

使用Python和AsyncOS REST API配置ESA字典

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[傳輸和驗證](#)

[背景資訊](#)

[設定](#)

[指令碼](#)

[驗證](#)

[相關資訊](#)

簡介

本文檔介紹如何使用Python指令碼使用AsyncOS API(REST API)將新詞典新增到思科郵件安全裝置(ESA)中。

必要條件

檢視以下前提條件：

- Python程式語言
- Python請求模組
- JSON
- 思科電子郵件安全裝置(ESA)

需求

使用以下軟體和訪問許可權運行指令碼：

- 運行AsyncOS 14.x或更高版本的思科電子郵件安全裝置(ESA)
- Python 3
- Python請求模組
- 具有使用AsyncOS API(REST API)許可權的ESA管理員憑據
- 通過埠6443上的HTTPS訪問ESA管理接口(推薦)

採用元件

完成本實驗所使用的元件有：

1)Cisco ESA - 15.0.0-104

2)優秀的文本編輯器

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

傳輸和驗證

- 傳輸：在埠6443上使用HTTPS用於AsyncOS API（如果可用）。僅在非生產或實驗室環境中使用HTTP/6080。
- 驗證:此示例使用HTTP基本身份驗證。提供標頭Authorization:基本<base64（使用者名稱：密碼）>。不要將真實憑證貼上到指令碼中。

背景資訊

對於本文檔：

目標：使用Python指令碼和AsyncOS API(REST API)將字典新增到ESA。

預期結果：API返回一個成功的響應，新的字典出現在ESA配置中。

設定

運行指令碼之前，在ESA管理介面上啟用AsyncOS API(REST API)服務。

1. 在圖形使用者介面(GUI)中，導航到包含AsyncOS API HTTP和AsyncOS API HTTPS設定的網路或介面頁面。
2. 啟用AsyncOS API HTTPS並驗證埠6443。
3. （可選）啟用AsyncOS API HTTP並驗證埠6080以供實驗室使用。
4. 按一下「Save」。

AsyncOS API

The Next Generation portal of your appliance uses AsyncOS API HTTP/HTTPS ports (6080/6443) and trailblazer HTTPS port (4431). You can use the trailblazerconfig command in the CLI to configure the trailblazer HTTPS ports. Make sure that the trailblazer HTTPS port is opened on the firewall.

☒ AsyncOS API HTTP

☒ AsyncOS API HTTPS

圖1:驗證管理介面上是否已啟用AsyncOS API HTTPS (埠6443)。

配置API日誌記錄以對AsyncOS API請求進行故障排除。

1. 在Graphical User Interface(GUI)中，導航到System Administration > Log Subscriptions。
2. 按一下Add Log Subscription。
3. 在「Log Type」下拉選單中，選擇「API Logs」。
4. 輸入其餘的必填欄位，如日誌名稱和檔名，然後按一下Submit。

New Log Subscription

Log Subscription

Log Type: API Logs

Log Name: api

File Name: api

Roller by File Size: 1GB

Roller by Time: None

Rate Limit: 60

Log Level: Information

Retrieval Method: Manually download logs from ahsan.esa.com

從API Logs 下拉選單Log Type中選擇，然後輸入其餘的必需詳情。

附註：可以通過GUI配置日誌。

指令碼

使用Python requests模組中的HTTP POST方法建立字典。

對POST請求使用以下參數：

參數	位置	必需	備註
device_type	URL查詢字串	是	設定為esa。
ignorecase	JSON負載	是	0 =區分大小寫，1 =不區分大小寫。
全字	JSON負載	是	0 =子字串匹配,1 =全字匹配。
編碼	JSON負載	是	除非需要不同的編碼，否則請使用utf-8。

單詞	JSON負載	是	條目陣列。將["term"]用於文字術語。對於智慧識別符號，使用ESA支援的元組格式，如["*credit", 2, "prefix"] (Smart Identifiers)。
模式	URL查詢字串	否	僅用於群集/組部署(例如mode=cluster)。

```

import base64
import requests

# ESA management IP address or FQDN
host = "

"

# Dictionary name to create
dictionary_name = "

"

# AsyncOS API endpoint (HTTPS recommended)
url = f"https://{host}:6443/esa/api/v2.0/config/dictionaries/{dictionary_name}?device_type=esa"

payload = {
    "data": {
        "ignorecase": 0,
        "wholewords": 1,
        "words": [
            ["*credit", 2, "prefix"],
            ["*aba"],
            ["Example Term"]
        ],
        "encoding": "utf-8"
    }
}

# Basic authentication header
username = "

"

password = "

"

creds = base64.b64encode(f"{username}:{password}".encode()).decode()

headers = {
    "Content-Type": "application/json",
    "Authorization": f"Basic {creds}"
}

response = requests.post(url, headers=headers, json=payload, timeout=30, verify=False)
print(f"Status code: {response.status_code}")
try:

```

```
print(response.json())
except ValueError:
    print(response.text)

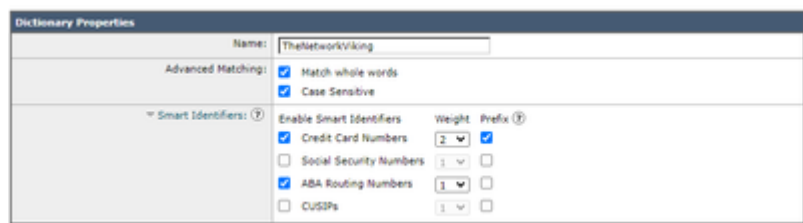
if response.status_code != 201:
    raise SystemExit("Dictionary creation failed.")
```

在指令碼中，詞典名稱()包含在請求URL路徑中。

在JSON負載中ignorecase包wholewords括所words需的encoding引數 (、和)。新增device_type到URL查詢字串。

僅mode當ESA位於群集或組中時，才需要此引數。在這種情況下，請按與以下方式相同的方式將mode=cluster附加到URL查詢字串： device_type.

*credit 並*aba新增到負載中，啟用信用卡號和ABA路由號的智慧識別符號，如螢幕截圖所示：



該術語將Example Term 作為要匹配的詞典項新增到負載中。

驗證

POST請求成功時，API日誌包含類似以下內容的條目：

```
API log example
Thu October 5 12:58:19 2023 Info: 198.51.100.10 - - 05/Oct/2023 12:58:19 +0000 POST /esa/api/v2.0/confi
```

預期響應：

```
{"data": {"message": "Added Successfully"}}
```

如果該字典已存在，並且POST請求再次運行，則API日誌包含類似以下內容的條目：

API log example

```
Thu October 5 13:00:31 2023 Info: 198.51.100.10 - - 05/Oct/2023 13:00:31 +0000 POST /esa/api/v2.0/confi
```

預期響應：

```
{"error": {"message": "Dictionary already exists.", "code": "409", "explanation": "409 = Request confli
```

相關資訊

使用以下外部資源檢視其他背景（本文檔中的過程是自包含的）：

[使用REST API + Python新增字典](#)

本文檔包括所需的身份驗證方法和最小過程。使用帶有標頭Authorization的HTTP基本身份驗證：基本<base64(username:password)>且優先連線埠6443上的HTTPS。

[REST API和Cisco ESA](#)

此外，此過程中使用的Python代碼可以從Postman中提取。如果需要，請檢視此影片教程：

[Postman — 如何生成Python指令碼](#)

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。