Notes

Cisco Device Eraser Script

Introduction
This Python script provides a solution for erasing Cisco devices such as switches or routers. It automates the process of erasing the startup configuration and reloading the device, effectively resetting it to factory defaults.

How it works
The script establishes a serial connection to the Cisco device through a specified COM port and BAUD rate. It sends commands to the device to enter privileged EXEC mode, erase the startup configuration, and reload the device. After initiating the erase process, it waits for the device to complete the operation and then saves the erase log to a specified file.

Features
- Establishes a serial connection with Cisco devices.
- Automatically enters privileged EXEC mode if required.
- Retrieves the running configuration.
- Saves the configuration to a specified file.

Getting Started
1. Connect the Cisco device to your computer via a serial connection.
2. Determine the COM port to which the device is connected and the appropriate BAUD rate.
3. Run the script and provide the necessary inputs when prompted:
4. Select the COM port corresponding to the connected device.
5. Enter the BAUD rate.
6. Provide the device password.
7. Specify the filename for the erase log.
8. The script will initiate the erase process and provide status updates. Once completed, the erase log will be saved to the specified file.

Dependencies
- Python 3.x
- pyserial library

License
This project is licensed under the MIT License.

GitHub
Share Link: https://github.com/Anthony-Constant/Cisco-Device-Eraser

PYTHON COPY & PASTED LOCAL SOURCE CODE

```
#################################################################
# Script to erase cisco devices i.e. switches/routers           #
# Author: Anthony Constant                                      #
# Date: 20/04/2024                                              #
#################################################################


import serial
import time
import serial.tools.list_ports

def read_until_prompt(ser, prompt, timeout=5):
    start_time = time.time()
    response = b""
    while True:
        if time.time() - start_time > timeout:
            break
        data = ser.read()
        if data:
            response += data
            if prompt in response:
                break
    return response

def erase_device(com_port, baud_rate, password, log_file):
    try:
        # Serial connection setup
        ser = serial.Serial(com_port, baud_rate, timeout=1)
        ser.write(b"\r\n")
        print("Initiating device erase process…")

        # Give some time for the serial connection to stabilize
        time.sleep(2)
```

```
        # Send command to the device to enter privileged EXEC mode
        print("Entering privileged EXEC mode...")
        ser.write(b"enable\r\n")

        # Check if already in enable mode
        output = read_until_prompt(ser, b"#")
        if b"Password:" in output:
            print("Sending password...")
            ser.write(password.encode('utf-8') + b"\r\n")
            read_until_prompt(ser, b"#")

        # Send commands to erase startup-config and reload the device
        print("Erasing startup configuration...")
        ser.write(b"erase startup-config\r\n")
        ser.write(b"\r\n")
        ser.write(b"reload\r\n")

        # Introduce a delay to allow the device to process the commands
        print("Waiting for device to complete the process...")
        time.sleep(30)  # Adjust this delay as needed

        # Read the output until no data is received for a short period
        erase_output = ser.read_all().decode()

        # Save the erase log to a file
        with open(log_file, "w") as file:
            file.write(erase_output)

        print(f"Device erased successfully. Erase log saved to {log_file}")

        # Close serial connection
        ser.close()

    except serial.SerialException as se:
        print(f"Serial port error: {se}")
    except Exception as e:
        print(f"An error occurred: {e}")
```

```python
if __name__ == "__main__":
    try:
        available_ports = list(serial.tools.list_ports.comports())
        print("Available COM ports:")
        for idx, port in enumerate(available_ports):
            print(f"{idx + 1}. {port.device}")

        selection = int(input("Enter the number corresponding to the desired COM port: "))
        selected_port = available_ports[selection - 1].device
        baud_rate = int(input("Enter the BAUD rate: "))
        password = input("Enter your password: ")
        log_file = input("Enter the filename for the erase log (e.g., erase_log.txt): ")

        erase_device(selected_port, baud_rate, password, log_file)
    except (IndexError, ValueError):
        print("Invalid selection. Please enter a valid number.")
```