PythonPing Network Scanner Tutorial

Contents

Time required: 60 minutes

Python Tabs and Spaces Issue

Visual Studio Code automatically changes a tab into four spaces. Other editors, like geany and nano in Linux, do not. You can end up with a combination of spaces and tabs. Python doesn't like a combination; it wants either one or the other. The preferred method is spaces.

Recommendation:

1. Create your Python files in Visual Studio Code in Windows.
2. Copy and paste the code into either nano or geany in Linux.

Objective: Write a cross platform Python script that uses branching, looping, and pythonping to scan a local network.

Find Network Address

If you are using a VM, set to a bridged adapter.

Find Your Network IP Address in Windows

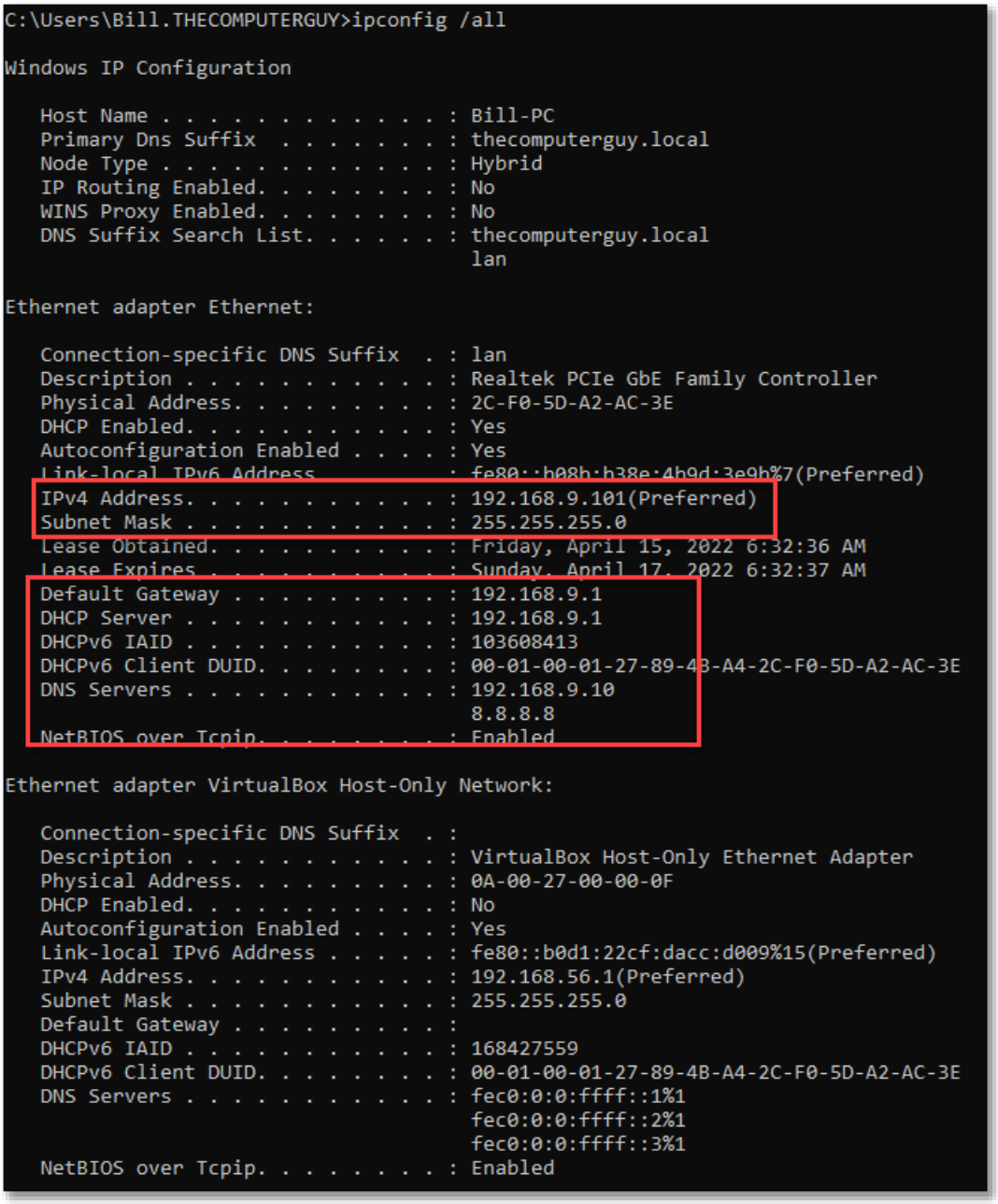
Use the network address of your local network. Example: 192.168.0.0/24

NOTE: 192.168.56.1 is the VirtualBox adapter address, that is not your network address.

1. Enter the following command at the command prompt: **ipconfig /all**
2. The screenshot below shows my network at home, 192.168.9 Your IP address will probably be different. I have an Ethernet adapter, you may have a wireless adapter.
3. Notice that my IP address information includes a Default Gateway, DHCP Server, and DNS Servers. Those are needed for a functioning network connection.
4. Note that my IPv4 Address for my computer is **192.168.9.101** My subnet Mask is **255.255.255.0** This makes my network a standard Class C network.

NOTE: A typical home network address is **192.168.1.0/24**

If you are not sure about this, please contact me. You will get a 0 for this assignment if you do not provide a screenshot showing a successful scan of your network.



How It Works

In this example, the **scan()** function takes a subnet as input (e.g., "192.168.1" for a typical home network) and scans all IP addresses in that subnet using ICMP ping. It uses the pythonping library to send ICMP Echo Request packets and receive ICMP Echo Reply packets to determine if hosts are online. If a host responds to the ping, its IP address is printed along with the response time, and it's added to the list of live hosts. Finally, the list of live hosts is printed at the end.

Note: ICMP ping may not always work, as some networks or hosts may have ICMP ping disabled for security reasons. In such cases, the `pythonping` library may not be effective for scanning hosts. Please ensure that you have the necessary permissions and legal rights before scanning any network or host. Always follow ethical guidelines and applicable laws when conducting network scanning or security testing.

Tutorial 1: PythonPing

Security professionals often need to automate or create tools to help them conduct security tests. In this activity, you will write a Python script that uses the ping command and a for loop to ping IP numbers for an entire network.

1. Install the pythonPing library. Enter the following command at a command prompt.

```
# Windows
pip install pythonping
# Linux
sudo apt install python3-pythonping
```

2. Create a new file called **pythonping_scanner_1.py**.
3. Enter the following code including the comments.

```python
#!/usr/bin/env python3
"""
    Filename: pythonping_scanner_1.py
    This program prompts the user to enter a host address
    Uses the pythonping library to send out ping packets
"""
# Windows: pip install pythonping
# Kali Linux distributions:
# sudo apt update
# sudo apt install python3-pythonping
# Use sudo to run script: sudo python3 pythonping_scanner.py
from pythonping import ping


def main():
    # Print program title
    print(48 * "-")
    print("|             Python Network Ping             |")
    print(48 * "-")

    # Get ip address or hostname
    host_address = input(" Enter single IP address or hostname: ")

    # Scan host
    result = scan(host_address)
    print(result)


# ----------------------- PING HOST ADDRESS ---------------------------#
def scan(host_address: str) -> str:
    """Ping a single target, return a string response."""
    # Ping the IP address
    result = ping(host_address)
    return result


# If a standalone program, call the main function
# Else, use as a module
if __name__ == "__main__":
    main()
```

Example run:

This program pings a single IP address or host name.

```
-----------------------------------------------
|             Python Network Ping             |
-----------------------------------------------
 Enter single IP address or hostname: 192.168.9.1
Reply from 192.168.9.1, 29 bytes in 0.67ms
Reply from 192.168.9.1, 29 bytes in 0.45ms
Reply from 192.168.9.1, 29 bytes in 0.43ms
Reply from 192.168.9.1, 29 bytes in 0.31ms
```

```
-----------------------------------------------
|             Python Network Ping             |
-----------------------------------------------
Enter single IP address or hostname: lab.wncc.edu
Reply from 198.206.239.241, 29 bytes in 34.3ms
Reply from 198.206.239.241, 29 bytes in 21.79ms
Reply from 198.206.239.241, 29 bytes in 21.83ms
Reply from 198.206.239.241, 29 bytes in 25.57ms

Round Trip Times min/avg/max is 21.79/25.87/34.3 ms
```

Tutorial 2: Scan Your Network from Windows

Let’s scan your entire local network. Only scan networks where you have permission.

```python
#!/usr/bin/env python3
"""
Filename: pythonping_scanner_2.py
This program prompts the user to enter a class C network address
it uses the pythonping library to send out ping packets
"""
# https://docs.python.org/3/library/ipaddress.html
# Convert ip/mask to list of hosts
import ipaddress
import sys

# Windows: pip install pythonping
# Linux Debian distributions:
# sudo apt update
# sudo apt install python3-pythonping
# Use sudo to run script: sudo python3 pythonping_scanner.py
from pythonping import ping


def main():
    display_program_title()

    # ------------------------ SET NETWORK ADDRESS ------------------------ #
    """Set network address x.x.x.x/x or x.x.x.x/x.x.x.x from user"""
    # Change this to the default value of your network
    default_local_network = "10.0.1.0/24"

    # Prompt the user to input a network address and press Enter
    # If they press enter without an network address, the default is used
    network_address = (
        input("\n Enter your network address (ex. 10.0.1.0/24): ")
        or default_local_network
    )

    # Create a network address object from user input
    ip_net = ipaddress.ip_network(network_address)

    # Convert all hosts on entered network into a list
    all_hosts = list(ip_net.hosts())
    scan(all_hosts)
```

```python
# ------------------------ SCAN NETWORK ---------------------------------- #
def scan(all_hosts: str):
    """Ping all Class C IP addresses 1-254"""
    for host_address in all_hosts:
        # Convert the ip address to a string
        ip = str(host_address)
        try:
            # Ping the IP address with two packets
            result = ping(
                ip,  # Target IP address
                count=1,  # Number of pings
                timeout=2,  # Timeout in seconds
            )

            # If there was a successful ping response
            if result.success():
                # Response time less than 2000ms, target is active
                print(f" {ip:14}-> RTT: {result.rtt_avg_ms:>6.2f}ms")
            else:
                print(f" {ip} Inactive")

        except KeyboardInterrupt:
            # Catch the Keyboard Interrupt exception
            print("CTRL-C pressed. Exit program")
            sys.exit()

        except Exception as e:
            # Catch all other exceptions
            # Print out the exception error for debugging
            print("Sorry", e)
            sys.exit()
```

```
76  # -------------------- DISPLAY PROGRAM TITLE ------------------------------ #
77  def display_program_title():
78      """Print program title to console"""
79      print(35 * "-")
80      print("|   Python Network Ping Scanner   |")
81      print(35 * "-")
82      print(" Press CTRL C to exit")
83
84
85  # If a standalone program, call the main function
86  # Else, use as a module
87  if __name__ == "__main__":
88      main()
```

1. Open a command prompt and navigate to the folder containing your **pythonping_scanner.py** file.
2. Type: **python pythonping_scanner.py** Press **Enter**.
3. If you have no errors, your program should begin pinging IP addresses, as shown below.
4. You can let the program run to completion (which will take a long time) or press **CTRL C** to terminate the script.

Example run in Windows:

```
-----------------------------------
|   Python Network Ping Scanner   |
-----------------------------------
 Press CTRL C to exit

 Enter your network address (ex. 192.168.9.0/24): 192.168.9.0/24
 192.168.9.1 --> Alive RTT:  0.53
 192.168.9.2 Inactive
 192.168.9.3 Inactive
 192.168.9.4 Inactive
 192.168.9.5 Inactive
 192.168.9.6 Inactive
 192.168.9.7 Inactive
 192.168.9.8 Inactive
 192.168.9.9 Inactive
 192.168.9.10 --> Alive RTT:  0.39
 192.168.9.11 Inactive
```

Tutorial 3: Scan Your Network from Linux

NOTE: If your class is using Linux for other assignments, complete this portion of the assignment.

This is a cross platform script. It will work on Windows or Linux or Mac. Copy and paste the code into Linux.

Your network in Linux should be the same as your network in Windows. If it isn't, set your VirtualBox network settings to Bridged Adapter.

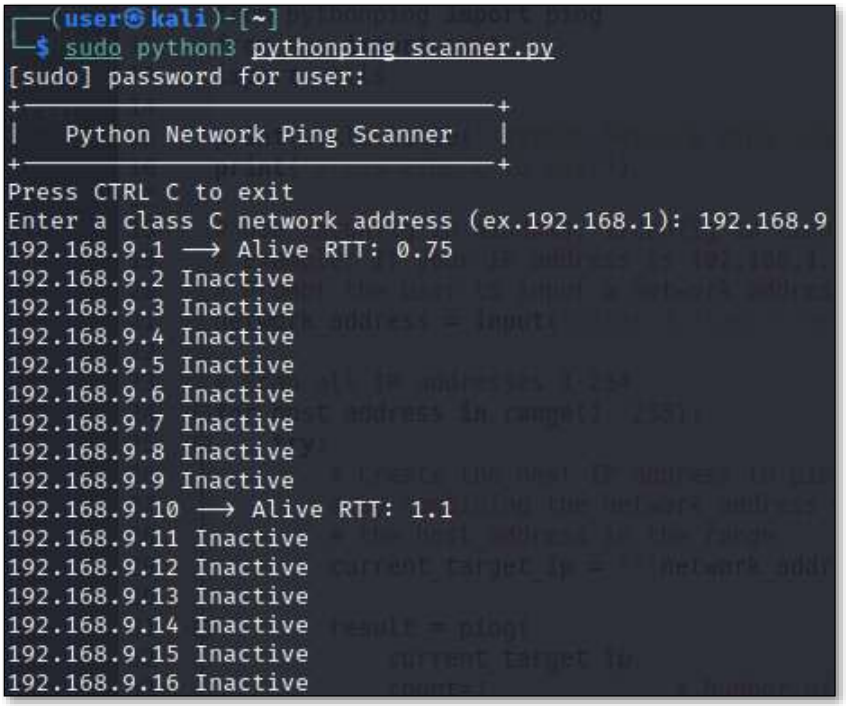
PythonPing Scanner in Linux

This will work on most Linux systems including older distributions of Kali Linux. Try this first.

1. Open a terminal session.
2. Install the **pythonping** library → **sudo apt install python3-pythonping**
3. Create a Python file with nano or geany or an editor of your choice.

```
nano pythonping_scanner.py
```

4. Copy and paste the code from your Windows version of the file into your new file.
5. In a terminal → **sudo python3 pythonping_scanner.py**
6. Example run in Linux:



This scanner isn't very fast, but it works. To get faster results, we would use multithreading.

Coming up in the next tutorial!

Assignment Submission

1. Attach all program files.
2. Attach a screenshot of each successful program run.
3. If you do not attach a screenshot of a successful program run on your correct local network address, you will receive a 0 for this assignment.
4. Submit the assignment in Blackboard.