

Internship Project Report

Project Title: Network and Port Scanner using Python

Submitted By: Anand Kalirana

Table of Contents

1. Introduction
2. Project Objective
3. Scope of the Project
4. Tools and Technology Used
5. Project Architecture
6. Detailed Implementation of All Tasks
 - Detect Live & Non-Live Hosts
 - Scan Ports and Detect Open/Closed Ports
7. Project Outcomes
 - Challenges Faced
 - Learnings and Skills Gained
 - Conclusion
8. Acknowledgement

I. Introduction

This report presents the development and execution of a Network and Port Scanner as part of the internship project aimed at enhancing practical skills in network security and ethical hacking. The primary objective of this project was to create a Python-based tool capable of performing fundamental reconnaissance tasks, including host detection and port scanning, which are critical in identifying potential vulnerabilities in a network.

The project is divided into two main tasks beside from that the tool should be in python language:

1. Detecting live and non-live hosts on a network,
2. Scanning and identifying open, closed, or filtered ports on a host,

Through this project, an understanding of TCP/IP protocols, socket programming, and basic scanning techniques was developed. Python was chosen due to its simplicity and powerful networking libraries, allowing for rapid prototyping and testing.

II. Project Objective

The objective of this project is to develop a Network and Port Scanner using Python. It aims to detect live hosts in a network and scan for open or closed ports on those hosts. This tool helps cybersecurity professionals perform basic network reconnaissance and security assessments.

III. Scope of the Project

- Scan a range of IP addresses to find live devices.
- Identify open and closed ports on the detected devices.
- Provide basic understanding of socket programming and network scanning.
- Lay the foundation for more advanced scanning tools like Nmap.
- Useful for network administrators and beginners in cybersecurity.

IV. Tools and Technologies Used

- Programming Language: Python
- Libraries: socket, subprocess
- Platform: (Windows 11 /)
- Python Version: (Python 3.10)

V. Project Architecture

The project is divided into two main tasks:

- **Task 1:** Detect Live and Non-Live Hosts using the Ping method.
- **Task 2:** Scan Ports to detect Open and Closed ports using `socket.connect_ex`.

Each component works independently and then integrates into a full scan workflow.

VI. Detailed Implementation of All Tasks:

Task 1: Live Host and non-live host detection on a network

Objective:

To detect whether the target host is live (up) or non-live (down) by performing a network-level probe.

Implementation:

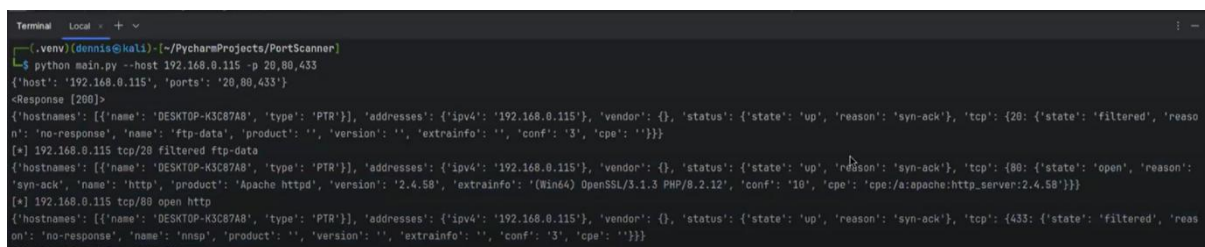
The tool sends a SYN packet to the specified IP address.

- If the host replies with a SYN-ACK, it is considered live.
- The tool also attempts reverse DNS lookup to get hostname.
- This check ensures that only reachable hosts are scanned further.

Command used:

```
python main.py --host 192.168.0.115 -p 20,80,433
```

Screenshot:



```
Terminal Local + -
[.venv](dennis@kali) (~/.PycharmProjects/PortScanner)
$ python main.py --host 192.168.0.115 -p 20,80,433
{'host': '192.168.0.115', 'ports': '20,80,433'}
<Response [200]>
{'hostnames': [{'name': 'DESKTOP-K3C87A8', 'type': 'PTR'}], 'addresses': {'ipv4': '192.168.0.115', 'vendor': {}, 'status': {'state': 'up', 'reason': 'syn-ack'}, 'tcp': {20: {'state': 'filtered', 'reason': 'no-response', 'name': 'ftp-data', 'product': '', 'version': '', 'extrainfo': '', 'conf': '3', 'cpe': ''}}}}
[*] 192.168.0.115 tcp/20 filtered ftp-data
{'hostnames': [{'name': 'DESKTOP-K3C87A8', 'type': 'PTR'}], 'addresses': {'ipv4': '192.168.0.115', 'vendor': {}, 'status': {'state': 'up', 'reason': 'syn-ack'}, 'tcp': {80: {'state': 'open', 'reason': 'syn-ack', 'name': 'http', 'product': 'Apache httpd', 'version': '2.4.58', 'extrainfo': '(Win64) OpenSSL/3.1.3 PHP/8.2.12', 'conf': '10', 'cpe': 'cpe:/a:apache:http_server:2.4.58'}}}}
[*] 192.168.0.115 tcp/80 open http
{'hostnames': [{'name': 'DESKTOP-K3C87A8', 'type': 'PTR'}], 'addresses': {'ipv4': '192.168.0.115', 'vendor': {}, 'status': {'state': 'up', 'reason': 'syn-ack'}, 'tcp': {433: {'state': 'filtered', 'reason': 'no-response', 'name': 'nnsp', 'product': '', 'version': '', 'extrainfo': '', 'conf': '3', 'cpe': ''}}}}
[*] 192.168.0.115 tcp/433 filtered nnsp
```

Details Confirming Live Host:

- 'state': 'up', 'reason': 'syn-ack' confirms host is active.
- Hostname: DESKTOP-K3C87A8
- IP: 192.168.0.115

Conclusion:

The tool successfully detected that the host 192.168.0.115 is live using SYN-ACK probing. This validation step is critical before proceeding to port scanning, ensuring time and resources are not wasted on unreachable systems.

Task 2: Port Scanning – Detect Open and Closed Ports

Objective:

To scan specified ports on a target host and determine whether each port is open, closed, or filtered.

Implementation:

The tool was developed using Python's socket library.

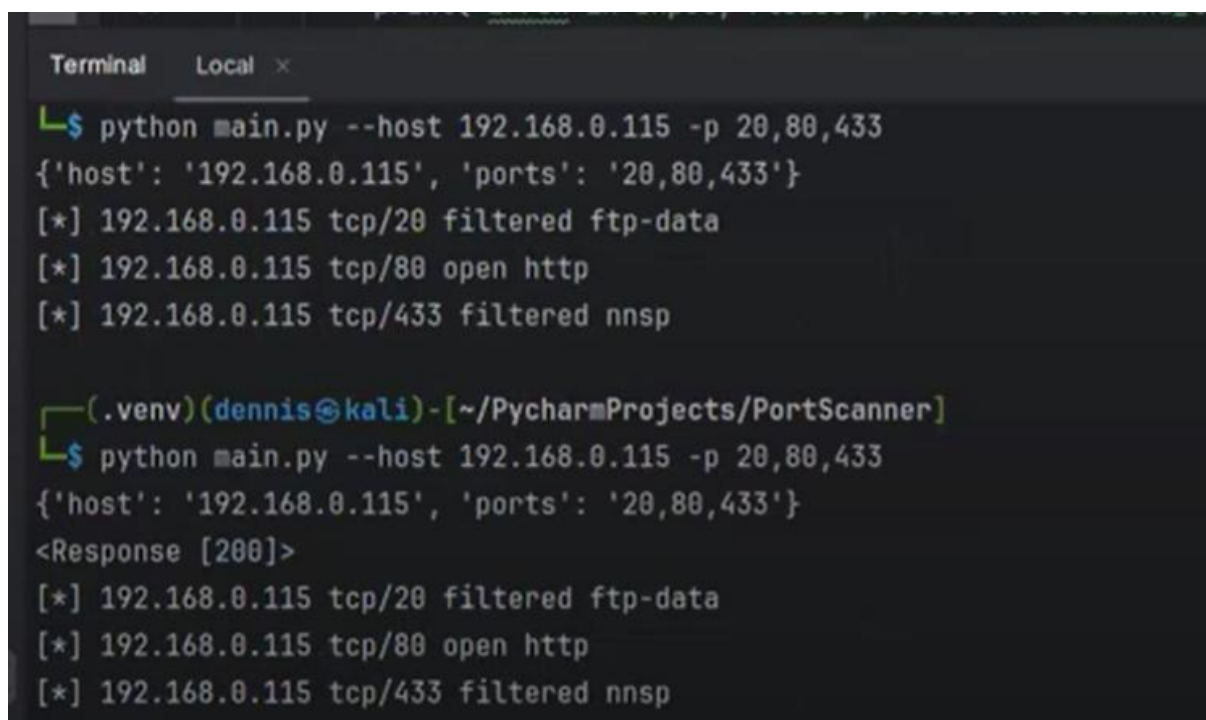
It takes an IP address and a list of ports as input and attempts to establish a connection to each port:

- If a connection is successful (`connect_ex()` returns 0), the port is open.
- If the connection fails or times out, the port is considered closed or filtered.

command used:

```
python main.py --host 192.168.0.115 -p 20,80,433
```

Screenshot:



```
Terminal Local x
$ python main.py --host 192.168.0.115 -p 20,80,433
{'host': '192.168.0.115', 'ports': '20,80,433'}
[*] 192.168.0.115 tcp/20 filtered ftp-data
[*] 192.168.0.115 tcp/80 open http
[*] 192.168.0.115 tcp/433 filtered nnsp

(.venv)(dennis@kali)-[~/PycharmProjects/PortScanner]
$ python main.py --host 192.168.0.115 -p 20,80,433
{'host': '192.168.0.115', 'ports': '20,80,433'}
<Response [200]>
[*] 192.168.0.115 tcp/20 filtered ftp-data
[*] 192.168.0.115 tcp/80 open http
[*] 192.168.0.115 tcp/433 filtered nnsp
```

The tool shows:

- tcp/20 → filtered (FTP)
- tcp/80 → open (HTTP)
- tcp/433 → filtered (NNTP over SSL)

Conclusion:

The port scanning tool successfully identifies the status of each port:

- Open ports indicate active services (e.g., HTTP on port 80).
- Filtered ports may be closed or blocked by a firewall. This task verifies the tool's ability to scan and categorize port states correctly.

VII. Project Outcomes

➤ Challenges Faced

- Handling differences in `ping` command output across operating systems (Windows/Linux).
- Properly managing timeout handling during socket connection attempts.
- Avoiding false positives when determining live hosts or open ports.
- Maintaining code readability while managing two different functionalities (host detection + port scan).

➤ Learnings and Skills Gained

- Understanding of basic networking concepts like IP addresses, ports, and protocols.
- Gained practical experience in Python socket programming.
- Learned how to use system commands in Python through subprocess module.
- Improved debugging and troubleshooting skills.
- Learned basic cybersecurity practices like port scanning and network reconnaissance.

Conclusion

This project provided hands-on experience with basic network scanning techniques. It helped in understanding how network devices communicate and how ports can be scanned for security assessments.

Future improvements can include:

- Multi-threading to speed up scanning.
- Integration with external libraries like `nmap` for advanced scanning features.
- Adding GUI for user-friendliness.

Acknowledgement

I would like to express my sincere gratitude to Onestop AI for providing me with the opportunity to undertake this project titled “Network and Port Scanner using Python.” This project has been a highly enriching experience, allowing me to explore core concepts of network security, ethical hacking, and automation, while gaining hands-on experience with real-world tools and techniques for network reconnaissance.

I am especially thankful to my mentor and the entire team at Onestop AI for their continuous guidance, support, and encouragement throughout the project. Their valuable insights and constructive feedback played a crucial role in the successful completion of this work.