

Configuración del diccionario ESA mediante las API REST de Python y AsyncOS

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Transporte y autenticación](#)

[Antecedentes](#)

[Configurar](#)

[Guión](#)

[Verificación](#)

[Información Relacionada](#)

Introducción

Este documento describe cómo utilizar el script Python para agregar un nuevo diccionario a un dispositivo de seguridad Cisco Email Security Appliance (ESA) mediante la API AsyncOS (API REST).

Prerequisites

Revise estos requisitos previos:

- Lenguaje de programación Python
- módulo de solicitudes Python
- JSON
- Dispositivo de seguridad Cisco Email Security Appliance (ESA)

Requirements

Utilice el software y el acceso siguientes para ejecutar la secuencia de comandos:

- Cisco Email Security Appliance (ESA) con AsyncOS 14.x o posterior
- Python 3

- módulo de solicitudes Python
- Credenciales de administrador de ESA con permiso para utilizar la API AsyncOS (API REST)
- Acceso a la interfaz de administración ESA a través de HTTPS en el puerto 6443 (recomendado)

Componentes Utilizados

Estos componentes se utilizaron para completar este laboratorio:

1) Cisco ESA: 15.0.0-104

2) Editor de texto sublime

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Transporte y autenticación

- Transporte: Utilice HTTPS en el puerto 6443 para la API AsyncOS cuando esté disponible. Utilice HTTP/6080 solo en entornos de laboratorio o sin producción.
- Autenticación: Este ejemplo utiliza la autenticación básica de HTTP. Proporcione el encabezado `Authorization: Básico <base64(username:password)>`. No pegue credenciales reales en el script.

Antecedentes

Para este documento:

Objetivo: Agregue un diccionario al ESA mediante un script Python y la API AsyncOS (API REST).

Resultado esperado: La API devuelve una respuesta correcta y el nuevo diccionario aparece en la configuración ESA.

Configurar

Habilite el servicio API AsyncOS (API REST) en la interfaz de administración ESA antes de ejecutar la secuencia de comandos.

1. En la interfaz gráfica de usuario (GUI), navegue hasta la página network o interface que contiene los parámetros de AsyncOS API HTTP y AsyncOS API HTTPS.
2. Habilite AsyncOS API HTTPS y verifique el puerto 6443.
3. (Opcional) Habilite AsyncOS API HTTP y verifique el puerto 6080 para uso de laboratorio.
4. Click Save.

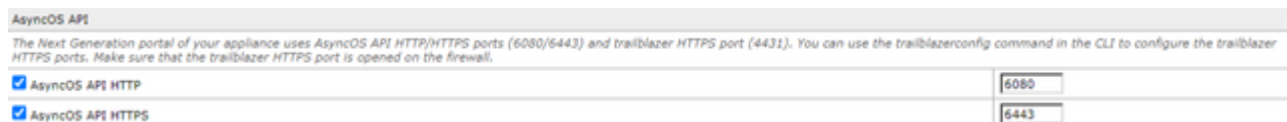
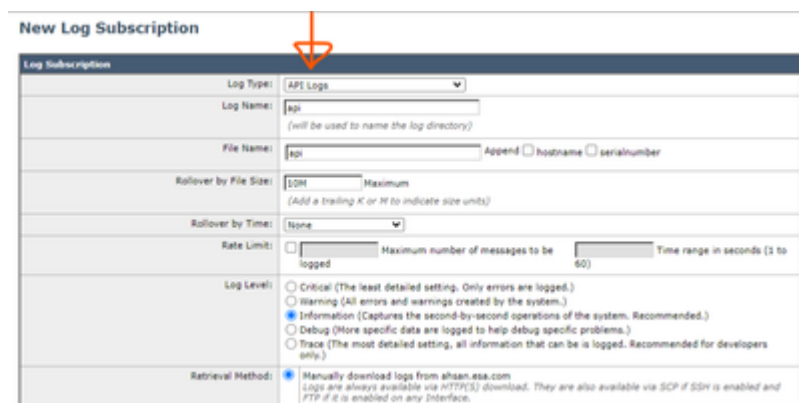


Imagen 1: Verifique que AsyncOS API HTTPS (puerto 6443) esté habilitado en la interfaz de administración.

Configure el registro de API para resolver problemas de solicitudes de API AsyncOS.

1. En la Interfaz gráfica de usuario (GUI), navegue hasta Administración del sistema > Suscripciones de registro.
2. Haga clic en Add Log Subscription.
3. En la lista desplegable Tipo de registro, seleccione Registros de API.
4. Introduzca los campos obligatorios restantes, como el nombre del registro y del archivo, y haga clic en Enviar.



Selecciónelo API Logs en la Log Type lista desplegable e introduzca los detalles restantes necesarios.



Nota: Los registros se pueden configurar desde la GUI.

Guión

Utilice el método HTTP POST del módulo de solicitudes de Python para crear un diccionario.

Utilice estos parámetros para la solicitud POST:

Parámetro	Ubicación	Necesario	Notas
device_type	cadena de consulta de URL	Yes	Establezca en esa.
caso ignorado	carga JSON	Yes	0 = distingue mayúsculas de minúsculas, 1 = no distingue mayúsculas de minúsculas.
palabras completas	carga JSON	Yes	0 = coincidencia de subcadena, 1 = coincidencia de palabra completa.
codificación	carga JSON	Yes	Utilice utf-8 a menos que se requiera una codificación diferente.
palabras	carga JSON	Yes	Matriz de entradas. Utilice ["término"] para un término literal. Para los identificadores inteligentes, utilice un formato de tupla como ["*crédito", 2, "prefijo"], tal como lo admite el ESA.
modo	cadena de consulta de URL	No	Necesario sólo para implementaciones en clúster/grupos (por ejemplo, mode=cluster).

```
import base64
import requests

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

# Dictionary name to create
dictionary_name = "<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 = "<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.")

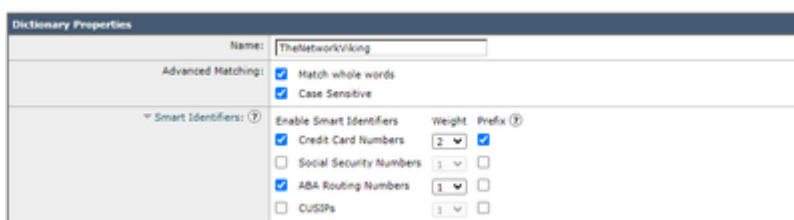
```

En la secuencia de comandos, el nombre del diccionario () se incluye en la ruta de la URL de la solicitud.

Incluya los parámetros necesarios (`ignorecase`, `wholewords`, `words`, y `encoding`) en la carga útil JSON. Agregue `device_type` a la cadena de consulta de URL.

El `mode` parámetro sólo es necesario cuando los ESA están en un clúster o grupo. En ese caso, anexe `mode=cluster` a la cadena de consulta de URL de la misma manera que `device_type`.

*`credit` y `se*` se agregan a la carga útil para habilitar los identificadores inteligentes para los números de tarjeta de crédito y los números de routing ABA, como se muestra en la captura de pantalla:



El término `Example Term` se agrega a la carga útil como un elemento del diccionario que debe coincidir.

Verificación

Cuando la solicitud `POST` se realiza correctamente, el registro de la API contiene una entrada similar a la siguiente:

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

Respuesta esperada:

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

Si el diccionario ya existe y la solicitud POST se ejecuta nuevamente, el registro de la API contiene una entrada similar a esta:

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

Respuesta esperada:

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

Información Relacionada

Utilice los siguientes recursos externos para obtener información adicional (el procedimiento de este documento es independiente):

[Adición de diccionarios mediante API REST + Python](#)

Este documento incluye el método de autenticación requerido y el procedimiento mínimo. Utilice la autenticación básica de HTTP con el encabezado `Authorization: Básico`
<base64(username:password)> y prefiera HTTPS en el puerto 6443.

[API REST y Cisco ESA](#)

Además, el código Python utilizado en este procedimiento se puede extraer de Postman. Revise este tutorial de vídeo si es necesario:

[Postman - Cómo generar un script Python](#)

Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).