ストリーミングテレメトリ

統合ダッシュボードと CLI を使用したデバイスのモニタリングに加えて、テレメトリ コレクタにデータを送信するようにデバイスを構成できます。コレクタを使用して、ネットワーク内の複数のデバイスをモニターできます。

次のトピックでは、ストリーミング テレメトリを使用するための要件と、テレメトリ コレクタのセットアップと Threat Defense デバイスへの接続方法について説明します。

ストリーミング テレメトリについて

Google リモートプロシージャコール（gRPC）を使用してデータを収集する外部テレメトリコレクタに、システムの正常性とテレメトリデータを送信するようにデバイスを設定できます。その後、テレメトリコレクタを使用してデバイスをモニターし、カスタムテレメトリ ソリューションと統合できます。

デバイスとテレメトリ コレクタの間の接続では、相互トランスポート層セキュリティ（mTLS）認証を使用してセキュリティを確保します。デバイスとテレメトリ コレクタは、クライアントとサーバーのアイデンティティを確認するために証明書を交換し、データ転送を暗号化します。デバイスは、テレメトリ サーバーへの接続を開始します（ダイヤルアウト モデル）。

ストリーミング テレメトリに関するガイドライン

ストリーミング テレメトリを構成するときは、次のガイドラインに留意してください。

- IPv4 アドレスのみを使用できます。
- Threat Defense デバイスとテレメトリ コレクタに使用される証明書は、通信を確保するために同じ認証局（CA）によって署名されている必要があります。Threat Defense デバイス

に必要な証明書を作成し（または Threat Defense Web サーバー用に設定された証明書を再利用することにし）、テレメトリ コレクタで使用する証明書をダウンロードしてください。

- 各 Threat Defense デバイスは、単一のテレメトリ コレクタにのみ接続できます。ただし、単一のコレクタを複数の Threat Defense デバイスで使用することはできます。
- 高可用性で構成されているデバイスの場合、各デバイスでストリーミングテレメトリを個別に構成する必要があります。テレメトリ構成が、アクティブ ユニットからスタンバイ ユニットに複製されることはありません。必要であれば、プライマリ ユニットとセカンダリ ユニットを構成して、異なるテレメトリコレクタを使用することができます。
- Threat Defense デバイスは、ストリーミング テレメトリに次のポートを使用します。
  - 制御チャネル：9276 over HTTP
  - データ チャネル：8087 over HTTP
  - ポート 9273 および 9276 は診断に使用されます。
- 次の Threat Defense API を使用してリモートのプロシージャを記述し、テレメトリ コレクタから Threat Defense デバイスを構成することができます。
  - /object/internalcertificates
  - /object/externalcacertificates
  - /object/networks
  - /devicesettings/default/telemetrystreamingconfig

ストリーミング テレメトリを有効にする

テレメトリ コレクタをセットアップしたら、 Threat Defense デバイスとテレメトリ サーバー間の接続を構成できます。接続が構成されると、Threat Defense デバイスは自動的に接続のセットアップを試み、成功すると、コレクタが要求するのと同じ頻度でデータをストリーミングします。

始める前に

テレメトリ コレクタが テレメトリ コレクタのセットアップ （7 ページ）に記載されている要件を満たしていることを確認します。

手順

ステップ 1 **[オブジェクト（Objects）]** > **[証明書（Certificates）]**を選択し、**[+]** > **[内部証明書の追加（Add Internal Certificate）]**を選択し、Threat Defense デバイスでセキュアな通信のために使用され

るクライアント証明書をアップロードします。詳細については、内部および内部 CA 証明書のアップロードを参照してください。

ステップ 2 **[オブジェクト（Objects）]**>**[証明書（Certificates）]**を選択し、**[+]**>**[信頼される CA 証明書の追加（Add Trusted CA Certificate）]**を選択し、Threat Defense デバイスでコレクタのアイデンティティを確認するために使用される CA 証明書をアップロードします。詳細については、信頼できる CA 証明書のアップロードを参照してください。

ステップ 3 **[オブジェクト（Objects）]**>**[ネットワーク（Network）]**を選択し、**[+]** を選択して、テレメトリ コレクタを識別するネットワーク オブジェクトを作成します。詳細については、ネットワーク オブジェクトとグループの設定を参照してください。

コレクタの IPv4 アドレスを使用してホスト オブジェクトを作成するか、telemetry.domain.com などのコレクタの完全修飾名を含む FQDN オブジェクトを作成します。FQDN は IPv4 アドレスに解決する必要があり、名前が正しく変換されるように DNS も構成する必要があります。

ステップ 4 ネットワーク オブジェクトの ID を取得します。

a) 詳細オプション ボタン ⋮ から **[API エクスプローラ（API Explorer）]** を選択して、API ページにアクセスします。

b) **[NetworkObject]**で、**[GET /object/networks]** を選択します。

c) **[パラメータ（Parameters）]**セクションの**[フィルタ（Filter）]**フィールドで、オブジェクト名と等しくなるように出力のフィルタを設定します。たとえば、作成したネットワーク オブジェクトが TelemetryCollector の場合、フィルタは次のようになります。

名前：TelemetryCollector

d) [GET /object/networks] セクションの下部までスクロールし、**[試行する（Try It Out）]** をクリックします。

e) 呼び出しが正しい場合は、次のような 200 応答コードと意味のあるオブジェクト本体を取得できるはずです。id エントリを探し、値をメモします。この例では、id の値は **79ee2ea9-76b7-11ef-9515-f5b34b7d9531**です。


```
{
  "items": [
    {
      "version": "p4qjmqtqn5c5e",
      "name": "TelemetryCollector",
      "description": null,
      "subType": "HOST",
      "value": "10.1.1.1",
      "isSystemDefined": false,
      "dnsResolution": "IPV4_AND_IPV6",
      "id": "79ee2ea9-76b7-11ef-9515-f5b34b7d9531",
      "type": "networkobject",
      "links": {
        "self":
"https://ftd1.domain.com/api/fdm/v6/object/networks/79ee2ea9-76b7-11ef-9515-f5b34b7d9531"

      }
    }
```


ステップ 5 内部証明書オブジェクトの ID を取得します。

a) API エクスプローラの **[証明書（Certificate）]** で、**[GET /object/internalcertificates]** を選択します。
b) **[フィルタ（Filter）**] フィールドで、証明書名でフィルタ処理します。たとえば、Threat Defense デバイスの内部証明書が FTD1Cert の場合、フィルタは次のようになります。

   名前：FTD1Cert

c) [GET /object/internalcertificates] セクションの下部までスクロールし、**[試行する（Try It Out）]** をクリックします。
d) 呼び出しが正しい場合は、次のような 200 応答コードと意味のあるオブジェクト本体を取得できるはずです。id エントリを探し、値をメモします。この例では、id の値は **d874dfa3-7423-11ef-b3a0-09429aedc3d3**です。

```
{
  "items": [
    {
      "version": "gr573izgdsj2o",
      "name": "FTD1Cert",
…
      ATTRIBUTES REMOVED
…
      "id": "d874dfa3-7423-11ef-b3a0-09429aedc3d3",
      "type": "internalcertificate",
      "links": {
        "self":
"https://ftd1.domain.com/api/fdm/v6/object/internalcertificates/d874dfa3-7423-11ef-b3a0-09429aedc3d3"

      }
    }
```

ステップ 6 信頼できる CA 証明書オブジェクトの ID を取得します。

a) API エクスプローラの **[証明書（Certificate）]** で、 **[GET /object/externalcacertificates]** を選択します。
b) **[フィルタ（Filter）**] フィールドで、証明書名でフィルタ処理します。たとえば、テレメトリ コレクタの信頼できる CA 証明書が TelemetryCollectorCert である場合、フィルタは次のようになります。

   名前：TelemetryCollectorCert

c) [GET /object/externalcacertificates] セクションの下部までスクロールし、**[試行する（Try It Out）]** をクリックします。
d) 呼び出しが正しい場合は、次のような 200 応答コードと意味のあるオブジェクト本体を取得できるはずです。id エントリを探し、値をメモします。この例では、id の値は **c3d925b4-7423-11ef-b3a0-bf815c0136ac**です。

```
{
  "items": [
    {
      "version": "fkry47nobvcnu",
      "name": "TelemetryCollectorCert",
…
      ATTRIBUTES REMOVED
…
      "id": "c3d925b4-7423-11ef-b3a0-bf815c0136ac",
```

```
      "type": "externalcacertificate",
      "links": {
        "self":
"https://ftd1.domain.com/api/fdm/v6/object/externalcacertificates/c3d925b4-7423-11ef-b3a0-bf815c0136ac"

      }
    }
```

ステップ 7 Threat Defense デバイスとテレメトリ コレクタ間の接続を構成します。

a) API エクスプローラの **[TelemetryStreamingConfig]**で、**[POST /devicesettings/default/telemetrystreamingconfig] を選択します。**

b) **[パラメータ（Parameters）] > [本文（Body）]**で、**[値（Value）]** 編集ボックスに次のテンプレートを入力します（非表示の無効な文字がコピーされないようにしてください）。各フィールドの意味は、このテンプレートで説明されています。<> 文字内の説明は、置換が必要な変数です。他の値は、表示されているとおりにする必要があります。カンマ、カッコ、コロン、および {} の配置は重要です。

```
{
"name": "<a unique name for the gRPC streaming config API>",
"connectionMode": "DIAL_OUT",
"port": "<port on which the collector is waiting for connections from the
          Threat Defense device, 1-65535. Check the collector configuration
          for the right value.>",
"targetHost": {
     "name": "<name of the network object that identifies the telemetry collector
host>",
     "id": "<ID of the network object>",
     "type": "networkobject"
},
"clientCertificate": {
      "name": "<The name of the internal certificate that identifies the Threat
Defense device>",
      "id": "<ID of the internal certificate object.>",
      "type": internalcertificate
}
"caCertificate": {
     "name": "<The name of the trusted CA certificate for the telemetry collector>",

       "id": "<ID of the trusted CA certificate>",
      "type": "externalcacertificate"
},
"type": "telemetrystreamingconfig"
}
```

例：

この手順に示されている例の値を指定すると、以下が正しいペイロードとなります。名前とポートの値は前の手順で決定されたものではないことに注意してください。必要に応じて変更できます。

```
{
"name": "YourCompanyTelemetry",
"connectionMode": "DIAL_OUT",
"port": 50051,
"targetHost": {
   "name": "TelemetryCollector",
   "id": "79ee2ea9-76b7-11ef-9515-f5b34b7d9531",
   "type": "networkobject"
```

```
},
"clientCertificate": {
    "name": "FTD1Cert",
    "id": "d874dfa3-7423-11ef-b3a0-09429aedc3d3",
    "type": "internalcertificate"
},
"caCertificate": {
     "name": "TelemetryCollectorCert",
     "id": "c3d925b4-7423-11ef-b3a0-bf815c0136ac",
     "type": "externalcacertificate"
},
"type": "telemetrystreamingconfig"
}
```


c) セクションの下部までスクロールし、**[試行する（Try It Out）]** をクリックします。

d) 200 の応答コードを探します。他にコードが表示された場合は、エラーを修正して再試行してください。成功すると、応答の本体は次のようなものになります。


```
{
  "version": "jfwu476cue32n",
  "name": "YourCompanyTelemetry",
  "connectionMode": "DIAL_OUT",
  "port": 50051,
  "targetHost": {
    "version": "p4qjmqtqn5c5e",
    "name": "TelemetryCollector",
    "id": "79ee2ea9-76b7-11ef-9515-f5b34b7d9531",
    "type": "networkobject"
  },
  "clientCertificate": {
    "version": "gr573izgdsj2o",
    "name": "FTD1Cert",
    "id": "d874dfa3-7423-11ef-b3a0-09429aedc3d3",
    "type": "internalcertificate"
  },
  "caCertificate": {
    "version": "fkry47nobvcnu",
    "name": "TelemetryCollectorCert",
    "id": "c3d925b4-7423-11ef-b3a0-bf815c0136ac",
    "type": "externalcacertificate"
  },
  "id": "b6dc6f28-76c1-11ef-9515-8ff976794f92",
  "type": "telemetrystreamingconfig",
  "links": {
    "self":
"https://ftd1.domain.com/api/fdm/v6/devicesettings/default/telemetrystreamingconfig/b6dc6f28-76c1-11ef-9515-8ff976794f92"

  }
}
```


次のタスク

テレメトリ ストリーミングが正しく機能していることを確認する方法については、次のトピックを参照してください。

- テレメトリ ストリーミング サービスのステータスの確認 （12 ページ）
- テレメトリ コレクタがデータを受信することを確認する （14 ページ）

テレメトリ コレクタのセットアップ

Threat Defense デバイスからテレメトリ データを受信し、情報を集約して、組織の運用要件を満たせるようわかりやすい方法で表示するには、独自のテレメトリ コレクタを用意する必要があります（市販またはカスタムのいずれか）。次に、gRPC コールを使用して Threat Defense デバイスで実行されている Telegraf コンポーネントからデータを収集するようにテレメトリ コレクタをセットアップする方法についての一般的な情報を示します。

手順

ステップ 1 テレメトリ コレクタがテレメトリ コレクタに関するガイドライン （7 ページ）に記載された要件を満たしていることを確認してください。

ステップ 2 テレメトリ コレクタでのプロトコル定義 （8 ページ）の説明に従って、proto 定義を構成します。

ステップ 3 Threat Defense デバイスとテレメトリ コレクタ間の通信 （9 ページ）で説明されているように、テレメトリ コレクタが Threat Defense デバイスで実行されている Telegraf クライアントを受信して応答できることを確認します。

テレメトリ コレクタに関するガイドライン

- Windows、Mac、Linux、または UNIX サーバーでクライアントを実行できます。
- テレメトリ コレクタには Go をインストールする必要があります。Go の最小バージョンは 1.20 です。
- テレメトリ コレクタには IPv4 アドレスが必要であり、それを使用する Threat Defense デバイスとの直接またはプロキシ経由の適切なルーティングが必要です。接続が失われた場合、 Threat Defense デバイスは 5 分ごとに接続を再試行します。
- テレメトリ コレクタのリスニング ポートは、他の目的に予約されていない有効な TCP ポート（1 ～ 65535）である必要があります。
- テレメトリ コレクタのサーバー証明書、サーバー キー、および CA 証明書は、次のパスにある必要があります。
  - サーバー キー：/root/grpc-certs/keys/server.key
  - サーバー証明書：/root/grpc-certs/keys/server.crt
  - CA 証明書：/root/grpc-certs/keys/ca.crt

- 証明書が期限切れになると、テレメトリ クライアントに認証エラーが表示され、ストリーミングが停止します。認証の問題を修正して送信を再開するには、証明書を置き換える必要があります。
- メッセージは、Prometheus時系列フォーマットを使用します。クライアントはこのフォーマットを処理できる必要があります。
- 次のテレメトリ コレクタはサポートされていません：https://github.com/CiscoSE/grpc_collector。

テレメトリ コレクタでのプロトコル定義

Threat Defense デバイスは、データの構造化にプロトコル バッファを使用します。テレメトリ コレクタの proto 定義には、次のものが含まれている必要があります。

```
syntax = "proto3";
// Update the go_package option to a local package path
option go_package = "grpcstreaming/grpc_streaming_proto";
package proto;
service GrpcStreamingService {
    rpc DataStream (stream DataResponse) returns (stream DataRequest);
    rpc ControlStream (stream ControlResponse) returns (stream ControlRequest);
}
message ControlResponse {
    string version = 1;
    string ftd_uuid = 2;
    string hostname = 3;
    bool init_streaming = 4;
    repeated string capabilities = 5; // ['metric_streaming']
    // cancel stream acknowledgement
    bool ack = 6;
}
message ControlRequest {
    string version = 1;
    int64 interval = 2;
    repeated string metric_subscriptions = 3;
    // cancel stream
    StreamCancellationMessage cancellation_message = 4;
}
message DataResponse {
    string ftd_uuid = 1;
    repeated Metric metrics = 2;
}
message DataRequest {
    bool ack = 1;
}
message StreamCancellationMessage {
    string collector_uuid = 1;
    bool cancel_stream = 2;
}
message Tag {
    string key = 1;
    string value = 2;
}
message Metric {
    int64 timestamp = 1;
    string metricFamily = 2;
    double value = 3;
```

```
    repeated Tag tags = 4;
    string metricType = 5; //Counter | Gauge
}
```

Threat Defense デバイスとテレメトリ コレクタ間の通信

最初に接続を確立するために、Threat Defense デバイスは、initial_request=True を示すペイロードを含む gRPC 要求を、テレメトリ コレクタに送信します。要求は、その機能（「metric_streaming」）もアドバタイズします。

テレメトリ コレクタは、接続を確認し、コレクタがストリーミング データを受信できる時間間隔を含むメッセージで応答する必要があります。この間隔は、Threat Defense デバイスがテレメトリをコレクタにストリーミングすることになる、予想される頻度を示します。範囲は 1 分（60 秒）から 24 時間です。有効な頻度が取得されると、システムはメトリックの初期セットを送信し、要求されたレートで追加情報を提供します。

通信に使用される RPC メソッドは次のとおりです。

- ストリーミングを設定するために Threat Defense デバイスの Telegraf コンポーネントから送信される単項 RPC（制御メッセージ）。

  rpc ConfigureMetricStreaming（TelegrafControlMessage）は（CollectorControlMessage）を返します。

- ストリーミング RPC（データ メッセージ）。 Threat Defense デバイスの Telegraf とテレメトリ コレクタ間のメトリック データの双方向ストリーミングを促進にします。

  rpc BiDirectionalMetricStreaming（stream TelegrafDataMessage）は（stream CollectorDataMessage）を返します。

次のトピックでは、テレメトリ コレクタがデバイスから受信するメッセージと、コレクタがデバイスに送信する必要があるメッセージについて詳しく説明します。

テレグラフ制御メッセージ（制御チャネル）

テレグラフ制御メッセージは、メトリックストリーミングを開始するため、Threat Defense デバイス上の Telegraf コンポーネントからコレクタに送信されます。その中では、init_request フラグが true に設定されています。

```
message TelegrafControlMessage {
// Indicates the proto version used by the FTD device. First version will be 1.0
    string version;

// Indicates the device id of the sender
    string device_uuid;

// Indicates the device hostname of the sender
    string hostname;

//list of strings indicating the capabilities of FTD. This will be “metric_streaming”
    repeated string capabilities;

// Flag to initiate a collector response for configuring telemetry streaming
```

```
    bool init_streaming = 1;
}
```

コレクタ制御メッセージ（制御チャネル）

コレクタ制御メッセージは、Telegraf制御メッセージへの応答としてコレクタからThreat Defenseデバイスに送信されます。これには、メトリックバッチの望ましい頻度としての間隔が含まれています。範囲は1分（60秒）から24時間です。メトリック サブスクリプション コンポーネントはオプションです。

```
message CollectorControlMessage {
// Indicates the proto version used by the target. Current version supported is 1.0
    string version;
// Time interval at which the FTD device should send metric batches
    int64 interval = 1;
// Set of metric families to subscribe to, the default value is the only supported value.
// Default: "all"
    repeated string metricSubscriptions = 2;
}
```

ストリーム キャンセル メッセージ（制御チャネル）

ストリーム キャンセル メッセージは、Threat Defense デバイスとテレメトリ コレクタ間の既存のテレメトリ ストリームをキャンセルするために使用されます。Threat Defense デバイスまたはコレクタのいずれかが、制御チャネルでこのメッセージを発行できます。キャンセル要求の受信者は、ACKメッセージで応答する必要があります。キャンセルが完了すると、 Threat Defense デバイスは、コレクタが新しいストリーミング要求を受け入れるまで、5分ごとにコレクタを再試行します。ストリーミングを完全に停止するには、Threat Defense デバイスのストリーミング構成を削除するだけです。

```
message StreamCancellationMessage {
// Indicates the proto version used by the FTD device. First version will be 1.0
    string version;

// Indicates the device id of the sender
    string device_uuid;
// This flag indicates that the cancel request is true
    bool cancel_request;
}
```

Telegraf データ メッセージ（データ チャネル）

Telegrafデータメッセージには 、Threat Defense デバイスからコレクタに送信されるメトリックのバッチが含まれています。これには、個々のメトリック メッセージを含む、メトリックと呼ばれる繰り返しフィールドが含まれています。

```
message TelegrafDataMessage {
    // Batch of metrics sent by Telegraf
    repeated Metric metrics = 1;
}
```

メトリック メッセージ（データ チャネル）

テレメトリ データは、インターフェイス、CPU、メモリ、ディスク使用率など、システムのさまざまなコンポーネントから収集されたメトリックであり、モニタリングと分析のためにThreat Defense デバイスからテレメトリ コレクタに送信されます。このデータのフォーマットは、プロトコル定義のメトリック メッセージによって定義されます。

次に、テレメトリ データを含むメトリック メッセージの例を示します。

```
METRIC=timestamp:1718257445000
metricFamily:"cpu"  value:0.7  tags:{key:"cpu"  value:"CPU))"}  tags:{key:"description"
  value:"cpu_utilisation"}  tags:{key:"process"  value:"lina"}  tags:{key:"rcpu"
value:"x86_cpu0"} tags:{key:"uuid"  value:"7eb19498-2519-11ef-a8dd-b74b4d43a7e7"}
metricType:"Gauge"
```

メトリック メッセージには、次のフィールドが含まれます。

- タイムスタンプ（int64 型のタイムスタンプ）：メトリックが記録された正確な時刻。エポック時間で表されます。
- メトリック ファミリ（文字列metricFamily）：「cpu」、「メモリ」、「ディスク」、「インターフェイス」など、測定対象のシステム コンポーネントまたはリソース。
- 値（double 型の値）：メトリックの数値。この値の解釈方法は、メトリックのタイプによって異なります。たとえば、CPUの使用率の場合には、パーセント値として解釈されます。
- タグ（繰り返しタグのタグ））：メトリックに関する追加のコンテキスト。各タグはキーと値のペアであり、キーは説明のためのラベル（「cpu」、「プロセス」、「インターフェイス」など）で、値は特定の詳細（「CPU0」、「lina」、「GigabitEthernet0/0」など）です）。
- メトリック タイプ（文字列metricType）：メトリックの特性。時間の経過とともに累積される「カウンタ」（送信された合計パケット数など）や、特定の時点の値を表す「ゲージ」（CPU 使用率など）を指定できます。

Telegraf データ メッセージ（データ チャネル）

Telegraf データメッセージには 、Threat Defense デバイスからコレクタに送信されるメトリックのバッチが含まれています。これには、個々のメトリック メッセージを含む、メトリックと呼ばれる繰り返しフィールドが含まれています。

```
message TelegrafDataMessage {
    // Batch of metrics sent by Telegraf
    repeated Metric metrics = 1;
}
```

テレメトリ ストリーミングのトラブルシューティング

次のトピックでは、テレメトリ ストリーミングのトラブルシューティング方法について説明します。

テレメトリ ストリーミング サービスのステータスの確認

テレメトリ ストリーミングを有効にすると、構成は直ちにデバイスにプッシュされます。すべての値が正しく、コレクタへのパスがある場合、サービスが再起動し、テレメトリサーバーに接続します。

手順

ステップ 1 API エクスプローラの **[TelemetryStreamingConfig]**で、 **[GET /operational/telemetrystreamingstatuses]**を選択します。

ステップ 2 **[試してみる（Try It Out）]**をクリックします。

ステップ 3 応答コードが 200 の場合は、応答本文の **state** 値を確認します。

理想的な応答は、エラーメッセージが出ずに CONNECTED 状態になることです。

状態が DISCONNECTED である場合は、エラーメッセージを調べて、考えられる問題を特定します。

たとえば、次のエラーは接続を確立できなかったことを示しています。このエラー例は、テレメトリ コネクタ用ではない IP アドレスを指定することによって生成されたものであるため、i/o タイムアウトが発生しています。これは、指定したポート値が正しくないことを示している可能性もあります。エラーはタイプに分類され、サービスの開始後にエラーが発生した回数のカウントが含まれていることに注意してください。

```
"items": [
    {
      "state": "DISCONNECTED",
      "errors": [
        {
          "errorType": "StreamingErrors",
          "errorMessage": "2024-09-30 19:20:33.575156378 +0000 UTC m=+308552.647839001
: rpc error: code = Unavailable desc = connection error: desc = \"transport: Error while
 dialing: dial tcp 10.1.1.1:50051: i/o timeout\"",
          "errorCount": 3857,
          "type": "telemetryerror"
        }
```

i/o タイムアウト以外のもう 1 つの一般的なエラーは、「ホストへのルートがない」ことです。この場合、ネットワークまたは Threat Defense デバイス構成にルーティングの問題があります。

エラー タイプの詳細については、ステータス エラーのカテゴリ （13 ページ） を参照してください。

ステータス エラーのカテゴリ

テレメトリ ストリーミング サービスでエラーが発生した場合、サービス ステータスが [接続済み（Connected）] か [切断済み（Disconnected）] かには関係なく、テレメトリ ストリーミング ステータス情報に、発生した問題に関連するエラー メッセージが含まれています。エラー メッセージには、エラーが発生した時刻、エラーコード、および説明が含まれます。

メッセージは次のカテゴリに分類されます。

中止エラー（コード 0）

テレメトリ サービスの中止につながるエラー。このエラーが続く場合には、シスコのテクニカル サポートに問い合わせてください。

システム エラー（コード 1 ～ 10）

これらのエラーは、通常、ホスト名またはUUIDが構成されていなかったことを意味します。これは、テレメトリ ストリーミングが正しく構成されていないことを意味します。構成をやり直してください。

バッファ エラー（コード 11 ～ 20）

これらのエラーは、メトリック変換の問題など、メトリック バッファに関連しています。このエラーが続く場合には、シスコのテクニカル サポートに問い合わせてください。

認証エラー（コード 21 ～ 30）

これらのエラーは、証明書に問題があることを示しています。メッセージを評価し、証明書の問題を解決してください。

たとえば、証明書の有効期限が切れている場合は、新しい有効な証明書をアップロードして、サービスを再度有効にする必要があります。すべての証明書が同じ認証局によって署名されていることを確認します。

ストリーミング エラー（コード 31 ～ 40）

これらのエラーは、Threat Defense デバイスとテレメトリ コレクタとの接続に関連しています。問題には、IP アドレスとポート番号の間違い、または DNS 解決の問題が含まれる場合があります。また、ルーティングの問題と関連している場合もあります。

修正には、テレメトリ ストリーミング構成のやり直しか、ネットワークのルーティング問題の解決が必要となる場合があります。ルーティング/DNS の問題は、アップストリーム リンクや DNS サーバーのダウンなど、一時的な問題による場合もあります。

引数無効のエラー（コード 41 ～ 50）

これらのエラーは、メッセージで返された誤ったデータに関連しています。たとえば、プロトコルのバージョンが間違っているか、間隔頻度が範囲外である、などです。これらの

エラーでは、Threat Defense デバイスではなくテレメトリ コレクタを修正する必要があります。これらのエラーが表示された場合でも、サービス ステータスが [接続済み（Connected）] のままになる場合があることに注意してください。たとえば、許容される最小値が 1 分であるのに、間隔が 30 秒になっている場合、値が範囲外であるため、次のエラーが発生します。

```
"state": "CONNECTED",
      "errors": [
        {
          "errorType": "InvalidArgumentErrors",
         "errorMessage": "2024-09-30 19:20:33.575156378 +0000 UTC m=+308552.647839001
 : Possible proto version mismatch or invalid streaming interval -
client=192.168.97.90, proto version=0.0.1, streaming interval=30s",
          "type": "telemetryerror"
        }
],
```

テレメトリ コレクタがデータを受信することを確認する

テレメトリ コレクタが Threat Defense デバイスから情報を受信していることを確認します。テレメトリ コレクタ コンソールに関連するメッセージとデータが表示されます。例：

```
2024/08/19 11:08:58 D! [grpc_client] Streaming interval set to=1m0s
2024/08/19 11:08:58 D! [grpc_client] Starting listener, attempting to listen at
address=:50051 over tcp
2024/08/19 11:08:58 D! [grpc_client] CERT_PATH: /root/grpc-certs/keys
2024/08/19 11:08:58 D! FTD signalling listener running on port=8087
2024/08/19 11:08:58 D! [grpc_client] Collector server started at port=50051
2024/08/19 11:09:24 D! [grpc_client] DataStream RPC invoked
2024/08/19 11:09:24 D! [grpc_client] ControlStream RPC invoked
2024/08/19 11:09:24 D! [grpc_client] Receiving metrics from
device=firepower-7eb19498-2519-11ef-a8dd-b74b4d43a7e7
2024/08/19 11:09:24 D! [grpc_client] ControlStream - received done signal
2024/08/19 11:09:24 D! [grpc_client] RPC
context=&{device:firepower-7eb19498-2519-11ef-a8dd-b74b4d43a7e7 controlStream:0xc000024120
 dataStream:0xc000096020}
2024/08/19 11:14:24 D! [grpc_client] info - received metric batch of count=479 from
device=firepower-7eb19498-2519-11ef-a8dd-b74b4d43a7e7
2024/08/19 11:14:24 D! [grpc_client] METRIC=timestamp:1718257445000
metricFamily:"interface"  value:35386  tags:{key:"duplex_mode"  value:"FULL"}
tags:{key:"interface"  value:"GigabitEthernet0/0"}  tags:{key:"interface_description"}
 tags:{key:"interface_name"  value:"inside_interface"}  tags:{key:"interface_type"
value:"GigabitEthernet"}  tags:{key:"mac_address"  value:"0050.5683.0a21"}
tags:{key:"uuid"  value:"7eb19498-2519-11ef-a8dd-b74b4d43a7e7"}  tags:{key:"description"
  value:"input_packets"}  metricType:"Counter"
```

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。