

Configurazione del dizionario ESA con l'API REST Python e AsyncOS

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Trasporto e autenticazione](#)

[Premesse](#)

[Configurazione](#)

[Script](#)

[Verifica](#)

[Informazioni correlate](#)

Introduzione

In questo documento viene descritto come usare gli script Python per aggiungere un nuovo dizionario a Cisco Email Security Appliance (ESA) usando l'API AsyncOS (REST API).

Prerequisiti

Esaminare i prerequisiti seguenti:

- Linguaggio di programmazione Python
- Modulo `richieste` Python
- JSON
- Cisco Email Security Appliance (ESA)

Requisiti

Utilizzare il software e l'accesso seguenti per eseguire lo script:

- Cisco Email Security Appliance (ESA) con AsyncOS 14.x o versioni successive
- Python 3
- Modulo `richieste` Python

- Credenziali amministratore ESA con autorizzazione a utilizzare l'API AsyncOS (REST API)
- Accesso all'interfaccia di gestione ESA su HTTPS sulla porta 6443 (consigliato)

Componenti usati

I seguenti componenti sono stati utilizzati per completare questa esercitazione:

1) Cisco ESA - 15.0.0-104

2) Editor di testo Sublime

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Trasporto e autenticazione

- Trasporto: Utilizzare HTTPS sulla porta 6443 per l'API AsyncOS, quando disponibile. Utilizzare HTTP/6080 solo in ambienti di laboratorio o non di produzione.
- Autenticazione: In questo esempio viene utilizzata l'autenticazione di base HTTP. Specificare l'intestazione `Authorization: Basic <base64(username:password)>`. Non incollare credenziali reali nello script.

Premesse

Per questo documento:

Obiettivo: Aggiungere un dizionario all'ESA utilizzando uno script Python e l'API AsyncOS (REST API).

Risultato previsto: L'API restituisce una risposta corretta e il nuovo dizionario viene visualizzato nella configurazione ESA.

Configurazione

Abilitare il servizio API AsyncOS (REST API) sull'interfaccia di gestione ESA prima di eseguire lo script.

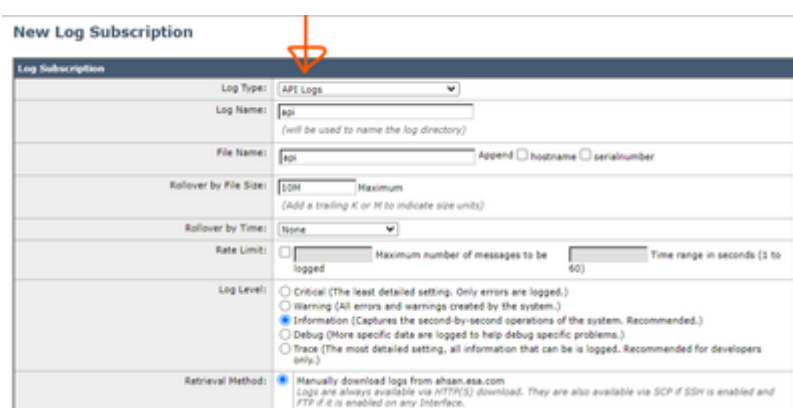
1. Nell'interfaccia utente grafica (GUI), passare alla rete o alla pagina interfaccia che contiene le impostazioni HTTP dell'API AsyncOS e HTTPS dell'API AsyncOS.
2. Abilitare AsyncOS API HTTPS e verificare la porta 6443.
3. (Facoltativo) Abilitare AsyncOS API HTTP e verificare la porta 6080 per l'utilizzo in laboratorio.
4. Fare clic su Save (Salva).



Immagine 1: Verificare che l'API HTTPS di AsyncOS (porta 6443) sia abilitata sull'interfaccia di gestione.

Configurare la registrazione API per la risoluzione dei problemi relativi alle richieste API AsyncOS.

1. Nell'interfaccia utente grafica (GUI), selezionare Amministrazione sistema > Sottoscrizioni registro.
2. Fare clic su Aggiungi sottoscrizione registro.
3. Dall'elenco a discesa Tipo di log, selezionare Log API.
4. Immettere i campi obbligatori rimanenti, ad esempio il nome del registro e il nome del file, quindi fare clic su Invia.



Effettuare una selezione API Logs dall'elenco a discesa Log Type e immettere i dettagli obbligatori rimanenti.



Nota: I log possono essere configurati dalla GUI.

Script

Utilizzare il metodo HTTP POST dal modulo delle richieste Python per creare un dizionario.

Utilizzare i seguenti parametri per la richiesta POST:

Parametro	Posizione	Obbligatorio	Note
tipo_periferica	Stringa di query URL	Sì	Impostare su esa.
ignorecase	Payload JSON	Sì	0 = con distinzione tra maiuscole e minuscole, 1 = senza distinzione tra maiuscole e minuscole.
parole intere	Payload JSON	Sì	0 = corrispondenza sottostringa, 1 = corrispondenza parola intera.
codifica	Payload JSON	Sì	Utilizzare utf-8 a meno che non sia necessaria una codifica diversa.
parole	Payload JSON	Sì	Matrice di voci. Utilizzare ["termine"] per un termine letterale. Per gli identificatori intelligenti, utilizzare un formato tupla come ["*credito", 2, "prefisso"] supportato dall'ESA.
modalità	Stringa di query URL	No	Richiesto solo per distribuzioni cluster/gruppi (ad esempio, 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.")

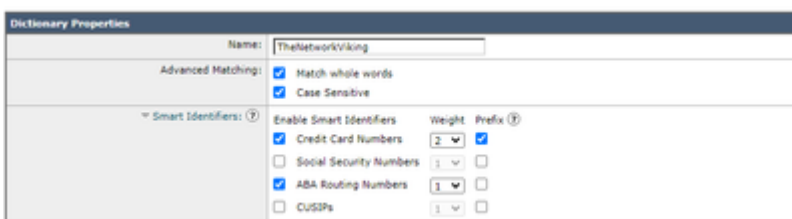
```

Nello script, il nome del dizionario () è incluso nel percorso URL della richiesta.

Includere i parametri obbligatori (**ignorecase**, **wholewords**, **words**, e **encoding**) nel payload JSON. Aggiungere **device_type** alla stringa di query dell'URL.

Il **mode** parametro è obbligatorio solo quando le ESA si trovano in un cluster o in un gruppo. In questo caso, aggiungere **mode=cluster** alla stringa di query dell'URL nello stesso modo in cui **device_type**.

***credit** e ***aba** vengono aggiunti al payload per abilitare gli identificatori intelligenti per i numeri di carta di credito e i numeri di routing ABA, come mostrato nello screenshot:



Il termine **Example Term** viene aggiunto al payload come elemento di dizionario corrispondente.

Verifica

Quando la richiesta **POST** ha esito positivo, il log API contiene una voce simile alla seguente:

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

Risposta prevista:

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

Se il dizionario esiste già e la richiesta POST viene eseguita nuovamente, il log API contiene una voce simile alla seguente:

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

Risposta prevista:

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

Informazioni correlate

Utilizzare le seguenti risorse esterne per ulteriori informazioni di base (la procedura illustrata in questo documento è autonoma):

[Aggiunta di dizionari tramite l'API REST + Python](#)

Questo documento include il metodo di autenticazione richiesto e la procedura minima. Utilizzare l'autenticazione di base HTTP con l'intestazione `Authorization: Basic <base64(username:password)>` e preferire HTTPS sulla porta 6443.

[API REST e Cisco ESA](#)

Inoltre, il codice Python utilizzato in questa procedura può essere estratto da Postman. Se necessario, rivedere questa esercitazione video:

[Postman - Come generare uno script Python](#)

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).