

# Konfigurieren des ESA-Wörterbuchs mithilfe der REST-API von Python und AsyncOS

## Inhalt

---

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Transport und Authentifizierung](#)

[Hintergrundinformationen](#)

[Konfigurieren](#)

[Skript](#)

[Überprüfung](#)

[Zugehörige Informationen](#)

---

## Einleitung

In diesem Dokument wird die Verwendung des Python-Skripts zum Hinzufügen eines neuen Wörterbuchs zu einer Cisco Email Security Appliance (ESA) mithilfe der AsyncOS-API (REST-API) beschrieben.

## Voraussetzungen

Überprüfen Sie folgende Voraussetzungen:

- Python-Programmiersprache
- Python-Anforderungsmodul
- JSON
- Cisco E-Mail Security Appliance (ESA)

## Anforderungen

Verwenden Sie die folgende Software und den folgenden Zugriff, um das Skript auszuführen:

- Cisco Email Security Appliance (ESA) mit AsyncOS 14.x oder höher
- Python 3

- Python-Anforderungsmodul
- ESA-Administratorberechtigungen mit der Berechtigung zur Verwendung der AsyncOS-API (REST-API)
- Zugriff auf die ESA-Verwaltungsschnittstelle über HTTPS auf Port 6443 (empfohlen)

## Verwendete Komponenten

Diese Komponenten wurden zur Durchführung dieser Übung verwendet:

1) Cisco ESA - 15.0.0-104

2) Sublime Text Editor

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

## Transport und Authentifizierung

- Verkehr: Verwenden Sie HTTPS auf Port 6443 für die AsyncOS-API, sofern verfügbar. Verwenden Sie HTTP/6080 nur in Nicht-Produktions- oder Laborumgebungen.
- Authentifizierung: In diesem Beispiel wird die HTTP-Standardauthentifizierung verwendet. Geben Sie die Header-Autorisierung an: Basic <base64(username:password)>. Fügen Sie keine echten Anmeldeinformationen in das Skript ein.

## Hintergrundinformationen

Für dieses Dokument:

Ziel: Fügen Sie der ESA mithilfe eines Python-Skripts und der AsyncOS-API (REST-API) ein Wörterbuch hinzu.

Erwartetes Ergebnis: Die API gibt eine erfolgreiche Antwort zurück, und das neue Wörterbuch wird in der ESA-Konfiguration angezeigt.

## Konfigurieren

Aktivieren Sie den Service AsyncOS API (REST API) auf der ESA-Verwaltungsschnittstelle, bevor Sie das Skript ausführen.

1. Navigieren Sie in der grafischen Benutzeroberfläche (GUI) zu der Seite für das Netzwerk oder die Schnittstelle, die die HTTP- und HTTPS-Einstellungen der AsyncOS-API enthält.
2. Aktivieren Sie HTTPS für die AsyncOS-API, und überprüfen Sie Port 6443.
3. (Optional) Aktivieren Sie AsyncOS API HTTP, und überprüfen Sie Port 6080 für die Laboranwendung.
4. Klicken Sie auf Speichern.

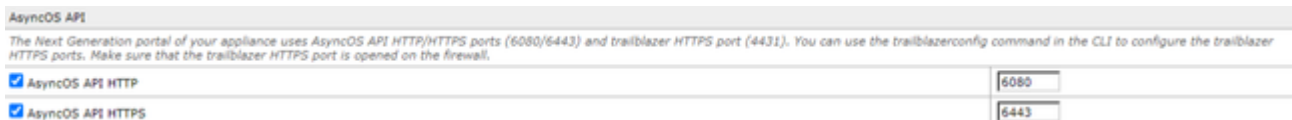
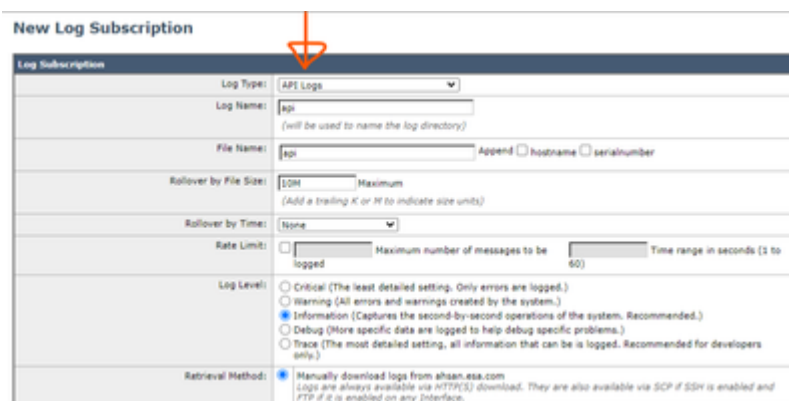


Image 1: Stellen Sie sicher, dass die asyncOS-API HTTPS (Port 6443) auf der Management-Schnittstelle aktiviert ist.

Konfigurieren der API-Protokollierung zur Fehlerbehebung bei AsyncOS-API-Anforderungen

1. Navigieren Sie in der grafischen Benutzeroberfläche zu Systemverwaltung > Protokollabonnements.
2. Klicken Sie auf Protokoll-Subscription hinzufügen.
3. Wählen Sie in der Dropdown-Liste Log Type (Protokolltyp) die Option API Logs (API-Protokolle) aus.
4. Geben Sie die übrigen erforderlichen Felder ein, z. B. den Protokollnamen und den Dateinamen, und klicken Sie dann auf Senden.



Wählen Sie API Logseine Option aus der Log Type Dropdown-Liste aus, und geben Sie die übrigen erforderlichen Details ein.



---

Anmerkung: Protokolle können über die GUI konfiguriert werden.

---

## Skript

Verwenden Sie die HTTP POST-Methode des Python-Anforderungsmoduls, um ein Wörterbuch zu erstellen.

Verwenden Sie diese Parameter für die POST-Anforderung:

| Parameter    | Location (Standort)     | Erforderlich | Hinweise                                                                                                                                                                                            |
|--------------|-------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Gerätetyp    | URL-Abfragezeichenfolge | Ja           | Auf "esa" eingestellt.                                                                                                                                                                              |
| ignorieren   | JSON-Nutzlast           | Ja           | 0 = Groß-/Kleinschreibung beachten, 1 = Groß-/Kleinschreibung wird nicht berücksichtigt.                                                                                                            |
| ganze Wörter | JSON-Nutzlast           | Ja           | 0 = Übereinstimmung mit Teilzeichenfolge, 1 = Übereinstimmung mit ganzen Wörtern.                                                                                                                   |
| Kodierung    | JSON-Nutzlast           | Ja           | Verwenden Sie utf-8, es sei denn, eine andere Kodierung ist erforderlich.                                                                                                                           |
| Wörter       | JSON-Nutzlast           | Ja           | Array von Einträgen. Verwenden Sie ["term"] für einen wörtlichen Begriff. Verwenden Sie für intelligente Bezeichner ein Tupelformat wie ["*credit", 2, "prefix"], das von der ESA unterstützt wird. |
| Modus        | URL-Abfragezeichenfolge | Nein         | Nur für Cluster-/Gruppenbereitstellungen erforderlich (z. B. mode=cluster).                                                                                                                         |

```
import base64
import requests

# ESA management IP address or FQDN
host = "<ESA_IP>"

# Dictionary name to create
dictionary_name = "<DICT_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 = "<USERNAME>"
password = "<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.")

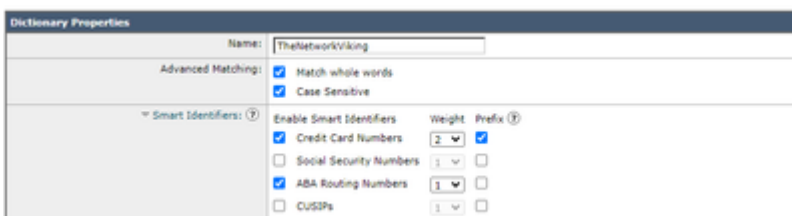
```

Im Skript ist der Wörterbuchname ( ) im Anforderungs-URL-Pfad enthalten.

Integrieren Sie die erforderlichen Parameter (`ignorecase`, `wholewords`, `words`, und `encoding`) in der JSON-Nutzlast. `device_type` Zur URL-Abfragezeichenfolge hinzufügen.

Der `mode` Parameter ist nur erforderlich, wenn sich die ESAs in einem Cluster oder einer Gruppe befinden. In diesem Fall fügen Sie `mode=cluster` auf die gleiche Weise an die URL-Abfragezeichenfolge an wie `device_type`.

\*`credit` und \*`aba` werden zur Nutzlast hinzugefügt, um die Smart Identifier für Kreditkartennummern und ABA-Weiterleitungsnummern zu aktivieren, wie im Screenshot gezeigt:



Der Begriff `Example Term` wird der Nutzlast als übereinstimmendes Wörterbuchelement hinzugefügt.

## Überprüfung

Wenn die POST-Anforderung erfolgreich ist, enthält das API-Protokoll einen Eintrag wie diesen:

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
```

Erwartete Antwort:

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

Wenn das Dictionary bereits vorhanden ist und die POST-Anforderung erneut ausgeführt wird, enthält das API-Protokoll einen Eintrag ähnlich dem folgenden:

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
```

Erwartete Antwort:

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

## Zugehörige Informationen

Verwenden Sie die folgenden externen Ressourcen für zusätzlichen Hintergrund (die Vorgehensweise in diesem Dokument ist eigenständig):

[Hinzufügen von Wörterbüchern mit REST-API + Python](#)

Dieses Dokument enthält die erforderliche Authentifizierungsmethode und die minimale Vorgehensweise. HTTP Basic authentication mit dem Header `Authorization: Basic <base64(username:password)>` und bevorzugen HTTPS auf Port 6443.

[REST-API und Cisco ESA](#)

Zusätzlich kann der in dieser Prozedur verwendete Python-Code aus Postman extrahiert werden.

Sehen Sie sich ggf. dieses Video-Tutorial an:

[Postman - Wie man ein Python-Skript generiert](#)

### Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.