

使用Python和AsyncOS REST API配置ESA词典

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[传输和身份验证](#)

[背景信息](#)

[配置](#)

[脚本](#)

[验证](#)

[相关信息](#)

简介

本文档介绍如何使用Python脚本通过使用AsyncOS API(REST API)将新词典添加到思科邮件安全设备(ESA)。

先决条件

查看以下必备条件：

- Python编程语言
- Python请求模块
- JSON
- 思科邮件安全设备(ESA)

要求

使用以下软件和访问权限运行脚本：

- 运行AsyncOS 14.x或更高版本的思科邮件安全设备(ESA)
- Python 3
- Python请求模块
- 有权使用AsyncOS API(REST API)的ESA管理员凭证
- 通过端口6443上的HTTPS访问ESA管理接口（推荐）

使用的组件

本实验使用以下组件完成：

1)思科ESA - 15.0.0-104

2)优秀文本编辑器

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

传输和身份验证

- 传输：在端口6443上使用HTTPS用于AsyncOS API（如果可用）。仅在非生产或实验室环境中使用HTTP/6080。
- 身份验证:本示例使用HTTP基本身份验证。提供信头 授权：基本<base64(username:password)>。请勿将真实凭证粘贴到脚本中。

背景信息

对于本文档：

目标：使用Python脚本和AsyncOS API(REST API)将字典添加到ESA。

预期结果：API返回成功的响应，新词典显示在ESA配置中。

配置

在运行脚本之前，在ESA管理接口上启用AsyncOS API(REST API)服务。

1. 在图形用户界面(GUI)中，导航到包含AsyncOS API HTTP和AsyncOS API HTTPS设置的网络或接口页。
2. 启用AsyncOS API HTTPS并验证端口6443。
3. （可选）启用AsyncOS API HTTP并验证端口6080供实验室使用。
4. Click 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

图 1:验证管理接口上是否已启用AsyncOS API HTTPS (端口6443) 。

配置API日志记录以排除AsyncOS API请求故障。

1. 在图形用户界面(GUI)中，导航到系统管理>日志订阅。
2. 点击添加日志订阅。
3. 从Log Type下拉列表中，选择API Logs。
4. 输入其余的必填字段，如日志名称和文件名，然后单击Submit。

New Log Subscription

Log Subscription

Log Type: API Logs

Log Name: api

File Name: api

Append ☐ hostname ☐ serialnumber

Rollover by File Size: 10M

Rollover by Time: None

Rate Limit: ☐ Maximum number of messages to be logged: 60

Log Level: ☒ Information

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

从下API Logs拉列表Log Type中进行选择，然后输入剩余的所需的详细信息。



注意：可以从GUI配置日志。

脚本

使用Python请求模块中的HTTP POST方法创建字典。

将以下参数用于POST请求：

参数	位置	必需	备注
device_type	URL查询字符串	Yes	设置为esa。
ignorecase	JSON负载	Yes	0 =区分大小写，1 =不区分大小写。
全字	JSON负载	Yes	0 =子字符串匹配,1 =全字匹配。
编码	JSON负载	Yes	除非需要不同的编码，否则请使用utf-8。

字词	JSON负载	Yes	条目数组。将["term"]用于文字术语。对于智能标识符，请使用ESA支持的元组格式，如["*credit", 2, "prefix"]。
模式	URL查询字符串	无	仅用于群集/组部署（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.")
```

在脚本中，词典名称()包含在请求URL路径中。

在JSON负载中ignorecase包wholewords括所words需的encoding参数 (、 和)。添加device_type到URL查询字符串。

仅mode当ESA位于集群或组中时才需要此参数。在这种情况下，将mode=cluster附加到该URL查询字符串的方式与 device_type。

*credit 并*aba添加到负载，以启用信用卡号和ABA路由号的智能标识符，如屏幕截图所示：



该术语将Example Term 作为要匹配的词典项添加到负载中。

验证

当POST请求成功时，API日志包含类似于以下内容的条目：

```
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
```

预期响应：

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

如果字典已存在，并且POST请求再次运行，则API日志包含类似以下内容的条目：

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
```

预期响应：

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

相关信息

使用以下外部资源了解其他背景（本文档中的过程是自包含的）：

[使用REST API + Python添加字典](#)

本文档包括所需的身份验证方法和最低程序。将HTTP基本身份验证与报头Authorization: 基本<base64(username:password)>和首选端口6443上的HTTPS。

[REST API和思科ESA](#)

此外，此程序中使用的Python代码可以从Postman中提取。如果需要，请查看此视频教程：

[Postman — 如何生成Python脚本](#)

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。