Secure Firewall Threat Defense REST API について

HTTPS 経由で Secure Firewall Threat DefenseREpresentational State Transfer（REST）アプリケーションプログラミングインターフェイス（API）を使用すると、クライアントプログラムを使用して 脅威に対する防御 デバイスと通信できます。REST API は、JavaScript Object Notation（JSON）形式を使用してオブジェクトを表します。

Secure Firewall Device Managerには、プログラムで使用可能なすべてのリソースおよびJSON オブジェクトを説明するAPIエクスプローラが含まれます。エクスプローラは各オブジェクトの属性と値のペアについて詳細情報を提供するため、さまざまな HTTP メソッドを試して各リソースに必要なコーディングを理解することができます。APIエクスプローラでは、各リソースに必要な URL の例も示します。

https://developer.cisco.com/site/ftd-api-reference/ では、参照情報と例をオンラインで検索することもできます。

API には独自のバージョン番号があります。API の 1 つのバージョン用に設計されたクライアントが将来のバージョンでエラーなく動作したり、プログラムへの変更が不要である保証はありません。

このプログラミング ガイドの対象読者

このガイドは、プログラミングの一般的な知識と、REST API および JSON の一定の理解があることを前提に書かれています。これらのテクノロジーになじみがない場合は、最初に REST API の一般的なガイドをお読みください。

サポートされる HTTP メソッド

次の HTTP メソッドのみを使用できます。他のメソッドはサポートされません。

- GET：システムからデータを読み取ります。
- POST：新しいオブジェクトを作成します。
- PUT：既存のオブジェクトを変更します。PUT を使用する場合は、JSON オブジェクト全体を含める必要があります。オブジェクト内の個々の属性を選択的に更新することはできません。
- DELETE：ユーザ定義オブジェクトを削除します。

API のベース URL

指定した 脅威に対する防御 デバイスのベース URL を決定する最も簡単な方法は、API エクスプローラで GET メソッドを試し、結果から URL のオブジェクト部分を削除することです。

たとえば、GET /object/networks を実行して、返される出力の要求 URL の下に、次に示すようなものを見ることができます。

```
https://ftd.example.com/api/fdm/v1/object/networks
```

URL のサーバー名の部分は、脅威に対する防御デバイスのホスト名または IP アドレスで、「ftd.example.com」の部分が使用デバイスによって異なります。この例では、パスから /object/networks を削除してベース URL を取得します。

```
https://ftd.example.com/api/fdm/v1/
```

リソースのすべての呼び出しは、要求 URL のベースとしてこの URL を使用します。

HTTPS データポートを変更した場合は、URL にカスタムポートを含める必要があります。たとえば、ポートを 4443 に変更した場合は、https://ftd.example.com:4443/api/fdm/v1/ のような URL にします。

URL の要素「v」は API バージョンです。通常、これはソフトウェアバージョンに応じて変化します。たとえば、脅威に対する防御 バージョン 6.3.0 の API バージョンは v2 です。そのため、ベース URL は次のようになります。

```
https://ftd.example.com/api/fdm/v2/
```

（注） 脅威に対する防御 6.4 以降では、パス内の v 要素の代わりに **latest** を使用することで、API コール内のパスを更新する必要性を省くことができます。たとえば、https://ftd.example.com/api/fdm/latest/ などとします。**latest** エイリアスは、デバイスでサポートされている最新の API バージョンに解決されます。

APIエクスプローラで、ページの一番下にスクロールすると、ベースURL（サーバ名を除く）および API のバージョンに関する情報を見ることができます。

REST API の SSL/TLS 通信の保護

Threat Defense デバイスには自己署名証明書が付属しているため、デバイスとの HTTPS 通信を開始できます。ただし、証明書は既知の認証局（CA）によって署名されていないため、SSL/TLS によるアクセス試行はすべて、接続が安全でないと見なされます。

ブラウザに接続すると、自己署名証明書を受け入れるように求められますが、curl などのコマンドでは証明書が拒否されます。Curl の場合は、**--insecure** キーワードを追加することによって、証明書チェックの失敗をバイパスできます。次に例を示します。

```
curl --insecure -X GET --header 'Accept: application/json'
'https://ftd.example.com/api/versions'
```

最初に行う必要がある操作の 1 つは、脅威に対する防御 デバイス用の CA 署名付きデバイス証明書の取得です。次に、Device Manager または API を使用して、この証明書を管理証明書として割り当てます。それ以降、SSL/TLS 証明書のチェックは失敗せず、API コールで安全でない通信を使用する必要はなくなります。

手順

ステップ 1 **POST /object/internalcertificates** リソースを使用して、CA 署名付きデバイス証明書をアップロードします。

ステップ 2 **PUT /devicesettings/default/webuicertificates/{objId}** リソースを使用して、この証明書を管理証明書にします。

GET /devicesettings/default/webuicertificates リソースを使用して、Web UI 証明書のオブジェクト ID を調べます。

ステップ 3 **POST /operational/deploy** リソースを使用して、変更を展開します。

サポートされる API バージョンの判別

GET /api/versions（ApiVersions）メソッドを使用して、デバイスでサポートされる API バージョンを判別できます。このメソッドでは認証は不要で、パスに version 要素を含める必要もありません。次に例を示します。

```
curl -X GET --header 'Accept: application/json' 'https://ftd.example.com/api/versions'
```

「ftd.example.com」を Threat Defense デバイスのホスト名または IP アドレスに置き換えます。

このメソッドは、使用可能な API バージョンのリストを返します。次に例を示します。

```
{
    "supportedVersions":["v3", "latest"]
}
```

バージョン文字列は、以降の API コールの URL で使用するものと同じです。特定のバージョン識別子の代わりに **latest** を使用した場合は、以降のリリースのコールを更新する必要性を省くことができます。ただし、この手法を使用しても、コールで使用するオブジェクトモデルに対する変更は解決されず、場合によってはリリースごとに調整が必要になります。

通常、次の手順は、OAuth を使用した REST API クライアントの認証 で説明されているようにアクセス トークンを取得することです。

API バージョンの後方互換性

Threat Defense API のバージョンは、Threat Defense ソフトウェアのメジャーリリースごとに変更されます。新しい機能は、追加または変更された機能の API コールに影響します。

とはいえ、リリースごとに多くの機能が変更されることはありません。たとえば、ネットワークおよびポートオブジェクトに関連した API は、新しいリリースでは変更されないことがよくあります。

Threat Defense バージョン 6.7 以降、ある機能の API リソースモデルがリリース間で変更されない場合、Threat Defense API は古い API バージョンに基づくコールを受け入れることができます。機能モデルが変更された場合でも、古いモデルを新しいモデルに変換する論理的な方法があれば、古いコールが機能します。たとえば、v5 コールを v6 システムで受け入れることができます。コールのバージョン番号として「latest（最新）」を使用する場合、「古い」コールは、このシナリオでは v6 コールとして解釈されるため、下位互換性を利用するかどうかは、API コールの構築方法によって決まります。

後方互換性がサポートされない方法で API バージョン間で機能モデルが変更された場合、エラーメッセージが表示されます。これらのエラーを確認し、特定のコールのコードを更新する必要があります。

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。