Automatize o isolamento de início/parada em vários endpoints

Contents

Introdução

Este documento descreve como automatizar o isolamento de parada/início em vários endpoints usando a API para Cisco Secure Endpoint.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Endpoint seguro da Cisco
- Console Cisco Secure Endpoint
- API do Cisco Secure Endpoint
- Python

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software:

- Endpoint Cisco Secure 8.4.0.30201
- Endpoint para ambiente python de host
- Python 3. 11. 7

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.

Informações de Apoio

- Use uma solicitação PUT para iniciar o isolamento.
- Uma solicitação DELETE é usada para interromper o isolamento.
- Verifique a documentação da API para obter mais informações.

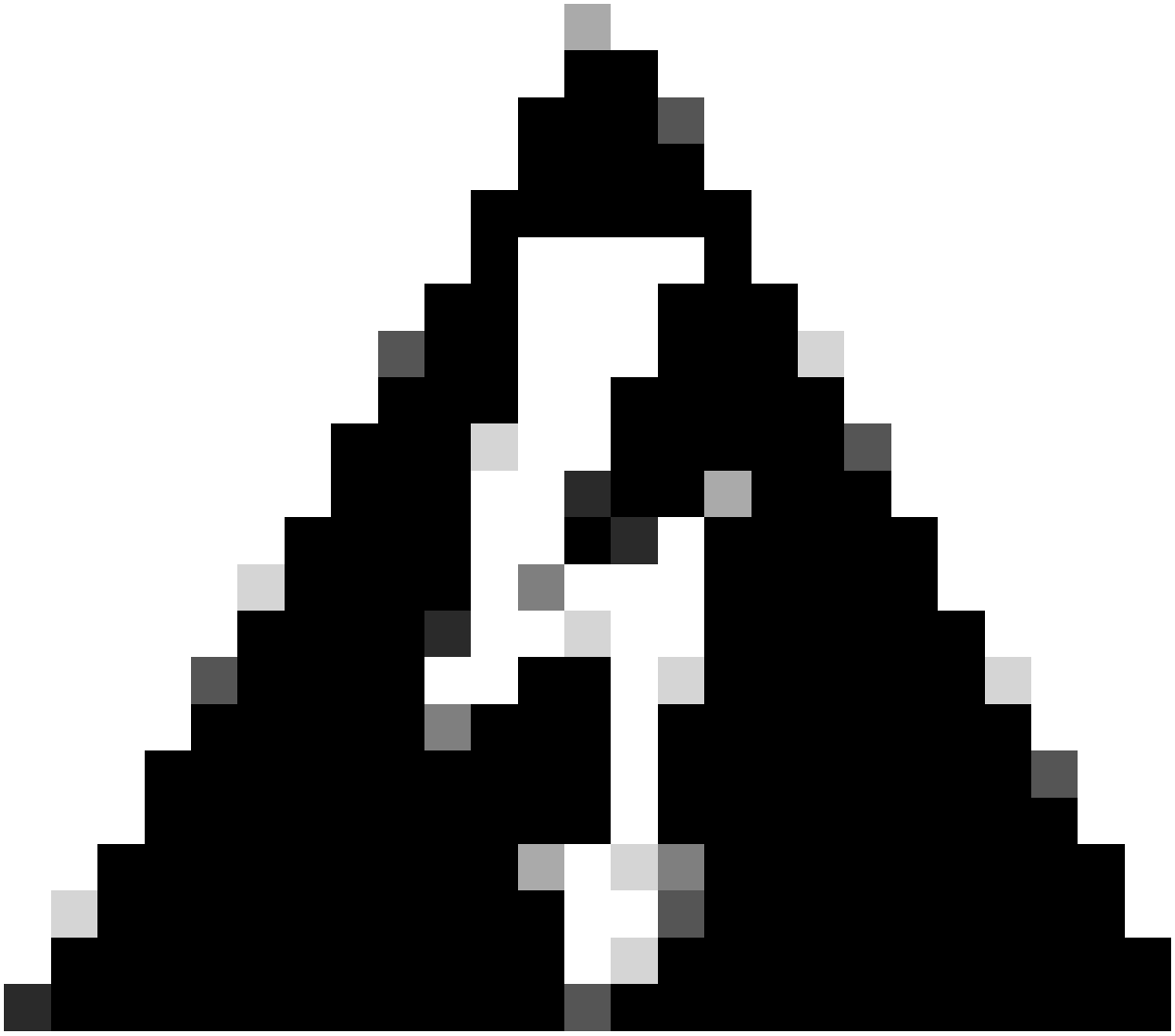
Problema

O Cisco Secure Endpoint permite iniciar/parar o isolamento em uma máquina de cada vez. No entanto, durante um incidente de segurança, muitas vezes é necessário executar essas operações em vários endpoints simultaneamente para conter as possíveis ameaças de forma eficaz. Automatizar o processo de isolamento de início/parada para endpoints em massa usando a API pode melhorar significativamente a eficiência da resposta a incidentes e reduzir o risco geral para a rede.

Solução

- O script Python fornecido neste artigo pode ser usado para iniciar/terminar o isolamento em vários endpoints em sua organização usando credenciais de API de endpoint seguro.
- Para gerar as credenciais da API do AMP, consulte Visão geral da API do Cisco AMP para endpoints
- Para usar o script fornecido, você precisa instalarpythonem seus endpoints.
- Após a instalação do python, instale o módulo requests

```
pip install requests
```



Aviso: o script é fornecido apenas para fins ilustrativos e tem como objetivo demonstrar como automatizar o recurso de isolamento de endpoint usando a API. O Cisco Technical Assistance Center (TAC) não está envolvido na solução de problemas relacionados a este script. Os usuários devem ter cuidado e testar completamente o script em um ambiente seguro antes de implantá-lo em uma configuração de produção.

Script

Você pode usar o script fornecido para iniciar o isolamento em vários endpoints em sua empresa:

```
import requests

def read_config(file_path):
    """
    Reads the configuration file to get the API base URL, client ID, and API key.
    """
    config = {}
    try:
```

```
        with open(file_path, 'r') as file:
            for line in file:
                # Split each line into key and value based on '='
                key, value = line.strip().split('=')
                config[key] = value
    except FileNotFoundError:
        print(f"Error: Configuration file '{file_path}' not found.")
        exit(1)  # Exit the script if the file is not found
    except ValueError:
        print(f"Error: Configuration file '{file_path}' is incorrectly formatted.")
        exit(1)  # Exit the script if the file format is invalid
    return config

def read_guids(file_path):
    """
    Reads the file containing GUIDs for endpoints to be isolated.
    """
    try:
        with open(file_path, 'r') as file:
            # Read each line, strip whitespace, and ignore empty lines
            return [line.strip() for line in file if line.strip()]
    except FileNotFoundError:
        print(f"Error: GUIDs file '{file_path}' not found.")
        exit(1)  # Exit the script if the file is not found
    except Exception as e:
        print(f"Error: An unexpected error occurred while reading the GUIDs file: {e}")
        exit(1)  # Exit the script if an unexpected error occurs

def isolate_endpoint(base_url, client_id, api_key, connector_guid):
    """
    Sends a PUT request to isolate an endpoint identified by the connector GUID.
    Args:
        base_url (str): The base URL for the API.
        client_id (str): The API client ID for authentication.
        api_key (str): The API key for authentication.
        connector_guid (str): The GUID of the connector to be isolated.
    """
    url = f"{base_url}/{connector_guid}/isolation"
    try:
        # Send PUT request with authentication
        response = requests.put(url, auth=(client_id, api_key))
        response.raise_for_status()  # Raise an HTTPError for bad responses (4xx and 5xx)

        if response.status_code == 200:
            print(f"Successfully isolated endpoint: {connector_guid}")
        else:
            print(f"Failed to isolate endpoint: {connector_guid}. Status Code: {response.status_code},
    except requests.RequestException as e:
        print(f"Error: An error occurred while isolating the endpoint '{connector_guid}': {e}")

if __name__ == "__main__":
    # Read configuration values from the config file
    config = read_config('config.txt')

    # Read list of GUIDs from the GUIDs file
    connector_guids = read_guids('guids.txt')

    # Extract configuration values
    base_url = config.get('BASE_URL')
    api_client_id = config.get('API_CLIENT_ID')
    api_key = config.get('API_KEY')
```

```
    # Check if all required configuration values are present
    if not base_url or not api_client_id or not api_key:
        print("Error: Missing required configuration values.")
        exit(1)  # Exit the script if any configuration values are missing

    # Process each GUID by isolating the endpoint
    for guid in connector_guids:
        isolate_endpoint(base_url, api_client_id, api_key, guid)
```

Instruções

- Para gerar as credenciais da API do AMP, consulte Visão geral da API do Cisco AMP para endpoints
- Use o BASE_URL mencionado para a sua região:

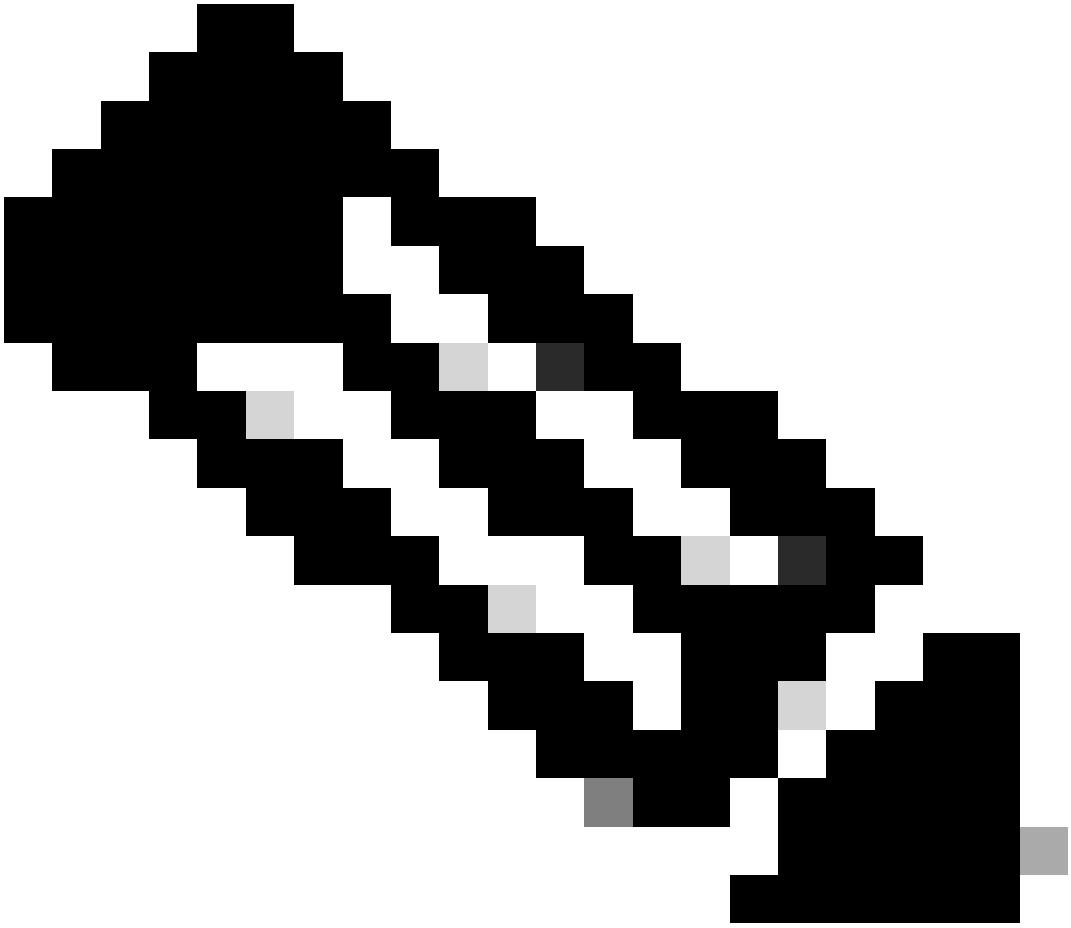
```
NAM - https://api.amp.cisco.com/v1/computers/
EU - https://api.eu.amp.cisco.com/v1/computers/
APJC - https://api.apjc.amp.cisco.com/v1/computers/
```

- Crie um arquivo config.txt no mesmo diretório do script com o conteúdo mencionado. Exemplo do arquivo config.txt:

```
BASE_URL=https://api.apjc.amp.cisco.com/v1/computers/
API_CLIENT_ID=xxxxxxxxxxxxxxxxxxxx
API_KEY=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

- Crie um arquivo guids.txt no mesmo diretório do script com a lista de GUIDs de conector, um por linha. Adicione o máximo de GUIDs necessário. Exemplo do arquivo guids.txt:

```
abXXXXXXXXXXXXcd-XefX-XghX-X12X-XXXXXX567XXXXXXX
yzXXXXXXXXXXXXlm-XprX-XmnX-X34X-XXXXXX618XXXXXXX
```



Observação: você pode coletar os GUIDs de seus endpoints por meio do API GET /v1/computers ou do Cisco Secure Endpoint Console navegando para Management > Computers, expandindo a entrada para um endpoint específico e copiando o GUID do conector.

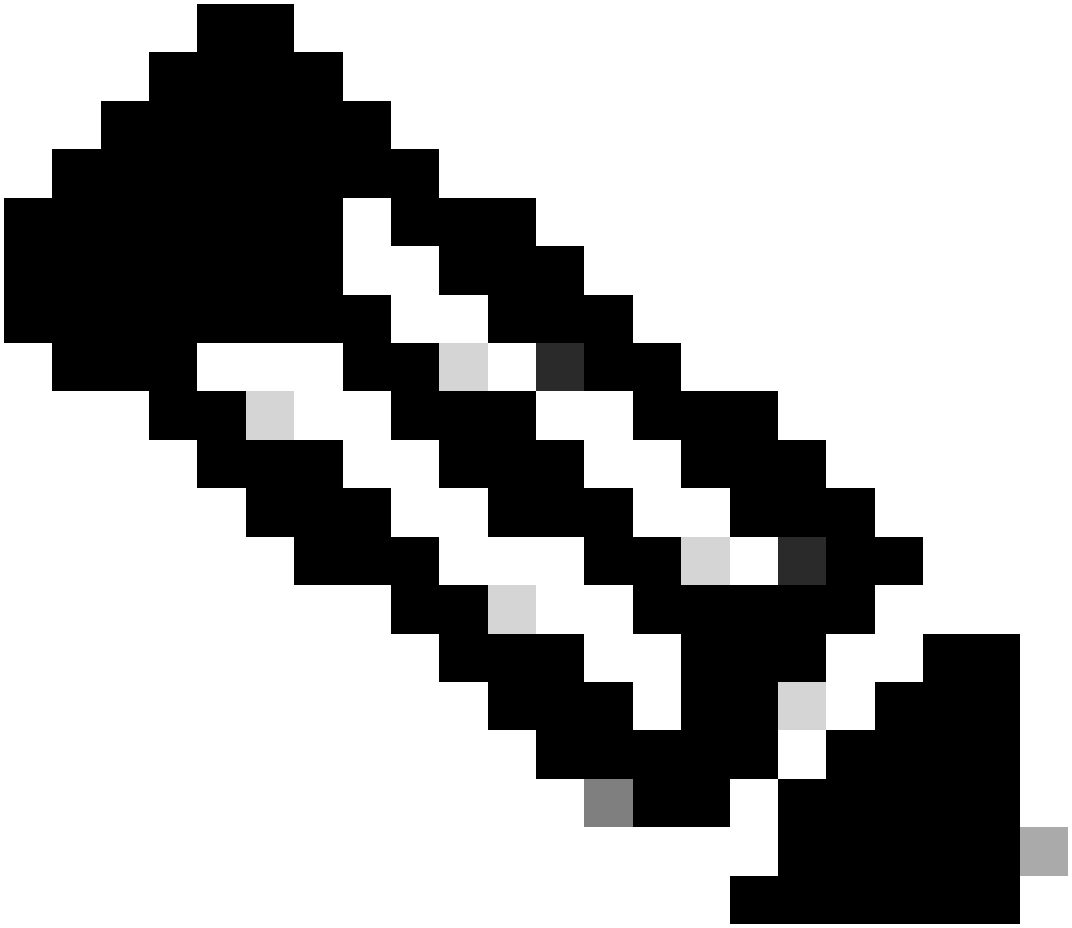
- Abra um Terminal ou um Prompt de Comando. Navegue até o diretório onde o start_isolamento_script.py está localizado.
- Execute o script executando o comando mencionado:

```
python start_isolation_script.py
```

Verificar

- O script tenta isolar cada ponto final especificado no arquivo guids.txt.

- Verifique o terminal ou o prompt de comando para obter mensagens de êxito ou de erro para cada endpoint.



Observação: o script start_isolamento.py anexado pode ser usado para iniciar o isolamento em pontos finais, enquanto stop_isolamento.py é projetado para parar o isolamento em pontos finais. Todas as instruções para executar o script permanecem as mesmas.

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.