Notes

Advanced_Ping_Verification.py

Introduction

This is an advanced Ping verification script created in Python by Anthony Constant (AC). The purpose of this script is to automate the process of pinging a maximum number of devices inside a network and return whether the network status is successful or unsuccessful. The script uses the write method to create/open/write the status to a file called ping_logs.txt.

How to Use

To use this script, you need to specify the maximum number of target IP addresses in the range method. The script will then ping each IP address in the range and return whether the ping was successful or unsuccessful. The status of each ping will be written to the ping_logs.txt file.

Requirements

This script requires Python 3 to be installed on your machine.

How It Works

The script uses the os module to execute the Ping command in the command prompt. It then captures the output of the command to determine whether the Ping was successful or not. The status of each Ping is then written to the ping_logs.txt file using the write method.

References

If you would like to learn more about Ping and Traceroute commands, you can refer to the following link:
https://www.cisco.com/c/en/us/support/docs/ios-nx-os-software/ios-software-releases-121-mainline/12778-ping-traceroute.html

Credits

This script was created by Anthony Constant (AC). If you have any questions or suggestions, you can contact him at anthonyconstant.co.uk/

License

This script is released under the MIT License. See the LICENSE file for more details.

REPL.IT

Share Link: https://replit.com/@Ant94x/Advanced-Ping-Verification?v=1

GitHub

Share Link: https://github.com/Anthony-Constant/Advanced-Ping-Verification

PYTHON COPY & PASTED LOCAL SOURCE CODE

```
# Advanced_Ping_Verification.py
# Created a Advanced Ping verification script in Python
# Author: Anthony Constant (AC)

################################# SOME NOTES #########################################

## Advanced Ping automation script (specify the range method to the maximum number of hosts to ping inside the network and use 'write' to create/open/write the status to ping_logs.txt file.)


######################## HOW DOES IT WORK ##############################################
## Ping a maximum number of devices inside a network to return whether the network status is successful or unsuccessful.

## Specify the maximum number of target IP addresses in the range method.


############################# REFERENCES ###############################################

## https://www.cisco.com/c/en/us/support/docs/ios-nx-os-software/ios-software-releases-121-mainline/12778-ping-traceroute.html


########################################################################################
############################ START PROJECT HERE #######################################
############################################################################################

import os


OS_TYPE = os.name ## verifies the OS type
count = '-n' if OS_TYPE == 'nt' else '-c' ## Sets the count modififer to the OS type. Specify amount of pings
logs_file = open("ping_logs.txt", "a" or "w") ## create a variable logs_file which opens/creates a ping_logs.txt file then writes the feedback status of the network to it.
ip_list = [] ## create an empty list variable called ip_list to store ip addresses
```

```
for ip in range(1, 10): ## create a for loop using the range method to loop through 1-256 hosts (change 256 to specify maximum number of hosts in a network)
    ip_list.append("192.168.5." + str(ip)) ## append the ip_list starting with the default gateway (192.168.1.1) up to maximum hosts specified inside the range method.
    ## convert the ip variable into a string format so we can use it.

for ip in ip_list: ## create a for loop here to loop through each ip address of ip_list
    response = os.popen(f"ping {ip} {count} 1").read() ## use -n 1 to send just one ping to the device to save time
    if "Received = 1" and "Approximate" in response: ## check to see if the response has "Received =1" AND Approximate inside the response variable. For more specific results we check for Approximate with response
        print(f"{ip} Ping was Successful! ")
        logs_file.write(f"{ip} Ping Successful " + "\n") ## write this to the logs_file
    else:
        print(f"{ip} Ping was Unsuccessful! ")
        logs_file.write(f"{ip} Ping Unsuccessful " + "\n") ## write this to the logs_file
```

[redacted]