

Verschieben mehrerer Objekte auf FMC mithilfe der REST-API

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Einschränkungen](#)

[Hintergrundinformationen](#)

[Konfigurieren](#)

[Überprüfen](#)

[Fehlerbehebung](#)

Einführung

In diesem Dokument wird beschrieben, wie ein Administrator einer API (Application Programming Interface) Netzwerk-, Port- und URL-Objekte größtenteils an das FirePOWER Management Center (FMC) übertragen kann.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über Kenntnisse in folgenden Bereichen zu verfügen:

- Verstehen verschiedener REST API-Aufrufe. ([Was sind REST-APIs?](#))
- Übersicht über die [FMC API-Schnellstartanleitung](#)
- Überprüfung wiederverwendbarer [FMC-Objekte](#)
- Grundkenntnisse der Python-Anforderungsbibliothek

Verwendete Komponenten

- FirePOWER Management Center unterstützt REST-APIs (Version 6.1 oder höher) mit aktivierter REST-API
- REST API-Interaktionen mit Python.

Einschränkungen

- FMC akzeptiert nicht den Namen des Objekts, der größer als 64 Zeichen sein soll.
- Der Objektname sollte am Anfang des Objektnamen und am Ende des Semikolons kein Leerzeichen haben.
- Die Nutzlast darf nicht mehr als 1.000 Einträge in einem einzelnen Bulk Push enthalten.

- Die Payload-Größe darf in einem einzelnen Bulk Push nicht mehr als 2 MB betragen.

Hintergrundinformationen

REST-APIs sind aufgrund des einfachen programmierbaren Ansatzes, den Netzwerkmanager zur Konfiguration und Verwaltung ihrer Netzwerke verwenden können, zunehmend beliebt. FMC unterstützt die Konfiguration und Verwaltung über jeden REST-Client und den integrierten API-Explorer.

Das Beispiel in diesem Dokument nimmt eine CSV-Datei als Eingabe und überträgt die Objekte über die REST API-Schnittstelle an FMC. Das Dokument behandelt nur den Host Network Bulk-Push, und eine ähnliche Logik kann für die anderen Objekte erweitert werden. Dem Dokument wird ein Beispielcode für URL- und Port-Objekte angefügt.

Nachfolgend finden Sie die API-Referenz für den POST auf den verwendeten Netzwerkhosts, wie im Bild gezeigt:

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>

Konfigurieren

Schritt 1: Aktivieren Sie die REST-API, und generieren Sie Authentifizierungstoken. Ausführliche

Konfigurationsschritte und Beispiele finden Sie unter [Generate Authentication Token \(Authentifizierungstoken generieren\) auf 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"]
```

Schritt 2: Konvertieren Sie die bereitgestellte CSV-Datei in ein Wörterbuch, das als JSON-Payload für die Anforderung verwendet werden soll. Eine CSV-Beispieldatei für jeden Objekttyp ist an das Dokument angehängt.

	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)
```

Die host_payload sieht in diesem Stadium wie im Bild aus:

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

Schritt 3. Erstellen Sie die Anforderung aus der Eingabe, die Sie von vorherigen Schritten erhalten haben, und senden Sie die Anforderung, wenn die Nutzlast nicht leer ist.

```
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")
```

Überprüfen

- Drucken Sie den Statuscode der Antwort, um zu überprüfen, ob die Anfrage erfolgreich war oder fehlgeschlagen ist, wie hier gezeigt.

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

- Melden Sie sich bei FMC an, navigieren Sie zu **Object > Object Management > Network**, und überprüfen Sie die Hostobjekte, wie im Bild gezeigt:

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

Fehlerbehebung

- Wenn Sie den REST-Client verwenden, können aufgrund eines selbstsignierten Zertifikats Fehler im Zusammenhang mit dem SSL-Zertifikatproblem auftreten. Sie können diese Validierung je nach dem verwendeten Client deaktivieren.
- Die Authentifizierungs-Token für die FMC REST API sind 30 Minuten gültig und können bis zu dreimal aktualisiert werden.
- Der Fehler im Zusammenhang mit der Anforderung kann aus dem Antworttext extrahiert werden. Diese können als Protokolldatei zur Fehlerbehebung gesammelt werden.

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

- Alle REST-Anfragen sind bei diesen beiden Protokolldateien auf dem FMC angemeldet. Suchen Sie nach Ihrer URL (z. B. .../object/hosts) mit der korrekten Operation(Wenn Sie einen Fehler für GET-Operation suchen, stellen Sie sicher, dass das Protokoll beispielsweise GET ...Objekt/Hosts)

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