

Duw objecten in bulk op FMC met behulp van REST-API

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Beperkingen](#)

[Achtergrondinformatie](#)

[Configureren](#)

[Verifiëren](#)

[Problemen oplossen](#)

Inleiding

Dit document beschrijft hoe een API-beheerder (Application Programming Interface) netwerk-, poort- en URL-objecten in bulk naar Firepower Management Center (FMC) kan duwen.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Het begrip van verschillende REST API-oproepen. ([Wat zijn REST API's?](#))
- Review of the [FMC API Quick Start Guide](#)
- Beoordeling van [herbruikbare objecten van het FMC](#)
- Basiskennis van de bibliotheek van Python-verzoeken

Gebruikte componenten

- Firepower Management Center dat REST API's (versie 6.1 of hoger) ondersteunt met REST API ingeschakeld
- REST API-interacties met Python.

Beperkingen

- FMC aanvaardt niet dat de naam van het object groter is dan 64 tekens.
- De naam van object moet geen ruimte hebben aan het begin van de objectnaam en de puntkomma aan het einde.
- De lading kan niet meer dan 1000 ingangen in één druk van het Bulk bevatten.
- De payloadgrootte kan niet meer dan 2 MB bedragen in één enkele Bulk Push.

Achtergrondinformatie

REST API's zijn steeds populairder vanwege de lichtgewicht programmeerbare benadering die netwerkbeheerders kunnen gebruiken om hun netwerken te configureren en te beheren. FMC ondersteunt configuratie en beheer met behulp van elke REST-client en ook met behulp van de ingebouwde API-verkenner.

Het voorbeeld in dit document neemt een CSV-bestand als invoer en drukt de objecten op FMC via de REST API-interface. Het document bestrijkt alleen de Host Network Bulk-toets en een soortgelijke logica kan voor de andere objecten worden uitgebreid. Er wordt een voorbeeldcode aan het document toegevoegd voor URL- en poortobjecten.

Hier is de API verwijzing voor POST op netwerkhosts die worden gebruikt, zoals in de afbeelding wordt getoond:

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> |

Configureren

Stap 1. Schakel REST API in en genereer verificatie-Token. Voor gedetailleerde configuratiestappen en voorbeelden moet u [verificatie genereren op FMC](#).

```
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"]
```

Stap 2. Converteer het CSV-bestand naar een woordenboek dat moet worden gebruikt als JSON Payload voor het verzoek. Een voorbeeldbestand van CSV-bestand voor elk doelttype wordt aan het document toegevoegd.

| | 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)
```

De host_payload ziet er in dit stadium hetzelfde uit als in de afbeelding:

```
[{ "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" } ]
```

Stap 3. Maak het verzoek op basis van de input die uit vorige stappen is ontvangen en verstuur het verzoek als de lading niet leeg is.

```
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")
```

Verifiëren

- Druk de statuscode van het antwoord af om te controleren of het verzoek succesvol was of niet, zoals hier wordt getoond.

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

- Meld u aan bij FMC navigeren om **object > Objectbeheer > Netwerk** en controleer de

hostobjecten, zoals in de afbeelding:

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 |

Problemen oplossen

- Wanneer u de REST-client gebruikt, kunt u fouten zien die te maken hebben met het SSL-certificeringsprobleem door een zelfondertekend certificaat. U kunt deze validatie uitzetten afhankelijk van de client die u gebruikt.
- De FMC REST API-verificatiepenningen zijn geldig voor 30 minuten en kunnen tot drie keer worden bijgewerkt.
- De fout in verband met het verzoek kan worden afgeleid uit de responsinstantie. Dit kan worden verzameld als een logbestand om te helpen bij het oplossen van problemen.

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

- Alle REST-verzoeken worden op FMC in deze twee logbestanden vastgelegd. Zoek naar uw URL (bijvoorbeeld `.../object/hosts`) met de juiste werking (als u op zoek bent naar een fout voor de GET operatie, zorg er dan voor dat uw logbestand start zoals `GET ...objecten/hosts`)

```
tail -f /var/opt/CSCOpX/MDC/tomcat/logs/stdout.logs tail -f /var/opt/CSCOpX/MDC/log/operation/usmsharredsvcs.log
```