Secure Firewall Threat Defense REST API 정보

HTTPS를 통해 Secure Firewall Threat Defense REST(Representational State Transfer) API(Application Programming Interface)를 사용하여 클라이언트 프로그램을 통해 위협 방어 디바이스와 상호 작용할 수 있습니다. REST API는 JSON(JavaScript Object Notation) 형식을 사용하여 개체를 표시합니다.

Secure Firewall device manager에는 프로그래밍 용도로 사용할 수 있는 모든 리소스 및 JSON 개체를 설명하는 API Explorer가 포함됩니다. API Explorer는 각 개체의 '특성-값' 쌍에 대한 자세한 정보를 제공합니다. 따라서 다양한 HTTP 방법을 실험하여 각 리소스를 사용하는 데 필요한 코딩을 파악할 수 있습니다. API Explorer는 각 리소스에 필요한 URL 예시도 제공합니다.

https://developer.cisco.com/site/ftd-api-reference/에서 참조 정보 및 예시 온라인을 확인할 수도 있습니다.

API에는 고유한 버전 번호가 있습니다. API의 한 버전을 위해 설계된 클라이언트가 오류 없이 또는 프로그램을 변경하지 않고도 향후 버전에서 작동한다는 보장은 없습니다.

이 프로그래밍 가이드의 대상 독자

이 가이드는 대상 독자가 프로그래밍에 대한 일반적인 지식을 갖추고 있으며 REST API 및 JSON에 대해 명확하게 이해하고 있다는 가정 하에 작성되었습니다. 이러한 기술을 처음 접하는 경우, REST API에 대한 일반적인 가이드를 먼저 읽어보십시오.

지원되는 HTTP 방법

다음과 같은 HTTP 방법만 사용할 수 있으며, 다른 방법은 지원되지 않습니다.

- GET — 시스템에서 데이터를 읽을 때 사용합니다.

- POST — 새로운 개체를 생성할 때 사용합니다.
- PUT — 기존 개체를 수정할 때 사용합니다. PUT을 사용할 때는 전체 JSON 개체를 포함해야 합니다. 개체 내에서 개별 특성을 선택적으로 업데이트할 수는 없습니다.
- DELETE — 사용자 정의 개체를 제거할 때 사용합니다.

API의 기본 URL

특정 위협 방어 디바이스의 기본 URL을 확인하는 가장 쉬운 방법은 API Explorer에서 GET 방법을 시도하는 것으로, 그 결과에서 URL의 개체 부분을 삭제하기만 하면 됩니다.

예를 들어, GET /object/networks를 수행하고 요청 URL 아래에 반환된 출력에서 다음과 유사한 출력을 확인할 수 있습니다.

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

URL의 서버 이름 부분은 위협 방어 디바이스의 호스트네임 또는 IP 주소이며, 사용자 디바이스의 경우 "ftd.example.com"이 다른 내용으로 대체됩니다. 이 예에서 기본 URL을 얻으려면 경로에서 /object/networks를 삭제합니다.

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

모든 리소스 호출은 이 URL을 요청 URL에 대한 기본 URL로 사용합니다.

HTTPS 데이터 포트를 변경한 경우 URL에 맞춤형 포트를 포함해야 합니다. 예를 들어 포트를 4443으로 변경한 경우 https://ftd.example.com:4443/api/fdm/v1/과 같습니다.

URL의 "v" 요소는 API 버전이며 일반적으로 소프트웨어 버전과 함께 변경됩니다. 예를 들어 위협 방어 버전 6.3.0의 API 버전은 v2이므로 기본 URL은 다음과 같습니다.

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

참고 위협 방어 6.4부터는 경로에서 v 요소 대신 **latest(**최신**)** 버전을 사용하여 API 호출에서 경로를 업데이트하지 않아도 됩니다. 예를 들어, https://ftd.example.com/api/fdm/latest/와 같습니다. **latest(**최신**)**라는 별칭은 디바이스에서 지원하는 최신 API 버전으로 해석됩니다.

API Explorer에서 페이지 하단으로 스크롤하면 기본 URL(서버 이름 제외)과 API 버전에 대한 정보를 확인할 수 있습니다.

REST API에 대한 SSL/TLS 통신 보안 유지

Threat Defense 디바이스는 디바이스와의 HTTPS 통신을 시작할 수 있도록 자체 서명 인증서와 함께 제공됩니다. 그러나 인증서가 알려진 CA(Certificate Authority)를 통해 서명되지 않기 때문에 모든 SSL/TLS 액세스 시도에서 연결을 안전하지 않은 것으로 간주합니다.

브라우저에 연결할 때 자체 서명 인증서를 수락하라는 프롬프트가 표시되지만, curl과 같은 명령에서는 인증서를 거부합니다. curl의 경우 **--insecure** 키워드를 추가하여 인증서 확인 실패를 우회할 수 있습니다. 예를 들면 다음과 같습니다.

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

가장 먼저 수행해야 하는 작업 중 하나는 위협 방어 디바이스에 대한 CA 서명 디바이스 인증서를 획득하는 것입니다. 그런 다음 device manager 또는 API를 사용하여 이 인증서를 관리 인증서로 할당합니다. 이후에는 SSL/TLS 인증서 검사가 실패하지 않으며 API 호출에서 보안되지 않은 통신을 사용할 필요가 없습니다.

프로시저

단계 **1** **POST /object/internalcertificates** 리소스를 사용하여 CA 서명 디바이스 인증서를 업로드합니다.

단계 **2** **PUT /devicesettings/default/webuicertificates/{objId}** 리소스를 사용하여 이 인증서를 관리 인증서로 설정합니다.

GET /devicesettings/default/webuicertificates 리소스를 사용하여 웹 UI 인증서의 개체 ID를 확인합니다.

단계 **3** **POST /operational/deploy** 리소스를 사용하여 변경 사항을 구축합니다.

지원되는 API 버전 확인

GET /api/versions (ApiVersions) 메서드를 사용해 디바이스에서 지원되는 API 버전을 확인할 수 있습니다. 이 메서드에서는 인증이 필요 없고 경로에 버전 요소를 포함하지도 않습니다. 예를 들면 다음과 같습니다.

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

threat defense 디바이스의 호스트네임 또는 IP 주소를 "ftd.example.com"으로 대체합니다.

이 메서드에서는 사용할 수 있는 API 버전의 목록을 반환합니다. 예를 들면 다음과 같습니다.

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

버전 문자열은 후속 API 호출에 대한 URL에서 사용하는 것과 동일합니다. 특정 버전의 식별자 대신 **latest(**최신**)** 버전을 사용하는 경우, 후속 릴리스를 위해 호출을 업데이트하지 않아도 됩니다. 그러나 이 기술을 사용해도 호출에 사용된 개체 모델에 대한 변경 사항은 해결되지 않으므로 릴리스에서 릴리스로 조정해야 할 수 있습니다.

일반적으로 다음 단계는 OAuth를 사용한 REST API 클라이언트 인증에 설명된 대로 액세스 토큰을 가져오는 것입니다.

API 버전 이전 버전과의 호환성

threat defense API 버전은 threat defense 소프트웨어의 각 주요 릴리스마다 변경됩니다. 새로운 기능은 추가 또는 변경되는 기능에 대한 API 호출에 영향을 줍니다.

그러나, 많은 기능은 릴리스에서 릴리스로 변경되지 않습니다. 예를 들어 네트워크 및 포트 개체와 관련된 API는 종종 새 릴리스에서 변경되지 않은 상태로 유지됩니다.

threat defense 버전 6.7부터는 기능에 대한 API 리소스 모델이 릴리스 간에 변경되지 않는 경우 threat defense API는 이전 API 버전을 기반으로 하는 통화를 수락할 수 있습니다. 기능 모델이 변경된 경우에도 이전 모델을 새 모델로 변환하는 논리적 방법이 있는 경우 이전 콜을 사용할 수 있습니다. 예를 들어, v5 통화는 v6 시스템에서 허용될 수 있습니다. "최신"을 통화의 버전 번호로 사용하는 경우 이러한 "이전" 통화는 이 시나리오에서 v6 통화로 해석되므로 이전 버전과의 호환성을 활용하고 있는지 여부는 API 호출을 어떻게 구성할지에 따라 달라집니다.

이전 버전과의 호환성을 지원할 수 없는 방식으로 API 버전 간에 기능 모델이 변경된 경우, 오류 메시지가 표시되며 이러한 오류를 확인하고 해당 특정 통화에 대한 코드를 업데이트해야 합니다.