Diffuser des objets en masse vers FMC en utilisant REST-API

Table des matières

Introduction

Conditions préalables

- Exigences
- Composants utilisés
- Limites

Informations générales

Configurer

Vérifier

Dépannage

Introduction

Ce document décrit comment un administrateur d'interface de programmation d'applications (API) peut envoyer des objets réseau, de port et d'URL en masse vers Firepower Management Center (FMC).

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Compréhension des divers appels de l'API REST. (Que sont les API REST ?)
- Révision du guide de démarrage rapide de l'API FMC
- Examen des objets réutilisables FMC
- Connaissance de base de la bibliothèque de requêtes Python

Composants utilisés

- Firepower Management Center qui prend en charge les API REST (version 6.1 ou ultérieure) avec l'API REST activée
- Interactions de l'API REST avec Python.

Limites

- FMC n'accepte pas que le nom de l'objet dépasse 64 caractères.
- Le champ Nom de l'objet ne doit pas contenir d'espace au début du nom de l'objet et un

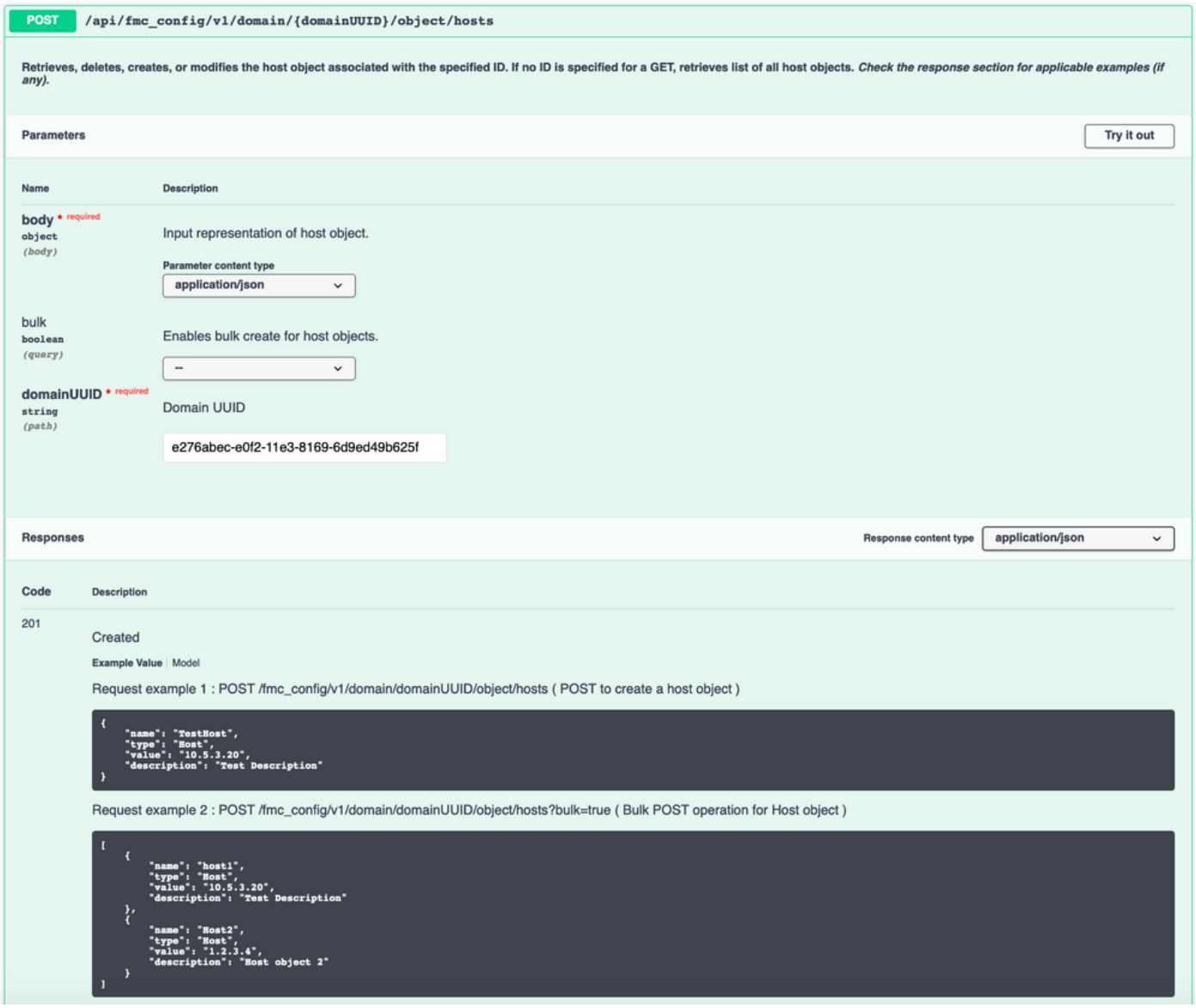
point-virgule à la fin.
- La charge utile ne peut pas contenir plus de 1 000 entrées dans une seule transmission en masse.
- La taille de la charge utile ne peut pas être supérieure à 2 Mo dans une seule transmission en masse.

Informations générales

Les API REST sont de plus en plus populaires en raison de l'approche programmable légère que les administrateurs réseau peuvent utiliser pour configurer et gérer leurs réseaux. FMC prend en charge la configuration et la gestion à l’aide de tout client REST et de l’explorateur d’API intégré.

L'exemple de ce document prend un fichier CSV comme entrée et pousse les objets vers FMC via l'interface API REST. Le document couvre uniquement la transmission groupée du réseau hôte et une logique similaire peut être étendue pour les autres objets. Un exemple de code est joint au document pour les objets URL et Port.

Voici la référence de l'API pour le POST sur les hôtes réseau utilisés, comme illustré dans l'image :



Configurer

Étape 1 : activation de l'API REST et génération du jeton d'authentification Pour des étapes de configuration détaillées et des exemples, référez-vous à Générer un jeton d'authentification sur 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"]
```

Étape 2.Convertissez le fichier CSV fourni en un dictionnaire à utiliser comme charge utile JSON pour la demande. Un exemple de fichier CSV pour chaque type d'objet est joint au document.

	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)
```

À ce stade, la charge utile host_payload est identique à celle illustrée dans l'image :

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

Étape 3. Créez la demande à partir de l'entrée reçue des étapes précédentes et envoyez la demande si la charge utile n'est pas vide.

```
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 }

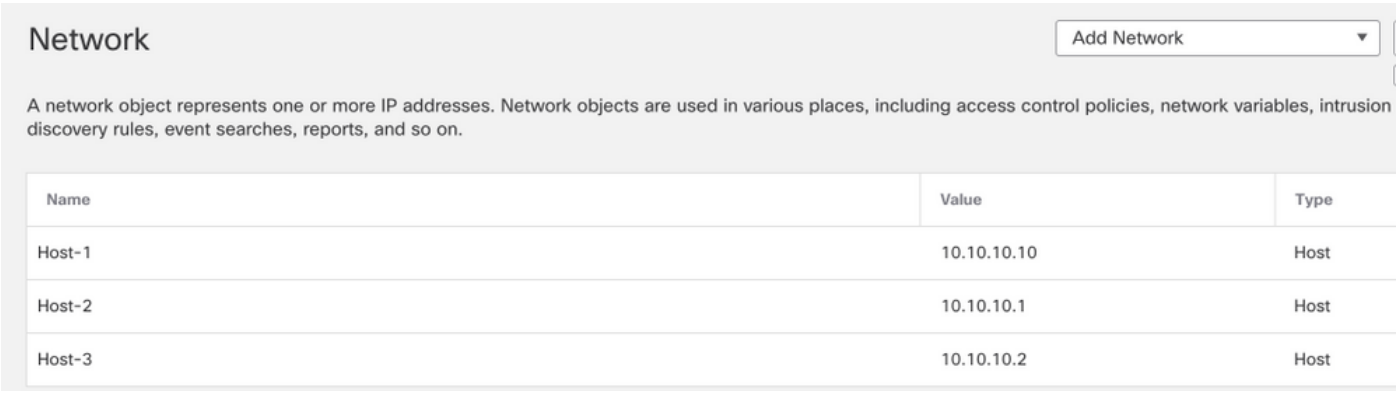
if host != []:
    response = requests.request("POST", host_url, headers=headers, data = host_payload, verify = False)
else :
    print("Please Validate that the CSV file provided is correct or at correct location")
```

Vérifier

- Imprimez le code d'état de la réponse pour vérifier si la demande a réussi ou échoué, comme indiqué ici.

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

- Connectez-vous à FMC Accédez à Object > Object Management > Network et vérifiez les objets hôtes, comme indiqué dans l'image :



Dépannage

- Lors de l'utilisation du client REST, vous pouvez voir des erreurs liées au problème de certificat SSL en raison d'un certificat auto-signé. Vous pouvez désactiver cette validation en fonction du client que vous utilisez.
- Les jetons d'authentification de l'API REST FMC sont valides pendant 30 minutes et peuvent être actualisés jusqu'à trois fois.
- L'erreur liée à la demande peut être extraite du corps de réponse. Il peut être collecté sous forme de fichier journal pour faciliter le dépannage.

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

- Toutes les requêtes REST sont consignées dans ces deux fichiers journaux sur FMC. Recherchez votre URL (par ex. .../object/hosts) avec l'opération correcte(Si vous recherchez une erreur pour l'opération GET, assurez-vous que votre journal démarre quelque chose comme GET ...objet/hôtes)

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

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.