

REST-API를 사용하여 대량 객체를 FMC로 푸시합니다.

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[제한 사항](#)

[배경 정보](#)

[구성](#)

[다음을 확인합니다.](#)

[문제 해결](#)

소개

이 문서에서는 API(Application Programming Interface) 관리자가 네트워크, 포트 및 URL 객체를 FMC(Firepower Management Center)에 대량으로 푸시하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- 다양한 REST API 호출 이해([REST API란 무엇입니까?](#))
- FMC [API 빠른 시작 가이드 검토](#)
- FMC 재사용 [가능 객체 검토](#)
- Python 요청 라이브러리에 대한 기본 지식

사용되는 구성 요소

- REST API가 활성화된 REST API(버전 6.1 이상)를 지원하는 Firepower Management Center
- Python을 사용한 REST API 상호 작용

제한 사항

- FMC는 객체 이름을 64자보다 길 수 없습니다.
- 객체 이름은 객체 이름의 시작 부분에 공백을 사용할 수 없고 끝 부분에는 세미콜론을 사용할 수 없습니다.
- 페이로드는 단일 대량 푸시에 1,000개 이상의 항목을 포함할 수 없습니다.
- 단일 대량 푸시에서는 페이로드 크기가 2MB를 초과할 수 없습니다.

배경 정보

REST API는 네트워크 관리자가 네트워크를 구성 및 관리하는 데 사용할 수 있는 가벼운 프로그래밍 방식 때문에 점점 더 인기를 끌고 있습니다. FMC는 모든 REST 클라이언트를 사용하여 컨피그레이션 및 관리를 지원하며, 내장된 API 탐색기도 사용합니다.

이 문서의 예에서는 CSV 파일을 입력으로 가져와 REST API 인터페이스를 통해 객체를 FMC에 푸시합니다. 이 문서에서는 호스트 네트워크 벌크 푸시만 다루며 다른 객체에 대해 유사한 로직을 확장할 수 있습니다. 샘플 코드는 URL 및 Port 개체에 대한 문서에 첨부됩니다.

다음은 이미지에 표시된 대로 사용되는 네트워크 호스트의 POST에 대한 API 참조입니다.

POST /api/fmc_config/v1/domain/{domainUUID}/object/hosts

Retrieves, deletes, creates, or modifies the host object associated with the specified ID. If no ID is specified for a GET, retrieves list of all host objects. Check the response section for applicable examples (if any).

Parameters Try it out

Name	Description
body * required object (body)	Input representation of host object. Parameter content type application/json
bulk boolean (query)	Enables bulk create for host objects. --
domainUUID * required string (path)	Domain UUID e276abec-e0f2-11e3-8169-6d9ed49b625f

Responses Response content type application/json

Code	Description
201	Created Example Value Model Request example 1 : POST /fmc_config/v1/domain/domainUUID/object/hosts (POST to create a host object) <pre>{ "name": "TestHost", "type": "Host", "value": "10.5.3.20", "description": "Test Description" }</pre> Request example 2 : POST /fmc_config/v1/domain/domainUUID/object/hosts?bulk=true (Bulk POST operation for Host object) <pre>[{ "name": "host1", "type": "Host", "value": "10.5.3.20", "description": "Test Description" }, { "name": "Host2", "type": "Host", "value": "1.2.3.4", "description": "Host object 2" }]</pre>

구성

1단계. REST API를 활성화하고 인증 토큰을 생성합니다. 자세한 컨피그레이션 단계 및 예는 FMC에서 [인증 토큰 생성을 참조하십시오](#).

```
import requests
import csv
import json
from requests.auth import HTTPBasicAuth
from getpass import getpass
address = input("Enter IP Address of the FMC: ")
username = input("Enter
```

```

Username: ") password = getpass("Enter Password: ") api_uri =
"/api/fmc_platform/v1/auth/generatetoken" url = "https://" + address + api_uri response =
requests.request("POST", url, verify=False, auth=HTTPBasicAuth(username, password)) accesstoken
= response.headers["X-auth-access-token"] refreshtoken = response.headers["X-auth-refresh-
token"] DOMAIN_UUID = response.headers["DOMAIN_UUID"]

```

2단계. 제공된 CSV 파일을 사전에 변환하여 요청에 대한 JSON 페이로드로 사용합니다. 각 객체 유형에 대한 샘플 CSV 파일이 문서에 첨부됩니다.

	A	B	C	D
1	name	description	type	value
2	Host-1	Host-1	Host	10.10.10.10
3	Host-2	Host-2	Host	10.10.10.1
4	Network-1	Network-1	Network	10.10.9.0/24
5	Host-3	Host-3	Host	10.10.10.2
6	Range-1	Rannge-1	Range	10.20.20.1-10.20.20.20
7				

```

csvFilePath = input("Please enter the CSV Filepath (For eg. : path/to/file/objects.csv) :") host
= [] with open(csvFilePath, encoding='utf-8-sig') as csvf: csvReader = csv.DictReader(csvf) for
rows in csvReader: if rows['type'] == "Host": host.append(rows) host_payload = json.dumps(host)
이 단계의 host_payload는 이미지에 표시된 것과 같습니다.

```

```

[{"name": "Host-1", "description": "Host-1", "type": "Host", "value": "10.10.10.10" }, {
"name": "Host-2", "description": "Host-2", "type": "Host", "value": "10.10.10.1" }, {"name":
"Host-3", "description": "Host-3", "type": "Host", "value": "10.10.10.2" } ]

```

3단계. 이전 단계에서 받은 입력에서 요청을 생성하고 페이로드가 비어 있지 않으면 요청을 보냅니다.

```

host_api_uri = "/api/fmc_config/v1/domain/" + DOMAIN_UUID + "/object/hosts?bulk =true" host_url
= "https://" + address + host_api_uri headers = { 'Content-Type': 'application/json', 'x-auth-
access-token': accesstoken } if host != []: response = requests.request("POST", host_url,
headers=headers, data = host_payload, verify = False) else : print("Please Validate that the CSV
file provided is correct or at correct location")

```

다음을 확인합니다.

- 응답의 상태 코드를 인쇄하여 요청이 성공 또는 실패했는지 확인합니다(아래 참조).

```

if response.status_code == 201 or response.status_code == 202: print("Host Objects successfully
pushed") else: print("Host Object creation failed")

```

- FMC에 로그인 다음 이미지에 표시된 대로 Object(개체) > Object Management(개체 관리) > Network(네트워크)로 이동하여 Host Objects(호스트 개체)를 확인합니다.

Network

Add Network ▼

A network object represents one or more IP addresses. Network objects are used in various places, including access control policies, network variables, intrusion discovery rules, event searches, reports, and so on.

Name	Value	Type
Host-1	10.10.10.10	Host
Host-2	10.10.10.1	Host
Host-3	10.10.10.2	Host

문제 해결

- REST 클라이언트를 사용할 때 자체 서명 인증서로 인해 SSL 인증서 문제와 관련된 오류가 표시될 수 있습니다. 사용 중인 클라이언트에 따라 이 유효성 검사를 해제할 수 있습니다.
- FMC REST API 인증 토큰은 30분 동안 유효하며 최대 3회 새로 고칠 수 있습니다.
- 요청과 관련된 오류는 응답 본문에서 추출할 수 있습니다. 이를 로그 파일로 수집하여 문제 해결을 지원할 수 있습니다.

```
logfile = "requestlog.txt" log = open(logfile, "w+") log.write(response.text) log.close
```

- 모든 REST 요청은 FMC의 이 두 로그 파일에 기록됩니다. URL 검색(예: .../object/hosts) 올바른 작업(GET 작업에 대한 오류를 찾는 경우 로그가 GET... 등의 작업을 시작하는지 확인하십시오. 객체/호스트)

```
tail -f /var/opt/CSC0px/MDC/tomcat/logs/stdout.logs tail -f /var/opt/CSC0px/MDC/log/operation/usmshredsvcs.log
```