

# Python 및 AsyncOS REST API를 사용하여 ESA 사전 구성

## 목차

---

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[전송 및 인증](#)

[배경 정보](#)

[구성](#)

[스크립트](#)

[다음을 확인합니다.](#)

[관련 정보](#)

---

## 소개

이 문서에서는 Python 스크립트를 사용하여 AsyncOS API(REST API)를 사용하여 Cisco ESA(Email Security Appliance)에 새 사전을 추가하는 방법에 대해 설명합니다.

## 사전 요구 사항

다음 전제 조건을 검토합니다.

- 파이썬 프로그래밍 언어
- Python requests 모듈
- JSON
- Cisco ESA(Email Security Appliance)

## 요구 사항

다음 소프트웨어 및 액세스 권한을 사용하여 스크립트를 실행합니다.

- AsyncOS 14.x 이상을 실행하는 Cisco ESA(Email Security Appliance)
- 파이썬 3
- Python requests 모듈

- AsyncOS API(REST API) 사용 권한이 있는 ESA 관리자 자격 증명
- 포트 6443의 HTTPS를 통해 ESA 관리 인터페이스에 액세스(권장)

## 사용되는 구성 요소

이 실습을 완료하는 데 다음 구성 요소가 사용되었습니다.

1) Cisco ESA - 15.0.0-104

2) 송고한 텍스트 편집기

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

## 전송 및 인증

- 전송: 사용 가능한 경우 AsyncOS API에 대해 포트 6443에서 HTTPS를 사용합니다. HTTP/6080은 비프로덕션 또는 랩 환경에서만 사용하십시오.
- 인증: 이 예에서는 HTTP 기본 인증을 사용합니다. 헤더 권한 부여를 제공합니다. 기본 <base64(사용자 이름:비밀번호)>. 스크립트에 실제 자격 증명을 붙여넣지 마십시오.

## 배경 정보

이 문서의 경우:

목표: Python 스크립트 및 AsyncOS API(REST API)를 사용하여 ESA에 사전을 추가합니다.

예상 결과: API에서 성공적인 응답을 반환하고 새 사전이 ESA 컨피그레이션에 나타납니다.

## 구성

스크립트를 실행하기 전에 ESA 관리 인터페이스에서 AsyncOS API(REST API) 서비스를 활성화합니다.

1. GUI(그래픽 사용자 인터페이스)에서 AsyncOS API HTTP 및 AsyncOS API HTTPS 설정을 포

- 합하는 네트워크 또는 인터페이스 페이지로 이동합니다.
- AsyncOS API HTTPS를 활성화하고 포트 6443을 확인합니다.
- (선택 사항) AsyncOS API HTTP를 활성화하고 실습용으로 포트 6080을 확인합니다.
- 저장을 클릭합니다.

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.

|                                                       |      |
|-------------------------------------------------------|------|
| <input checked="" type="checkbox"/> AsyncOS API HTTP  | 6080 |
| <input checked="" type="checkbox"/> AsyncOS API HTTPS | 6443 |

이미지 1: 관리 인터페이스에서 AsyncOS API HTTPS(포트 6443)가 활성화되었는지 확인합니다.

AsyncOS API 요청 트러블슈팅을 위한 API 로깅을 구성합니다.

- GUI(그래픽 사용자 인터페이스)에서 System Administration(시스템 관리) > Log Subscriptions(로그 서브스크립션)로 이동합니다.
- Add Log Subscription을 클릭합니다.
- Log Type 드롭다운 목록에서 API Logs를 선택합니다.
- 로그 이름과 파일 이름과 같은 나머지 필수 필드를 입력한 다음 Submit(제출)을 클릭합니다.

New Log Subscription

Log Subscription

Log Type: API Logs

Log Name: api  
(will be used to name the log directory)

File Name: api Append ☐ hostname ☐ serialnumber

Rollover by File Size: 10M Maximum  
(Add a trailing K or M to indicate size units)

Rollover by Time: None

Rate Limit: ☐ Maximum number of messages to be logged 60 Time range in seconds (1 to 60)

Log Level: ☐ Critical (The least detailed setting. Only errors are logged.) ☐ Warning (All errors and warnings created by the system.) ☒ Information (Captures the second-by-second operations of the system. Recommended.) ☐ Debug (More specific data are logged to help debug specific problems.) ☐ Trace (The most detailed setting, all information that can be is logged. Recommended for developers only.)

Retrieval Method: ☒ Manually download logs from ahsan-asa.com  
Logs are always available via HTTP(S) download. They are also available via SCP if SSH is enabled and FTP if it is enabled on any interface.

드롭다운 목록 API Logs에서 Log Type 선택하고 나머지 필수 세부사항을 입력합니다.



참고: GUI에서 로그를 구성할 수 있습니다.

## 스크립트

Python 요청 모듈에서 HTTP POST 메서드를 사용하여 디렉터리를 생성합니다.

POST 요청에 다음 매개변수를 사용합니다.

| 매개변수      | 위치         | 필수  | 참고                                                                                              |
|-----------|------------|-----|-------------------------------------------------------------------------------------------------|
| 장치_유형     | URL 쿼리 문자열 | 예   | esa로 설정합니다.                                                                                     |
| 이그노레케이스   | JSON 페이로드  | 예   | 0 = 대/소문자 구분, 1 = 대/소문자 구분 안 함                                                                  |
| 전두환       | JSON 페이로드  | 예   | 0 = 부분 문자열 일치, 1 = 전체 단어 일치.                                                                    |
| 인코딩       | JSON 페이로드  | 예   | 다른 인코딩이 필요하지 않은 경우 utf-8을 사용합니다.                                                                |
| 단어        | JSON 페이로드  | 예   | 항목 배열. 리터럴 용어에 ["용어"]를 사용합니다. 스마트 식별자의 경우 ESA에서 지원하는 ["*credit", 2, "prefix"]와 같은 튜플 형식을 사용합니다. |
| 모드로 들어갑니다 | URL 쿼리 문자열 | 아니요 | 클러스터링된/그룹 구축에만 필요합니다(예: 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.")

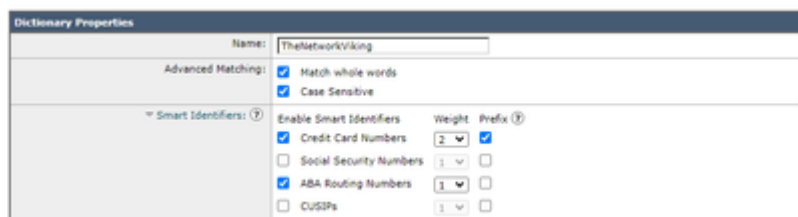
```

스크립트에서 사전 이름(  
)은 요청 URL 경로에 포함됩니다.

JSON 페이로드에 필요한 매개변수(ignorecase, wholewords, words, encoding)를 포함합니다. URL 쿼리 문자열device\_type에 추가합니다.

이 mode 매개변수는 ESA가 클러스터 또는 그룹에 있는 경우에만 필요합니다. 이 경우 mode=cluster를 URL 쿼리 문자열에 추가하는 방법은 다음과 같습니다 device\_type.

\*credit 그리고\*aba다음 스크린샷과 같이 페이로드에 추가되어 신용카드 번호 및 ABA 라우팅 번호에 대한 스마트 식별자를 활성화합니다.



용어Example Term 는 일치시킬 사전 항목으로 페이로드에 추가됩니다.

## 다음을 확인합니다.

POST 요청이 성공하면 API 로그에 다음과 유사한 항목이 포함됩니다.

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

예상 응답:

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

사전이 이미 있고 POST 요청이 다시 실행되는 경우 API 로그에 다음과 유사한 항목이 포함됩니다.

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

예상 응답:

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

## 관련 정보

추가 배경에 대해 다음 외부 리소스를 사용합니다(이 문서의 절차는 자체 포함).

[REST API + Python을 사용하여 사전 추가](#)

이 문서에는 필요한 인증 방법과 최소 절차가 포함되어 있습니다. 헤더 권한 부여와 함께 HTTP 기본 인증 사용: 기본 <base64(사용자 이름:비밀번호)>이며 포트 6443에서 HTTPS를 선호합니다.

[REST API 및 Cisco ESA](#)

또한 이 절차에서 사용되는 파이썬 코드는 Postman에서 추출할 수 있습니다. 필요한 경우 이 비디오 자습서를 검토합니다.

[Postman - Python 스크립트를 생성하는 방법](#)

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.