Notes

Port_Scanner.py

A Python script to scan a target IP address and determine which ports are open.

How to use

Enter the target IP address when prompted.

The script will automatically scan all 1000 ports and determine which ones are open on the target machine.

Once the scan is complete, the script will print "Port Scan Complete!" in the terminal.

The script will also create and write to a "port_logs.txt" file on your local machine, which will contain all open ports found during the scan.

References

Python documentation on concurrent.futures

Python Programming/Threading

Python documentation on socket

Palo Alto Networks Cyberpedia: Port Scan

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

GitHub

Share Link: https://github.com/Anthony-Constant/Port-Scanner

PYTHON COPY & PASTED LOCAL SOURCE CODE

```
# # Port_Scanner.py
# Created a Port scanning script in Python
# Author: Anthony Constant (AC)

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

## A port scanner to be used on a target IP address.

######################## HOW DOES IT WORK ############################################

## Enter the target IP address
## Let the script do the work and return which ports are available/open on the target machine.
## the script will tell you once the scan is complete.
## the script will automatically create and write to a port_logs.txt file on your local machine.



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

## https://docs.python.org/3/library/concurrent.futures.html
##
https://en.wikibooks.org/wiki/Python_Programming/Threading#:~:text=Threading%20in%20python%20is%20used,calls)%20at%20the%20same%20time.&te
xt=Threading%20allows%20python%20to%20execute,simulated%20with%20the%20sleep%20function.
## https://docs.python.org/3/library/socket.html
## https://www.paloaltonetworks.com/cyberpedia/what-is-a-port-scan


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

import socket ## use this to connect to the IP and check if the port is open
import threading ## used for print log
```

```python
import concurrent.futures ## used for loop threading to scan all ports
import os ## used to automatically create/write to a port_logs.txt file

print_lock = threading.Lock() ## use this to print new line for each open port found.
logs_file = open("port_logs.txt", "a" or "w") ## create and open a port_logs.txt file on the local machine ready to write all open ports to the file for automation storage.


ip = input("Enter the Target IP address to scan: ") ## allow for user to enter the target ip address as input

def scan(ip, port):
    scanner = socket.socket(socket.AF_INET, socket.SOCK_STREAM) ## use INET to specify IPv4 to connect to outside host. use Sock_stream for the TCP protocol.
    scanner.settimeout(2) ## wait 2 seconds before continue to the next port and decide if its opened or closed.
    try:
        scanner.connect((ip,port)) ## if the scanner is connected.
        scanner.close() ## close the connection instantly.
        with print_lock: ## new line for each open port found.
            print(f"Port {port} open.  ") ## show this in the terminal to notify user port is open
            logs_file.write(f"Port {port} is open on target {ip} " + "\n") ## write this to the port_logs.txt file

    except:
        pass ## use pass in this case as only thing which causes an error is not connecting to the port.


with concurrent.futures.ThreadPoolExecutor(max_workers = 100) as executor: ## package this in variable executor.
    for port in range(1000): ## scan the range of 1000 ports
        executor.submit(scan, ip, port + 1) ## use executor to call the scan function with variables ip, port. use + 1 to avoid port 0(invalid)

print("\nPort Scan Complete!") ## show this in the terminal to notify the user the scan is now complete.
```