Automatisierung der Start-/Stopp-Isolierung auf mehreren Endgeräten

Inhalt

Einleitung

In diesem Dokument wird beschrieben, wie die Stopp/Start-Isolierung auf mehreren Endgeräten mithilfe der API für Cisco Secure Endpoint automatisiert werden kann.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Sichere Endgeräte von Cisco
- Cisco Secure Endpoint Console
- Cisco Secure Endpoint-API
- Python

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Softwareversionen:

- Cisco Secure Endpoint 8.4.0.30201
- Endpunkt zu Host-Python-Umgebung
- Python 3.11.7

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.

Hintergrundinformationen

- Sie verwenden eine PUT-Anforderung, um die Isolierung zu initiieren.
- Eine DELETE-Anforderung wird verwendet, um die Isolation zu beenden.
- Weitere Informationen finden Sie in der API-Dokumentation.

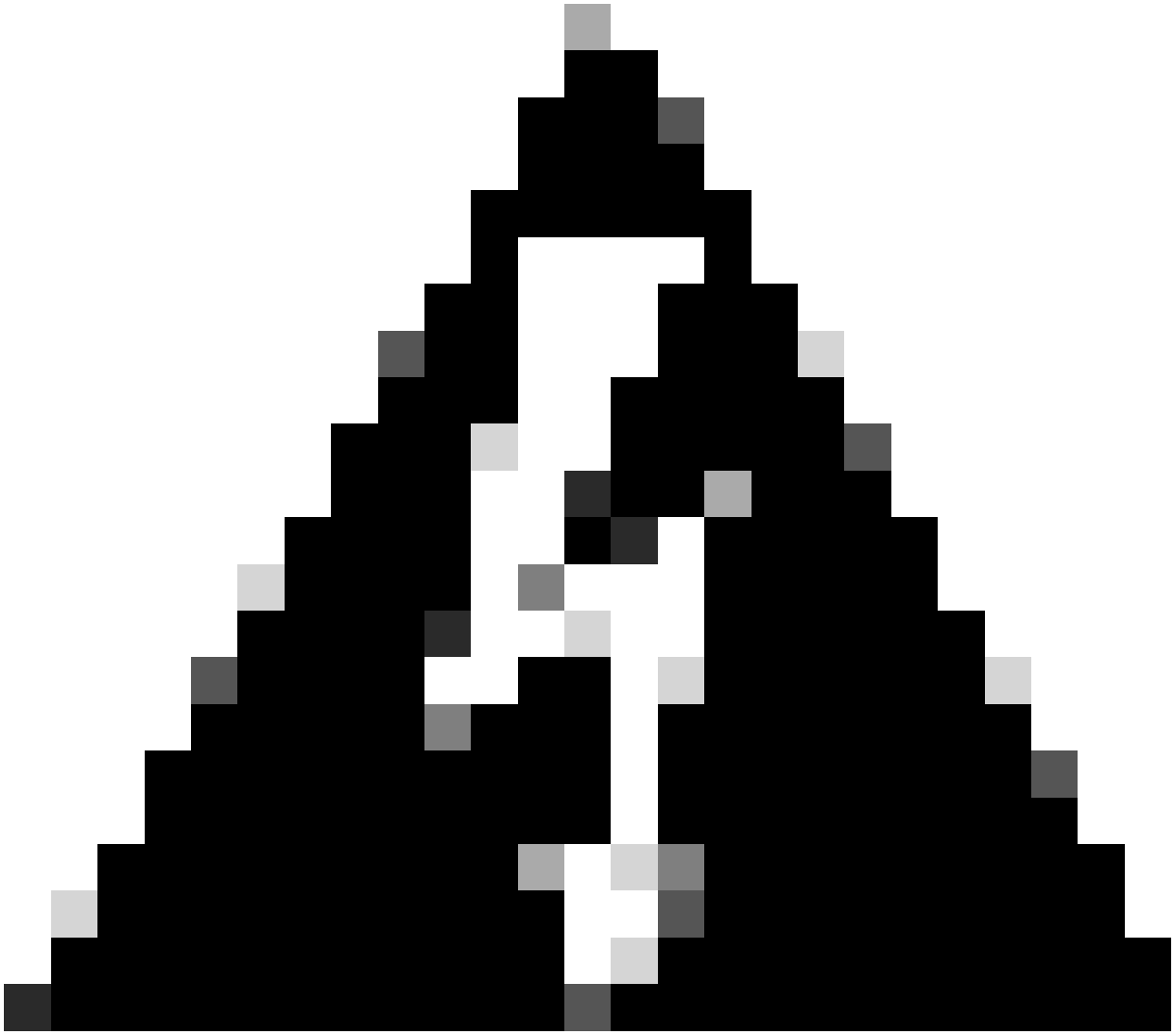
Problem

Cisco Secure Endpoint ermöglicht das Starten/Stoppen der Isolation auf einem Gerät nach dem anderen. Bei einem Sicherheitsvorfall müssen diese Vorgänge jedoch häufig auf mehreren Endpunkten gleichzeitig ausgeführt werden, um potenzielle Bedrohungen effektiv einzudämmen. Die Automatisierung des Start/Stopp-Isolationsprozesses für große Mengen von Endpunkten mithilfe der API kann die Effizienz der Reaktion auf Vorfälle erheblich steigern und das Gesamtrisiko für das Netzwerk verringern.

Lösung

- Das in diesem Artikel bereitgestellte Python-Skript kann verwendet werden, um die Isolation auf mehreren Endpunkten in Ihrer Organisation mithilfe von API-Anmeldeinformationen für sichere Endpunkte zu initiieren/zu beenden.
- Informationen zum Generieren der AMP-API-Anmeldeinformationen finden Sie unter Übersicht über die Cisco AMP für Endgeräte-API
- Um das bereitgestellte Skript zu verwenden, müssen Sie pythonauf Ihren Endgeräten installieren.
- Nach der Installation von python, bitte installieren Sie Requests Module

```
pip install requests
```



Warnung: Das Skript wird nur zur Veranschaulichung bereitgestellt und soll veranschaulichen, wie die Funktion zur Isolierung von Endpunkten mithilfe der API automatisiert werden kann. Das Cisco Technical Assistance Center (TAC) ist nicht an der Fehlerbehebung für dieses Skript beteiligt. Benutzer müssen das Skript in einer sicheren Umgebung mit Vorsicht testen, bevor es in einer Produktionsumgebung bereitgestellt wird.

Skript

Sie können das bereitgestellte Skript verwenden, um die Isolierung auf mehreren Endgeräten in Ihrem Unternehmen zu starten:

```
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)
```

Anweisungen

- Informationen zum Generieren der AMP-API-Anmeldeinformationen finden Sie unter Übersicht über die Cisco AMP für Endgeräte-API
- Verwenden Sie die für Ihre Region angegebene BASE_URL:

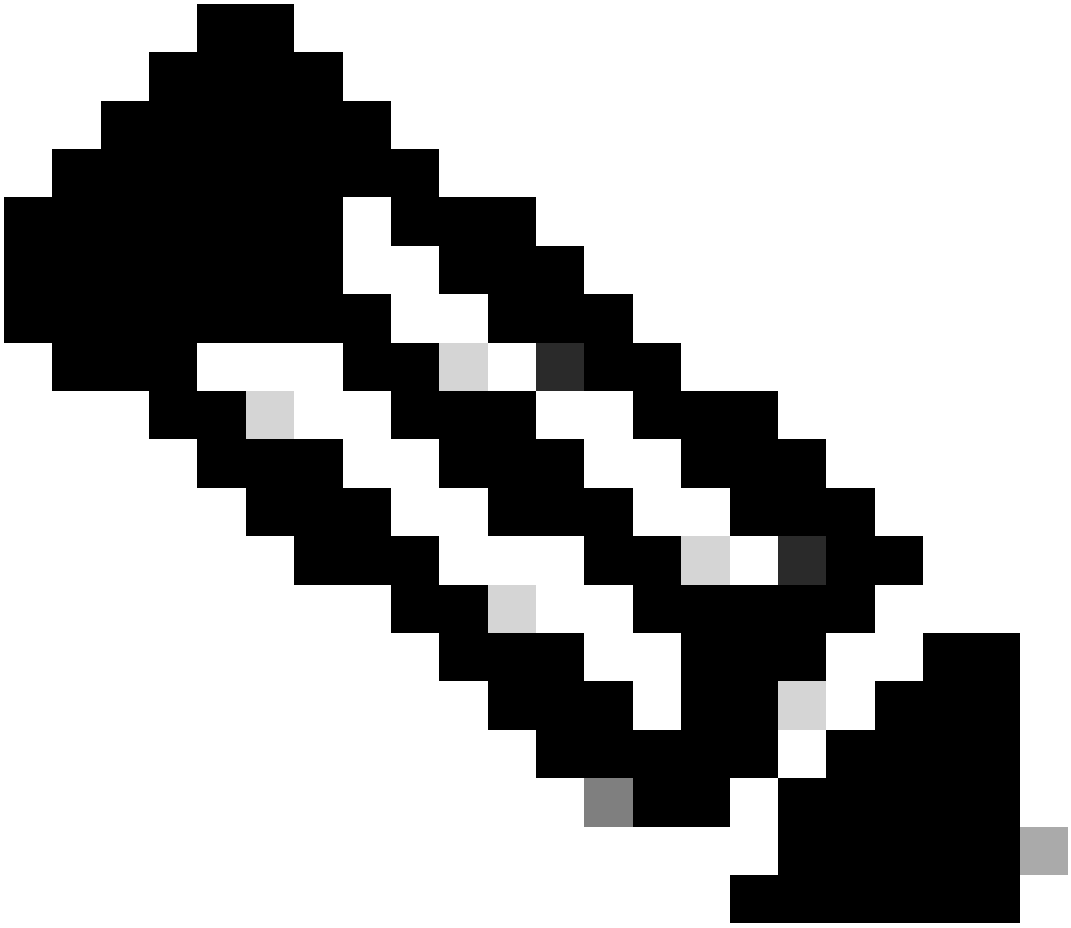
```
NAM - https://api.amp.cisco.com/v1/computers/
EU - https://api.eu.amp.cisco.com/v1/computers/
APJC - https://api.apjc.amp.cisco.com/v1/computers/
```

- Erstellen Sie eine config.txt-Datei im gleichen Verzeichnis wie das Skript mit dem genannten Inhalt. Beispiel für die Datei "config.txt":

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

- Erstellen Sie eine Datei guids.txt im gleichen Verzeichnis wie das Skript mit der Liste der Connector-GUIDs, eine pro Zeile. Fügen Sie nach Bedarf so viele GUIDs hinzu. Beispiel für die Datei guids.txt:

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



Hinweis: Sie können die GUIDs Ihrer Endpunkte entweder über die API GET /v1/computers oder über die Cisco Secure Endpoint Console erfassen, indem Sie zu Management > Computers navigieren, den Eintrag für einen bestimmten Endpunkt erweitern und die Connector GUID kopieren.

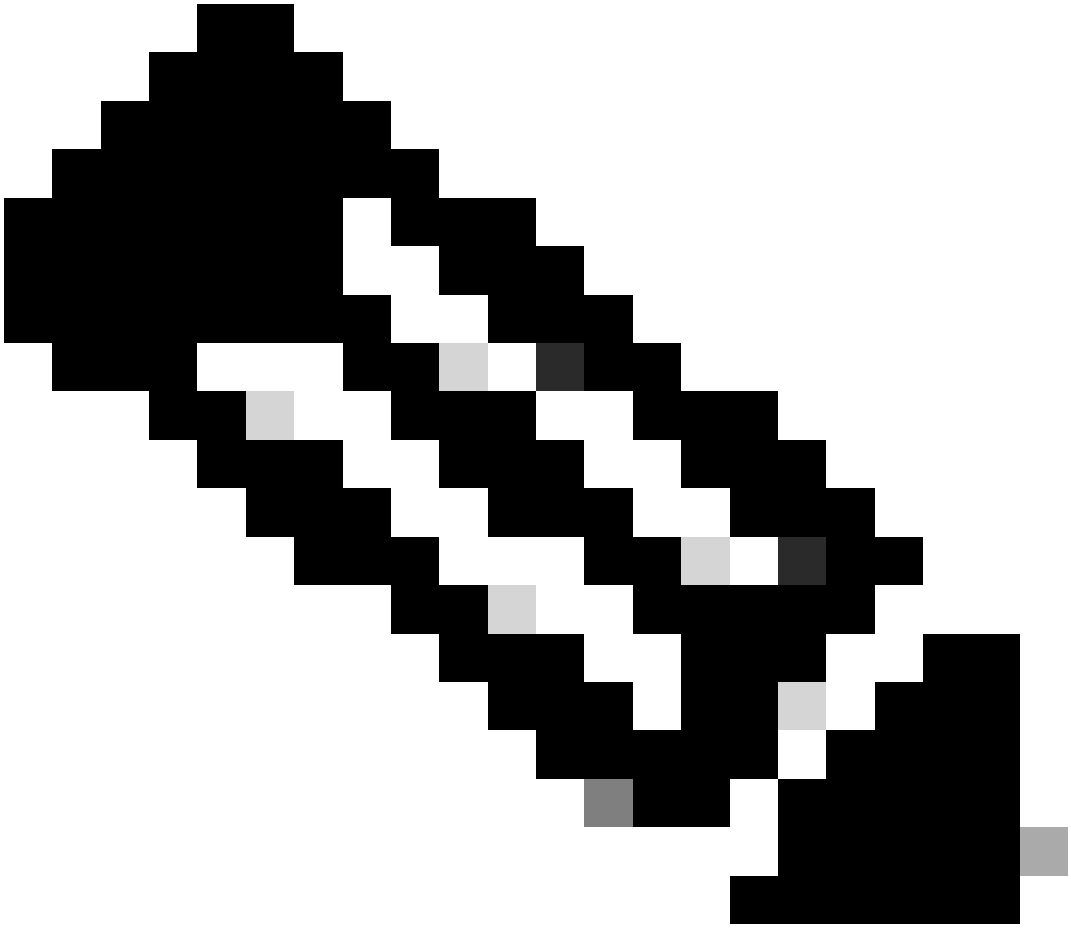
- Öffnen Sie ein Terminal oder eine Eingabeaufforderung. Navigieren Sie zum Verzeichnis, in dem sich start_isolation_script.py befindet.
- Führen Sie das Skript mit dem folgenden Befehl aus:

```
python start_isolation_script.py
```

Überprüfung

- Das Skript versucht, alle in der Datei guids.txt angegebenen Endpunkte zu isolieren.

- Überprüfen Sie das Terminal oder die Eingabeaufforderung auf Erfolgs- oder Fehlermeldungen für die einzelnen Endpunkte.



Hinweis: Das angefügte Skript start_isolation.py kann verwendet werden, um die Isolation auf Endpunkten zu initiieren, während stop_isolation.py dazu dient, die Isolation auf Endpunkten zu stoppen. Alle Anweisungen zur Ausführung und Ausführung des Skripts bleiben unverändert.

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.