)」を参照してください。
2. 必要に応じて、手動で eStreamer サービス (eStreamer) を実行します。サービスのステータスを停止、開始、および表示できます。また、コマンドライン オプションを使用して、クライアント/サーバ通信をデバッグできます。
詳細については、[eStreamer サービスの管理 \(6-4 ページ\)](#) を参照してください。
3. オプションとして、eStreamer 参照クライアントを使用して接続またはデータ ストリームをトラブルシューティングするには、クライアントの実行を予定しているコンピュータで参照クライアントを設定します。
[eStreamer 参照クライアントの設定 \(6-6 ページ\)](#) を参照してください。

eStreamer サーバでの eStreamer の設定

ライセンス:任意 (Any)

eStreamer サーバとして使用する Management Center または管理対象デバイスが、クライアントアプリケーションへのイベントのストリームを開始する前に、クライアントにイベントを送信するように eStreamer サーバを設定し、クライアントに関する情報を指定して、通信を確立するときに使用する認証クレデンシャルを生成する必要があります。これらのタスクはすべて、Management Center または管理対象デバイスのユーザ インターフェイスから実行できます。

詳細については、次の各項を参照してください。

- [eStreamer イベント タイプの設定 \(6-2 ページ\)](#)
- [eStreamer クライアントの認証の追加 \(6-3 ページ\)](#)

eStreamer イベント タイプの設定

ライセンス:任意 (Any)

eStreamer サーバはどのタイプのイベントを要求するクライアント アプリケーションに送信できるかを制御できます。

管理対象デバイスまたは Management Center で使用可能なイベント タイプは、以下のとおりです。

- 侵入イベント
- 侵入イベント パケット データ
- 侵入イベント追加データ

次のものを含む Management Center で使用可能なイベントのタイプ:

- 検出イベント(これも、接続イベントを有効にします)
- 相関およびホワイトリスト イベント
- 影響フラグ アラート
- ユーザ アクティビティ イベント
- マルウェア イベント
- ファイル イベント

スタック構成 3D9900 ペアのプライマリとセカンダリは、それらが別の管理対象デバイスであるかのように、Management Center に侵入イベントを報告することに注意してください。3D9900 スタックのプライマリで eStreamer クライアントとの通信を設定する場合は、セカンダリでもクライアントを設定する必要があります。クライアント設定は複製されません。同様に、クライアントを削除する場合は、両方で削除します。スタック構成で 3D9900 を管理する Management Center に eStreamer クライアントを設定する場合は、同じイベントが両方によって報告されても、両方の管理対象デバイスから受信するすべてのイベントは Management Center が報告することに注意してください。

高可用性の構成の Management Center で eStreamer クライアントを設定する場合は、クライアントの設定は、プライマリの Management Center からセカンダリの Management Center に複製されません。

eStreamer によってキャプチャされるイベントのタイプを設定する方法:

アクセス:管理

-
- ステップ 1 [システム (System)] > [統合 (Integration)] > [eStreamer (eStreamer)] を選択します。。
- ステップ 2 **eStreamer** をクリックします。
[] ページには、[イベント設定 (eStreamer Event Configuration)] メニューが表示されます。
eStreamer
- ステップ 3 eStreamer でキャプチャし、要求するクライアントに転送するイベントのタイプの横にあるチェックボックスを選択します。チェックボックスが現在オフにされている場合は、データはキャプチャされていないことに注意してください。チェックボックスをオフにしても、すでにキャプチャされたデータは削除されません。

Management Center または管理対象デバイスで、次のいずれかまたはすべてを選択できます。

- [侵入イベント (Intrusion Events)]: 管理対象デバイスによって生成された侵入イベントを送信します。
- [侵入イベント パケット データ (Intrusion Event Packet Data)]: 侵入イベントに関連付けられたパケットを送信します。
- [侵入イベント 追加データ (Intrusion Event Extra Data)]: HTTP プロキシまたはロードバランサ経由で Web サーバに接続しているクライアントの発信元 IP アドレスに関連付けられている URL など、侵入イベントに関連付けられた追加データを送信します。

Management Center で、次のいずれかまたはすべてを選択できます。

- [検出イベント (Discovery Events)]: ホスト検出イベントを送信します。
- [関連イベント (Correlation Events)]: 関連イベントおよびホワイトリスト イベントを送信します。
- [影響フラグ アラート (Impact Flag Alerts)]: Management Center によって生成される影響アラートを送信します。
- [ユーザ アクティビティ イベント (User Activity Events)]: ユーザ イベントを送信します。
- [侵入イベント 追加データ (Intrusion Event Extra Data)]: HTTP プロキシまたはロードバランサ経由で Web サーバに接続しているクライアントの発信元 IP アドレスに関連付けられている URL など、侵入イベントの追加データを送信します。



コメント

これは、eStreamer サーバが送信できるイベントを制御することに注意してください。クライアントアプリケーションは、ユーザが受信する必要があるイベントのタイプを明確に要求する必要があります。詳細については、[要求フラグ \(2-13 ページ\)](#) を参照してください。

ステップ 4 [Save] をクリックします。

設定が保存され、選択したイベントが、要求時に、eStreamer クライアントに転送されます。

eStreamer クライアントの認証の追加

ライセンス: 任意 (Any)

eStreamer がクライアントにイベントを送信する前に、eStreamer サーバのピア データベースにクライアントを追加しておく必要があります。また、eStreamer サーバによって生成された認証証明書をクライアントにコピーする必要があります。

eStreamer クライアントを追加する方法:

アクセス: 管理

ステップ 1 [システム (System)] > [統合 (Integration)] > [eStreamer (eStreamer)] を選択します。

[eStreamer] ページが表示されます。

ステップ 2 [クライアントの作成 (Create Client)] をクリックします。

[クライアントの作成 (Create Client)] ページが表示されます。

ステップ 3 [ホスト名 (Hostname)] フィールドに、eStreamer クライアントを実行しているホストのホスト名または IP アドレスを入力します。



コメント

ホスト名を使用する場合は、ホスト入力サーバはホストを IP アドレスに解決できる必要があります。DNS 解決を設定していない場合、最初に設定するか、IP アドレスを使用する必要があります。

ステップ 4 証明書ファイルを暗号化するには、[パスワード (Password)] フィールドにパスワードを入力します。

ステップ 5 [保存 (Save)] をクリックします。

eStreamer サーバはクライアント コンピュータから Management Center 上のポート 8302 へのアクセスを許可し、クライアント/サーバ認証時に使用する認証証明書を作成します。新しいクライアントが [クライアント (eStreamer Client)] の下に表示された状態で、[クライアント (eStreamer Client)] ページが再表示されます。Management Center

ステップ 6 証明書ファイルの横にあるダウンロードアイコン(📄)をクリックします。

ステップ 7 SSL 認証のためにクライアント コンピュータが使用するディレクトリに証明書ファイルを保存します。

これで、クライアントは Management Center に接続できるようになりました。



ヒント

クライアントのアクセスを取り消すには、削除するホストの横にある削除アイコン(🗑️)をクリックします。Management Center でホスト入力サービスを再開する必要はありません。アクセスはただちに取り消されます。

eStreamer サービスの管理

ライセンス:任意 (Any)

eStreamer サービスはユーザ インターフェイスから管理できます。ただし、サービスを開始/停止する場合は、コマンドラインも使用できます。以降のセクションで eStreamer のコマンドライン オプションについて説明します。

- [eStreamer サービスの開始および停止 \(6-4 ページ\)](#) では、eStreamer サービスを開始および停止する方法を説明しています。
- [eStreamer サービスのオプション \(6-5 ページ\)](#) では、eStreamer サービスで使用可能なコマンドライン オプションとそれらを使用する方法について説明しています。

eStreamer サービスの開始および停止

ライセンス:任意 (Any)

eStreamer サービスは、サービスを開始、停止、リロード、および再開できる `manage_estreamer.pl` スクリプトを使用して管理できます。



ヒント

また、eStreamer の初期化スクリプトにコマンドライン オプションを追加することもできます。詳細については、[eStreamer サービスのオプション \(6-5 ページ\)](#) を参照してください。

次の表で、Management Center または管理対象デバイスで使用可能な `manage_estreamer.pl` スクリプトのオプションについて説明します。

表 6-1 eStreamer 管理オプション


| オプション | 説明 | 選択するオプション番号 |
|---------|------------------------|-------------|
| enable | サービスを開始します。 | 3 |
| disable | サービスを停止します。 | 2 |
| restart | サービスを再開します。 | 4 |
| status | サービスが実行されているかどうかを示します。 | 1 |

eStreamer サービスのオプション

ライセンス:任意(Any)

eStreamer には、サービスをトラブルシューティングすることを可能にする多くのサービス オプションが含まれています。次の表に記載されているオプションは、eStreamer サービスとともに使用できます。

表 6-2 eStreamer サービスのオプション

| オプション | 説明 |
|---------------|---|
| --debug | デバッグ レベル ログで eStreamer を実行します。エラーは <code>syslog</code> に保存され(--nodaemon とともに使用される際)、画面に表示されます。 |
| --nodaemon | フォアグラウンド プロセスとして eStreamer を実行します。エラーは画面上に表示されます。 |
| --nohostcheck | <p>ホスト名の確認を無効化して eStreamer を実行します。つまり、クライアントホスト名がクライアント証明書の <code>subjectAltName:dNSName</code> エントリに含まれているホスト名と一致しない場合も、アクセスは依然として許可されます。nohostcheck オプションは、ネットワーク DNS および NAT の設定が、正常なホスト名の確認を防げる場合に役立ちます。その他のセキュリティの確認はすべて実行されることに注意してください。</p> <p> 注意 このオプションを有効にすると、システムのセキュリティにマイナスに影響する可能性があります。</p> |

最初に eStreamer サービスを停止し、次に必要なオプションでサービスを実行し、最後にサービスを再開して、上記のオプションを使用します。たとえば、eStreamer の機能をデバッグするには、[デバッグ モードでの eStreamer サービスの実行\(6-6 ページ\)](#)に記載されている手順に従うことができます。

デバッグモードでの eStreamer サービスの実行

ライセンス:任意 (Any)

デバッグモードで eStreamer サービスを実行すると、サービスによって生成される各ステータスメッセージを端末画面に表示できます。デバッグを実行するには、次の手順を使用します。

デバッグモードでの eStreamer サービスの実行:

アクセス:管理

-
- ステップ 1 Management Center または管理対象デバイスに SSH を使用してログインします。
- ステップ 2 `manage_estreamer.pl` を使用して、オプション 2 を選択し、eStreamer サービスを停止します。
- ステップ 3 `./usr/local/sf/bin/sfestreamer --nodaemon --debug` を使用して、デバッグモードで eStreamer サービスを再開します。
- サービスのステータスメッセージが端末画面に表示されます。
- ステップ 4 デバッグを終了したら、`manage_estreamer.pl` を使用し、オプション 4 を選択して通常モードでサービスを再開します。
-

eStreamer 参照クライアントの設定

eStreamer SDK とともに提供される参照クライアントとは、eStreamer API の使用方法を示すために含まれているサンプルクライアントスクリプトおよび Perl モジュールのセットです。これらを実行して eStreamer の出力に習熟したり、これらを使用してカスタム設計クライアントのインストールの問題をデバッグしたりできます。

参照クライアントのセットアップの詳細については、以降の各項を参照してください。

- [eStreamer Perl 参照クライアントの設定 \(6-6 ページ\)](#)
- [eStreamer Perl 参照クライアントの実行 \(6-12 ページ\)](#)

eStreamer Perl 参照クライアントの設定

eStreamer Perl 参照クライアントを使用するには、まず環境と要件に合うようにサンプルスクリプトを設定する必要があります。

詳細については、次の項を参照してください。

- [eStreamer Perl 参照クライアントについて \(6-7 ページ\)](#)
- [eStreamer 参照クライアントの通信の設定 \(6-7 ページ\)](#)
- [Perl 参照クライアントのための一般的な前提条件のロード \(6-8 ページ\)](#)
- [Perl SNMP 参照クライアントのための前提条件のロード \(6-8 ページ\)](#)
- [テストスクリプトで要求されるデータについて \(6-8 ページ\)](#)
- [テストスクリプトで要求されるデータタイプの変更 \(6-10 ページ\)](#)
- [Perl 参照クライアントのための証明書の作成 \(6-11 ページ\)](#)

eStreamer Perl 参照クライアントについて

eStreamer Perl 参照クライアントを含む eStreamerSDK.zip パッケージは、[Cisco サポート サイト](#) からダウンロードできます。eStreamerSDK.zip パッケージには次のファイルが含まれています。

- SF_CUSTOM_ALERT.MIB

この MIB ファイルは、SNMP トラップを設定するために snmp.pm ファイルによって使用されます。

- SFRecords.pm

この Perl モジュールには、検出メッセージのレコードブロックの定義が含まれています。

- SFStreamer.pm

この Perl モジュールには、Perl クライアントが呼び出す関数が含まれています。

- SFPkcs12.pm

この Perl モジュールはクライアント証明書を解析し、クライアントが eStreamer サーバに接続できるようにします。

- SFRNABlocks.pm

この Perl モジュールには、検出データのブロックの定義が含まれています。

- ssl_test.pl

この Perl スクリプトは、SSL 接続を介した侵入イベント要求をテストするために使用できます。

- OutputPlugins/csv.pm

この Perl モジュールは、侵入イベントをカンマ区切り値の (CSV) の形式に出力します。

- OutputPlugins/print.pm

この Perl モジュールは、人間が解読可能な形式でイベントを出力します。

- OutputPlugins/snmp.pm

この Perl モジュールは、特定の SNMP サーバにイベントを送信します。

- OutputPlugins/pcap.pm

この Perl モジュールは、パケット キャプチャを pcap ファイルとして保存します。

- OutputPlugins/syslog.pm

この Perl モジュールは、ローカルの syslog サーバにイベントを送信します。

eStreamer 参照クライアントの通信の設定

参照クライアントは、データ通信にセキュア ソケット レイヤ (SSL) を使用します。クライアントとして使用する予定のコンピュータに OpenSSL をインストールし、環境に合わせて適切に設定する必要があります。



コメント

Linux のオペレーティングシステムの初期インストールの場合は、このダウンロードの一部として libssl-dev コンポーネントをインストールする必要があります。

クライアントでの SSL の設定:

-
- ステップ 1 OpenSSL を <http://openssl.org/source/> からダウンロードします。
 - ステップ 2 /usr/local/src にソースを展開します。
 - ステップ 3 Configure スクリプトを実行して、ソースを設定します。
 - ステップ 4 コンパイル対象のソースに Make を実行し、インストールします。
-

Perl 参照クライアントのための一般的な前提条件のロード

eStreamer Perl 参照クライアントを実行する前に、クライアント コンピュータに IO::Socket::SSL Perl モジュールをインストールする必要があります。モジュールは手動でインストールすることも、cpan を使用してインストールすることもできます。



コメント

クライアント コンピュータに Net::SSLeay モジュールがインストールされていない場合は、そのモジュールも同様にインストールします。Net::SSLeay は OpenSSL との通信に必要です。

eStreamer サーバへの SSL 接続をサポートするためには、OpenSSL もインストールし、設定する必要があります。詳細については、[eStreamer 参照クライアントの通信の設定 \(6-7 ページ\)](#) を参照してください。

Perl SNMP 参照クライアントのための前提条件のロード

Perl 参照クライアントの eStreamer SNMP モジュールを実行する前に、クライアント コンピュータのクライアント オペレーティング システムで使用可能な最新の net-snmp Perl モジュールをインストールする必要があります。

Perl 参照クライアントのダウンロードと展開

eStreamer Perl 参照クライアントを含む EventStreamerSDK.zip ファイルは、[Cisco サポート サイト](#) からダウンロードできます。

クライアントを実行する予定の Linux オペレーティング システムを実行しているコンピュータで zip ファイルを展開します。

テスト スクリプトで要求されるデータについて

デフォルトで、参照クライアントで `ssl_test -o` 設定を使用する際は、次の表に示すようにデータを要求します。

表 6-3 出力プラグインで作成されるデフォルト要求

| 構文 | プラグインの呼び出し | 送信内容 | 要求するデータ |
|---|------------|---|---|
| <code>./ssl_test.pl eStreamerServerName -h HostIPAddresses</code> | 該当なし | ホスト要求、 メッセージ タ イプ 5、ビット 11 で 1 に設定 | ホスト データ (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 (2-35 ページ) を参照して ください。) |

表 6-3 出力プラグインで作成されるデフォルト要求(続き)

| 構文 | プラグインの呼び出し | 送信内容 | 要求するデータ |
|---|-------------------------|--|---|
| <code>./ssl_test.pl eStreamerServerName -d "Global \ domain \ subdomain"</code> | 該当なし | 指定されたドメインまたはサブドメインに対するイベントストリーム要求。 | 指定されたドメインに対するイベント情報のストリーム(ドメインストリーミング要求メッセージの形式(2-39 ページ)を参照してください。) |
| <code>./ssl_test.pl eStreamerServerName -o print -f TextFile</code> | OutputPlugins/print.pm | イベントストリーム要求、メッセージタイプ 2、ビット 2 および 20 ~ 24 を 1 に設定 | イベントデータ(イベントストリーム要求メッセージの形式(2-11 ページ)、関連ポリシーレコード(3-25 ページ)、関連ルールレコード(3-26 ページ)、ディスカバリイベントのメタデータ(4-8 ページ)、イベントタイプ別ホストディスカバリ構造(4-46 ページ)、およびイベントタイプ別のユーザーデータ構造(4-64 ページ)を参照してください。 eStreamer は、ビット 2 がイベントストリーム要求に設定されているため、タイプ 1 の侵入イベントを送信します。 |
| <code>./ssl_test.pl eStreamerServerName -o pcap -f TargetPCAPFile</code> | OutputPlugins/pcap.pm | イベントストリーム要求、メッセージタイプ 2、ビット 0 および 23 を 1 に設定 | パケットデータ(イベントデータメッセージの形式(2-19 ページ)およびパケットレコード 4.8.0.2 以上(3-7 ページ)を参照してください。 eStreamer は、ビット 0 がイベントストリーム要求に設定されているため、パケットデータのみを送信します。 |
| <code>./ssl_test.pl eStreamerServerName -o csv -f CSVFile</code> | OutputPlugins/csv.pm | イベントストリーム要求、メッセージタイプ 2、ビット 2 および 23 を 1 に設定 | 侵入イベントデータ(イベントデータメッセージの形式(2-19 ページ)および侵入イベントレコード 6.0 以上(3-9 ページ)を参照してください。 eStreamer は、ビット 2 がイベントストリーム要求に設定されているため、タイプ 1 の侵入イベントを送信します。 |
| <code>./ssl_test.pl eStreamerServerName -o snmp -f SNMPServer</code> | OutputPlugins/snmp.pm | イベントストリーム要求、メッセージタイプ 2、ビット 2、20、および 23 を 1 に設定 | 侵入イベントデータ(イベントデータメッセージの形式(2-19 ページ)および侵入イベントレコード 6.0 以上(3-9 ページ)を参照してください。 eStreamer は、ビット 2 がイベントストリーム要求に設定されているため、タイプ 1 の侵入イベントを送信します。 |
| <code>./ssl_test.pl eStreamerServerName -o syslog</code> | OutputPlugins/syslog.pm | イベントストリーム要求、メッセージタイプ 2、ビット 2、20、および 23 を 1 に設定 | 侵入イベントデータ(イベントデータメッセージの形式(2-19 ページ)および侵入イベントレコード 6.0 以上(3-9 ページ)を参照してください。 eStreamer は、ビット 2 がイベントストリーム要求に設定されているため、タイプ 1 の侵入イベントを送信します。 |

テスト スクリプトで要求されるデータ タイプの変更

SFStreamer.pm Perl モジュールは、データを要求する際に、サンプル スクリプトで使用できる複数の要求フラグの変数を定義します。次の表では、イベントストリーム要求メッセージで、各要求フラグを設定するために呼び出す要求フラグの変数を示しています。出力モジュールのいずれかを使用してさまざまなデータを要求する場合は、モジュールの \$FLAG の設定を編集できます。

要求フラグ、お客様が要求するデータ、各フラグに対応する製品バージョンの詳細については、[要求フラグ\(2-13 ページ\)](#)を参照してください。

表 6-4 サンプル スクリプトで使用される要求フラグ変数

| 変数 | 設定する要求フラグ | 要求するデータ |
|--------------------------------|-----------|---|
| \$FLAG_PKTS | 0 | パケット データ |
| \$FLAG_METADATA | 1 | バージョン 1 のメタデータ |
| \$FLAG_IDS | 2 | タイプ 1 の侵入イベント |
| \$FLAG_RNA | 3 | バージョン 1 の検出イベント |
| \$FLAG_POLICY_EVENTS | 4 | バージョン 1 の関連イベント |
| \$FLAG_IMPACT_ALERTS | 5 | 侵入の影響アラート |
| \$FLAG_IDS_IMPACT_FLAG | 6 | タイプ 7 の侵入イベント |
| \$FLAG_RNA_EVENTS_2 | 7 | バージョン 2 の検出イベント |
| \$FLAG_RNA_FLOW | 8 | バージョン 1 の接続データ |
| \$FLAG_POLICY_EVENTS_2 | 9 | バージョン 2 の関連イベント |
| \$FLAG_RNA_EVENTS_3 | 10 | バージョン 3 の検出イベント |
| \$FLAG_HOST_ONLY | 11 | \$FLAG_HOST_SINGLE(1 台のホスト用)または \$FLAG_HOST_MULTI(複数のホスト用)とともに送信される場合は、イベント データのないホスト データのみ |
| \$FLAG_RNA_FLOW_3 | 12 | バージョン 3 の接続データ |
| \$FLAG_POLICY_EVENTS_3 | 13 | バージョン 3 の関連イベント |
| \$FLAG_METADATA_2 | 14 | バージョン 2 のメタデータ |
| \$FLAG_METADATA_3 | 15 | バージョン 3 のメタデータ |
| \$FLAG_RNA_EVENTS_4 | 17 | バージョン 4 の検出イベント |
| \$FLAG_RNA_FLOW_4 | 18 | バージョン 4 の接続データ |
| \$FLAG_POLICY_EVENTS_4 | 19 | バージョン 4 の関連イベント |
| \$FLAG_METADATA_4 | 20 | バージョン 4 のメタデータ |
| \$FLAG_RUA | 21 | ユーザ アクティビティ イベント |
| \$FLAG_POLICY_EVENTS_5 | 22 | バージョン 5 の関連イベント |
| \$FLAGS_SEND_ARCHIVE_TIMESTAMP | 23 | タイムスタンプを含む拡張されたイベント ヘッダーは、eStreamer サーバでの処理のためにイベントがアーカイブされたときに適用されます |
| \$FLAG_RNA_EVENTS_5 | 24 | バージョン 5 の検出イベント |
| \$FLAG_RNA_EVENTS_6 | 25 | バージョン 6 の検出イベント |

表 6-4 サンプル スクリプトで使用される要求フラグ変数(続き)

| 変数 | 設定する要求フラグ | 要求するデータ |
|------------------------|-----------|-----------------------|
| \$FLAG_RNA_FLOW_5 | 26 | バージョン 5 の接続データ |
| \$FLAG_EXTRA_DATA | 27 | 侵入イベント追加データレコード |
| \$FLAG_RNA_EVENTS_7 | 36 | バージョン 7 の検出イベント |
| \$FLAG_POLICY_EVENTS_6 | 29 | バージョン 6 の関連イベント |
| \$FLAG_DETAIL_REQUEST | 30 | eStreamer に対する拡張された要求 |



注意

バージョン 5.x より前は、すべてのイベントタイプでは、参照クライアントは detection engine ID フィールドを sensor ID としてラベル付けしています。

Perl 参照クライアントのための証明書の作成

ライセンス:任意(Any)

Perl 参照クライアントを使用する前に、Management Center または管理対象デバイスで、クライアントを実行するコンピュータ用に証明書を作成する必要があります。次に、証明書ファイルをクライアント コンピュータにダウンロードし、それを使用して証明書(server.crt)および RSA キーファイル(server.key)を作成します。

Perl 参照クライアントのための証明書の作成:

アクセス:管理

- ステップ 1** [システム(System)] > [統合(Integration)] > [eStreamer(eStreamer)] を選択します。
[] ページが表示されます。
- ステップ 2** [クライアントの作成(Create Client)] をクリックします。
[クライアントの作成(Create Client)] ページが表示されます。
- ステップ 3** [ホスト名(Hostname)] フィールドに、eStreamer クライアントを実行しているホストのホスト名または IP アドレスを入力します。
- コメント** ホスト名を使用する場合は、ホスト入力サーバはホストを IP アドレスに解決できる必要があります。DNS 解決を設定していない場合、最初に設定するか、IP アドレスを使用する必要があります。
- ステップ 4** 証明書ファイルを暗号化するには、[パスワード>Password] フィールドにパスワードを入力します。
- ステップ 5** [保存(Save)] をクリックします。

eStreamer サーバはクライアント コンピュータから Management Center 上のポート 8302 へのアクセスを許可し、クライアント/サーバ認証時に使用する認証証明書を作成します。新しいクライアントが [クライアント(eStreamer Client)] の下に表示された状態で、[クライアント(eStreamer Client)] ページが再表示されます。Management Center

- ステップ 6 証明書ファイルの横にあるダウンロードアイコン(📄)をクリックします。
- ステップ 7 SSL 認証のためにクライアント コンピュータが使用するディレクトリに証明書ファイルを保存します。

これで、クライアントは Management Center に接続できるようになりました。



ヒント

クライアントのアクセスを取り消すには、削除するホストの横にある削除アイコン(🗑️)をクリックします。Management Center でホスト入力サービスを再開する必要はありません。アクセスはただちに取り消されます。

eStreamer Perl 参照クライアントの実行

eStreamer Perl 参照クライアント スクリプトは、Linux カーネルを備えた 64 ビットのオペレーティングシステムで使用するよう設計されていますが、クライアント マシンが [eStreamer Perl 参照クライアントの設定 \(6-6 ページ\)](#) で定義されている前提条件を満たしていれば、任意の POSIX ベースの 64 ビットのオペレーティングシステムでも機能します。

詳細については、次の項を参照してください。

- [ホストの要求を使用した SSL 上のクライアント接続のテスト \(6-12 ページ\)](#)
- [参照クライアントを使用した PCAP のキャプチャ \(6-13 ページ\)](#)
- [参照クライアントを使用した CSV レコードのキャプチャ \(6-13 ページ\)](#)
- [参照のクライアントを使用した SNMP サーバへのレコードの送信 \(6-13 ページ\)](#)
- [参照クライアントを使用した Syslog へのイベントのロギング \(6-13 ページ\)](#)
- [IPv6 アドレスへの接続 \(6-14 ページ\)](#)

ホストの要求を使用した SSL 上のクライアント接続のテスト

ssl_test.pl スクリプトを使用すると、eStreamer サーバおよび eStreamer クライアント間で接続をテストできます。ssl_test.pl スクリプトはどのレコードタイプも処理し、STDOUT または指定する出力プラグインにこれを出力します。出力オプションを使用せずに -h オプションを使用すると、指定したホストのホストデータが端末にストリームされます。



コメント

STDOUT へ raw パケット データを出力すると端末を干渉するため、出力プラグインへの方向付けをせずに、このスクリプトを使用してパケットデータをストリームすることはできません。

次の構文と、ssl_test.pl スクリプトを使用して、標準的な出力にホスト データを送信します。

```
./ssl_test.pl eStreamerServerIPAddress -h HostIPAddresses
```

たとえば、10.10.0.4 の IP アドレスの eStreamer サーバへの接続を介した 10.0.0.0/8 サブネット上のホストのホストデータの受信をテストするには、次の構文を使用します。

```
./ssl_test.pl 10.10.0.4 -h 10.0.0.0/8
```

参照クライアントを使用した PCAP のキャプチャ

ストリームされたパケットデータを PCAP ファイルでキャプチャし、クライアントが受信するデータの構造を確認する場合に、参照クライアントを使用できます。`-o pcap` 出力オプションを使用する際は、`-f` を使用してターゲット ファイルを指定する必要があることに注意してください。

`ssl_test.pl` スクリプトを使用して、ストリームされたパケットデータを PCAP ファイルでキャプチャするには、次の構文を使用します。

```
./ssl_test.pl eStreamerServerIPAddress -o pcap -f ResultingPCAPFile
```

たとえば、10.10.0.4 の IP アドレスの eStreamer サーバからストリームされたイベントを使用して、`test.pcap` という名前の PCAP ファイルを作成するには、次の構文を使用します。

```
./ssl_test.pl 10.10.0.4 -o pcap -f test.pcap
```

参照クライアントを使用した CSV レコードのキャプチャ

ストリームされた侵入イベントデータを CSV ファイルでキャプチャし、クライアントが受信するデータの構造を確認する場合も、参照クライアントを使用できます。

次の構文を使用して `streamer_csv.pl` スクリプトを実行します。

```
./ssl_test.pl eStreamerServerIPAddress -o csv -f ResultingCSVFile
```

たとえば、10.10.0.4 の IP アドレスの eStreamer サーバからストリームされたイベントを使用して、`test.csv` という名前の CSV ファイルを作成するには、次の構文を使用します。

```
./ssl_test.pl 10.10.0.4 -o csv -f test.csv
```

参照のクライアントを使用した SNMP サーバへのレコードの送信

侵入イベントデータを SNMP サーバにストリームする場合も、参照クライアントを使用できます。`-f` オプションを使用して、イベントを受信する SNMP トラップサーバの名前を示します。この出力方法では、パスに `snmptrapd` という名前のバイナリが必須であるため、UNIX のようなシステムでのみ機能することに注意してください。

SNMP サーバに侵入イベントを送信するには、次の構文を使用します。

```
./ssl_test.pl eStreamerServerIPAddress -o snmp  
-f SNMPServerName
```

たとえば、10.10.0.4 の IP アドレスの eStreamer サーバからストリームされたイベントを使用して、10.10.0.3 で SNMP サーバにイベントを送信するには、次の構文を使用します。

```
./ssl_test.pl 10.10.0.4 -o snmp -f 10.10.0.3
```

参照クライアントを使用した Syslog へのイベントのロギング

クライアントのローカル syslog サーバに侵入イベントをストリームする場合も、参照クライアントを使用できます。

Syslog にイベントを送信するには、次の構文を使用します。

```
./ssl_test.pl eStreamerServerIPAddress -o syslog
```

たとえば、10.10.0.4 の IP アドレスの eStreamer サーバからストリームされたイベントを記録するには、次の構文を使用します。

```
./ssl_test.pl 10.10.0.4 -o syslog
```

IPv6 アドレスへの接続

プライマリ管理インターフェイスを介して IPv6 アドレスの Management Center に接続する場合も、参照クライアントを使用できます。クライアントのマシンには **Socket6** および **IO::Socket::INET6 Perl** モジュールがインストールしてある必要があり、`-ipv6` オプションまたは短縮形式の `-i` を使用します。

`ssl_test.pl` スクリプトを使用して IPv6 アドレスを指定するには、次の構文を使用します。

```
./ssl_test.pl -ipv6 eStreamerServerIPAddress
```

または

```
./ssl_test.pl -i eStreamerServerIPAddress
```

たとえば、IPv6 アドレス `2001:470:e09c:20:7c1e:5248:1bf7:2ea0` を使用して Management Center に接続するには、次の構文を使用します。

```
./ssl_test.pl -ipv6 2001:470:e09c:20:7c1e:5248:1bf7:2ea0
```