

Configurar o Dicionário ESA usando Python e API REST AsyncOS

Contents

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Transporte e autenticação](#)

[Informações de Apoio](#)

[Configurar](#)

[Script](#)

[Verificar](#)

[Informações Relacionadas](#)

Introdução

Este documento descreve como usar o script Python para adicionar um novo dicionário a um Cisco Email Security Appliance (ESA) usando a API AsyncOS (REST API).

Pré-requisitos

Revise estes pré-requisitos:

- Linguagem de programação Python
- Módulo de requisições Python
- JSON
- Dispositivo de segurança de e-mail (ESA) da Cisco

Requisitos

Use o seguinte software e acesso para executar o script:

- Cisco Email Security Appliance (ESA) com AsyncOS 14.x ou posterior
- Python 3
- Módulo de requisições Python

- Credenciais de administrador do ESA com permissão para usar a API AsyncOS (REST API)
- Acesso à interface de gerenciamento do ESA através de HTTPS na porta 6443 (recomendado)

Componentes Utilizados

Estes componentes foram usados para concluir este laboratório:

1) Cisco ESA - 15.0.0-104

2) Editor de Texto Sublime

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Transporte e autenticação

- Transporte: Use HTTPS na porta 6443 para a API do AsyncOS quando disponível. Use HTTP/6080 somente em ambientes de laboratório ou que não sejam de produção.
- Autenticação: Este exemplo usa a autenticação básica HTTP. Forneça a autorização do cabeçalho: Básico <base64(username:password)>. Não cole credenciais reais no script.

Informações de Apoio

Para este documento:

Meta: Adicione um dicionário ao ESA usando um script Python e a API AsyncOS (REST API).

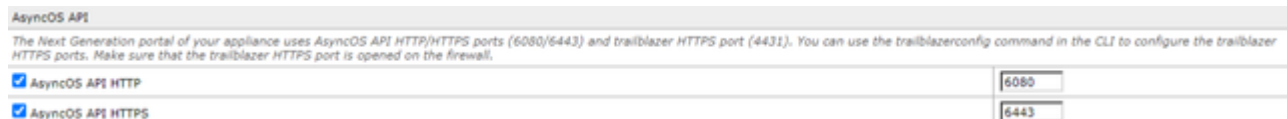
Resultado esperado: A API retorna uma resposta com êxito e o novo dicionário aparece na configuração do ESA.

Configurar

Ative o serviço AsyncOS API (REST API) na interface de gerenciamento ESA antes de executar o

script.

1. Na Interface gráfica do usuário (GUI), navegue até a página network ou interface que contém as configurações de AsyncOS API HTTP e AsyncOS API HTTPS.
2. Habilite AsyncOS API HTTPS e verifique a porta 6443.
3. (Opcional) Habilite AsyncOS API HTTP e verifique a porta 6080 para uso em laboratório.
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.

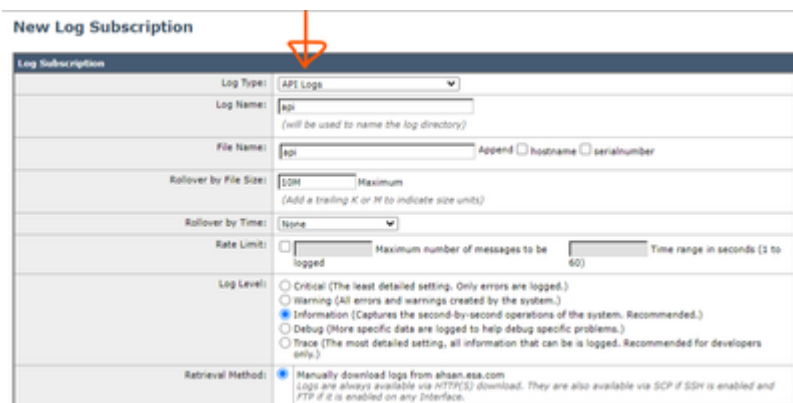
☒ AsyncOS API HTTP

☒ AsyncOS API HTTPS

Imagem 1: Verifique se AsyncOS API HTTPS (porta 6443) está Habilitado na interface de Gerenciamento.

Configure o log de API para solucionar problemas de solicitações de API do AsyncOS.

1. Na Interface gráfica do usuário (GUI), navegue para Administração do sistema > Inscrições de registro.
2. Clique em Add Log Subscription.
3. Na lista suspensa Tipo de log, selecione Logs de API.
4. Insira os campos obrigatórios restantes, como o nome do log e o nome do arquivo, e clique em Enviar.



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: **1GB** Maximum
(Add a trailing K or M to indicate size units)

Rollover by Time: **None**

Rate Limit: ☐ Maximum number of messages to be logged 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.esa.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.

Selecione API Logs na lista suspensa Log Type e insira os demais detalhes obrigatórios.



Note: Os logs podem ser configurados na GUI.

Script

Use o método HTTP POST do módulo Python `requests` para criar um dicionário.

Use estes parâmetros para a solicitação POST:

Parâmetro	Local	Necessário	Notas
<code>device_type</code>	Cadeia de caracteres de consulta URL	Yes	Defina como ESA.
<code>ignorecase</code>	payload JSON	Yes	0 = diferencia maiúsculas de minúsculas, 1 = não diferencia maiúsculas de minúsculas.
<code>palavras inteiras</code>	payload JSON	Yes	0 = correspondência de substring, 1 = correspondência de palavra inteira.
<code>codificação</code>	payload JSON	Yes	Use <code>utf-8</code> a menos que uma codificação diferente seja necessária.
<code>palavras</code>	payload JSON	Yes	Matriz de entradas. Use <code>["term"]</code> para um termo literal. Para identificadores inteligentes, use um formato de tupla como <code>["*credit", 2, "prefix"]</code> , conforme suportado pelo ESA.
<code>modo</code>	Cadeia de caracteres de consulta URL	No	Necessário apenas para implantações em cluster/grupo (por exemplo, <code>mode=cluster</code>).

```
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.")

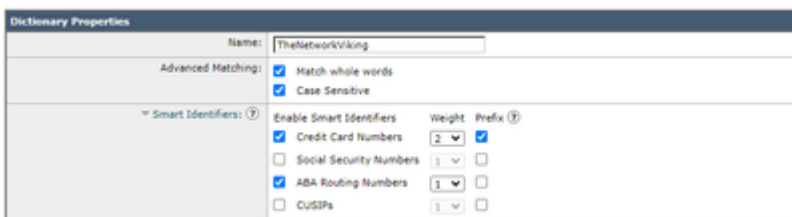
```

No script, o nome do dicionário () é incluído no caminho da URL de solicitação.

Inclua os parâmetros necessários (`ignorecase`, `wholewords`, `words`, e `encoding`) na carga JSON. Adicionar `device_type` à string de consulta da URL.

O parâmetro `mode` é necessário apenas quando os ESAs estão em um cluster ou grupo. Nesse caso, anexe `mode=cluster` à string de consulta da URL da mesma forma que `device_type`.

*`credit` e *`aba` são adicionados ao payload para ativar os identificadores inteligentes para números de cartão de crédito e números de roteamento ABA, como mostrado na captura de tela:



O termo `Example Term` é adicionado à carga como um item de dicionário para correspondência.

Verificar

Quando a solicitação POST é bem-sucedida, o log de API contém uma entrada semelhante a esta:

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

Resposta esperada:

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

Se o dicionário já existir e a solicitação POST for executada novamente, o log de API conterá uma entrada semelhante 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

Resposta esperada:

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

Informações Relacionadas

Use os seguintes recursos externos para obter informações adicionais de segundo plano (o procedimento neste documento é independente):

[Adicionando dicionários usando REST API + Python](#)

Este documento inclui o método de autenticação necessário e o procedimento mínimo. Usar a autenticação Básica HTTP com a Autorização do cabeçalho: Básico <base64(username:password)> e prefira HTTPS na porta 6443.

[API REST e Cisco ESA](#)

Além disso, o código Python usado neste procedimento pode ser extraído do Postman. Revise este tutorial em vídeo se necessário:

[Postman - Como gerar um script Python](#)

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.