| | |
|---|---|
| Course | COMP 7003 |
| Program | Bachelor of Science in Applied Computer Science |
| Term | January 2025 |

- This is an individual programming assignment.

# Objective

- This assignment will help you develop a custom SYN port scanner using Python and Scapy.
- Craft and send TCP `SYN` packets.
- Analyze network responses to identify open, closed, and filtered ports.
- Implement threading for efficient scanning.
- Use command-line arguments to control scanning behaviour.
- Identify running applications on open ports and analyze their security risks.

# Learning Outcomes

- Understand how `SYN` scanning works and why it is used.
- Be able to write a Python script to perform network reconnaissance.
- Use Scapy to send and receive raw packets.
- Interpret port scanning results and identify potential vulnerabilities.
- Use `lsof` to analyze open ports and their associated applications.
- Recognize security risks associated with running services.

# Assignment Details

## Scanner Program

- Accepts command-line arguments for target hosts and ports.
- Sends TCP `SYN` packets to the specified targets.
- If the host responds with TCP SYN/ACK then send a TCP `RST` packet.
- Interprets responses to classify ports as open, closed, or filtered.
- Displays results in a structured format (see the example at the end of this document).
- Your scanner must accept the following arguments:

```
usage: scanner.py [-h] [-t TARGET] [-p PORTS] [--show
SHOW]
```

```
Simple SYN Scanner using Scapy

options:
  -h, --help            Show this help message and exit
  -t TARGET, --target TARGET
                        Target IP, range, or subnet (e.g.,
192.168.1.1, 192.168.1.1-192.168.1.10, 192.168.1.0/24)
  -p PORTS, --ports PORTS
                        Port(s) to scan (e.g., 80, 1-100)
  --show SHOW           Filter results:
open,closed,filtered (comma-separated)
```

- If no target is provided, the scanner should scan the local subnet (/24).
- If no port is provided, the scanner should scan all 65535 ports.
- If no show is provided, the scanner should show OPEN, CLOSED, and FILTERED
- Based on the response:
  - SYN-ACK received → Port is OPEN.
  - RST received → Port is CLOSED.
  - No response → Port is FILTERED (possibly blocked by a firewall).
- The program must scan ports sequentially (no threading).

## Manual Service Identification

- Once the scan is complete:
  - Use lsof to find the running application for each open port:

    sudo lsof -i :PORT

- (Replace PORT with the actual open port.)
- Document the process name and PID of the application using the port.
- For each identified application, write a short security analysis covering:
  - What the program does.
  - Why might it be running on the system?
  - Potential security risks of exposing the service.

## Identifying Host Types from Open Ports

- Take the list of IP:port below and infer the type of host based on the services running on those ports.
- IANA Port Registry:
  - https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml
- AdminSub Port Finder:

    - https://www.adminsub.net/tcp-udp-port-finder
- For each open port below, answer:
    - What service is commonly associated with this port?
    - What type of device typically runs this service?
    - What are the potential security risks?
- For each host, make an informed guess as to what type of host it is (e.g. Linux, Windows, Wireless Access Point, Switch, iPhone, Android, smart lightbulb, etc…).
- Be sure to explain what makes you believe the host is that type.

```
- 192.168.0.1:21
- 192.168.0.1:53
- 192.168.0.1:1900
- 192.168.0.1:8200
- 192.168.0.1:20001
- 192.168.0.2:23
- 192.168.0.2:80
- 192.168.0.2:443
- 192.168.0.2:40001
- 192.168.0.2:40002
- 192.168.0.3:23
- 192.168.0.3:80
- 192.168.0.3:443
- 192.168.0.3:40001
- 192.168.0.3:40002
- 192.168.0.40:22
- 192.168.0.200:853
- 192.168.0.200:49152
- 192.168.0.200:62078
- 192.168.0.203:853
- 192.168.0.203:5000
- 192.168.0.203:7000
- 192.168.0.203:7100
- 192.168.0.203:49152
- 192.168.0.203:49159
- 192.168.0.203:61029
- 192.168.0.203:62078
```

## Requirements

- Run the scanner on localhost (127.0.0.1) for all ports:

```
sudo python3 scanner.py -t 127.0.0.1 --show open
```

- Run the scanner on a remote host, scanning all ports:

```
sudo python3 scanner.py -t <TARGET_IP> --show
open,filtered
```

- Run the scanner on all hosts scanning a specific port (22):

```
sudo python3 scanner.py -t 192.168.0.1-192.168.0.201 -p 22
--show open,closed
```

## Constraints

- You must use Python - no other languages are allowed.
- You must use Scapy for packet crafting, but no external scanning libraries..
- No multi-threading - scans must be performed sequentially.
- Only scan machines you own or have explicit permission to scan (you have permission to scan all machines in the lab between 192.168.0.1 and 192.168.0.201 - announce to anyone in the lab that you will be doing a scan so they can disconnect their laptop/phone/tablet/other devices before you do the scan if they wish).

# Resources

# Submission

- Ensure your submission meets all the guidelines, including formatting, file type, and submission.
- Follow the AI usage guidelines.
- Be aware of the late submission policy to avoid losing marks.
- ***Note: Please strictly adhere to the submission requirements to ensure you don't lose any marks.***

# Evaluation

| Topic | Value |
|---|---|
| SYN Scan | 15 |
| Arguments | 10 |
| Analysis of OPEN ports | 15 |
| Analysis of IP:port | 10 |
| Design | 20 |

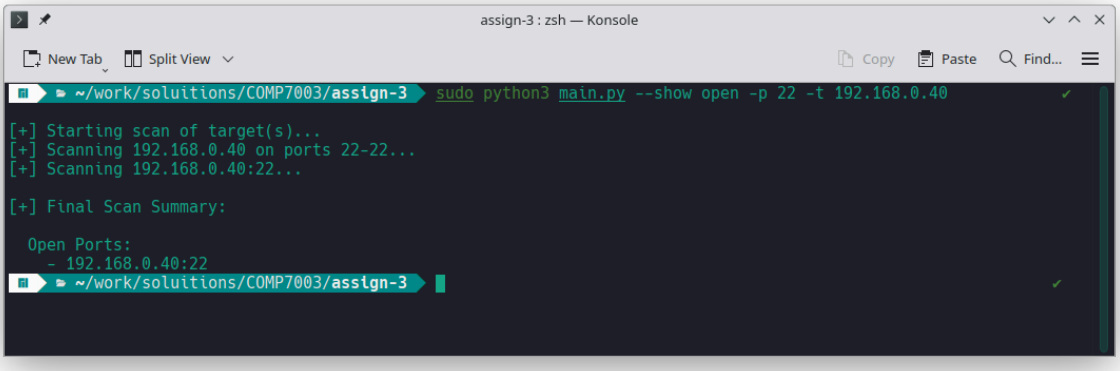
| | |
|---|---|
| Testing | 30 |
| Total | 100% |

# Hints

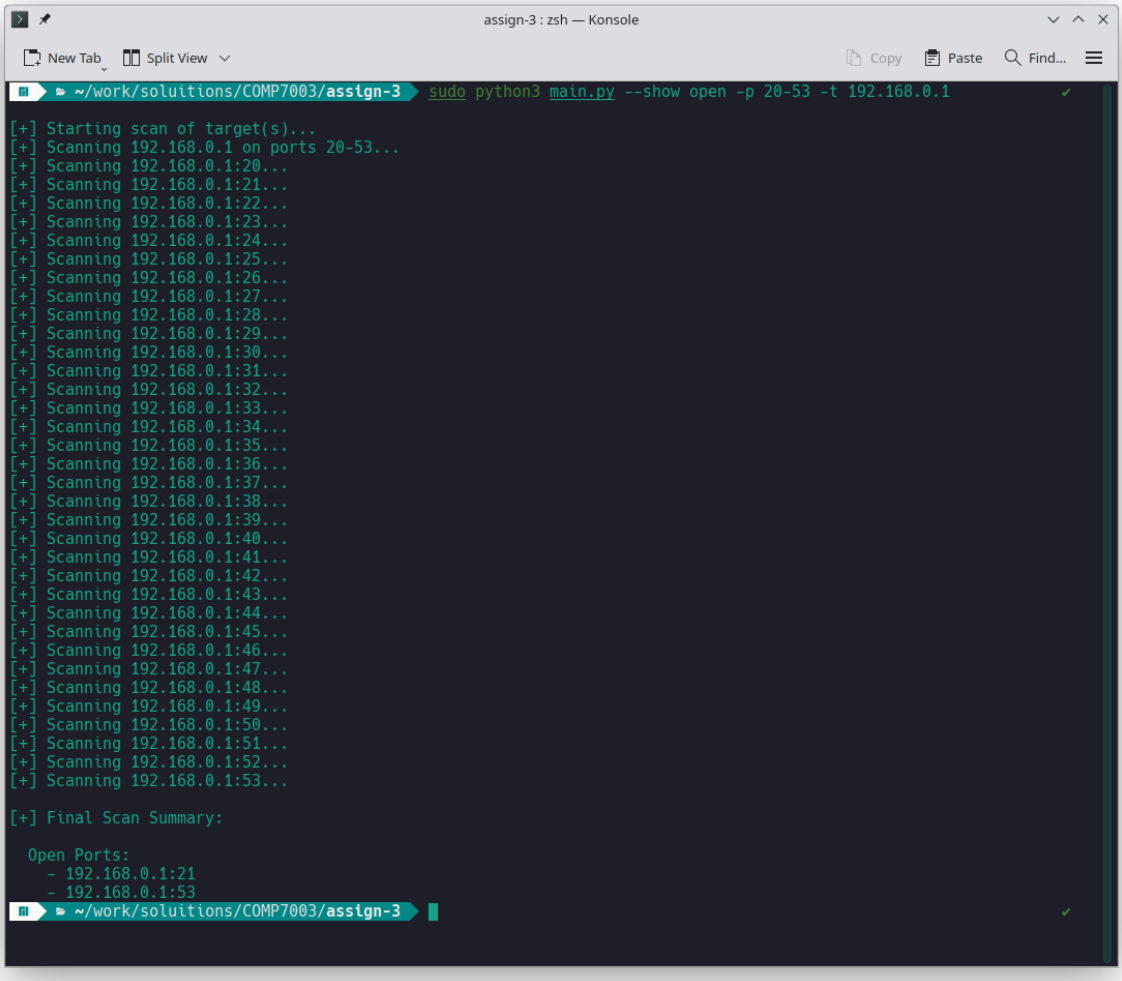
- Start by sending a `SYN` packet to a single port and checking the response.
- Use argparse to handle command-line arguments.
- Use `lsof -i :PORT` to identify running applications.
- `sudo` is required when running Scapy.
- Filter results with --show to test different cases.
- Document everything = your analysis is just as crucial as the scan results.

# Screenshots

## 1 host, 1 port



# 1 host, many ports



## Many hosts, 1 port

```
assign-3 : zsh — Konsole
New Tab  Split View                                   Copy  Paste  Find...
~/work/soluitions/COMP7003/assign-3  sudo python3 main.py --show open -p 23 -t 192.168.0.1-192.168.0.10

[+] Starting scan of target(s)...
[+] Scanning 192.168.0.1 on ports 23-23...
[+] Scanning 192.168.0.1:23...
[+] Scanning 192.168.0.2 on ports 23-23...
[+] Scanning 192.168.0.2:23...
[+] Scanning 192.168.0.3 on ports 23-23...
[+] Scanning 192.168.0.3:23...
[+] Scanning 192.168.0.4 on ports 23-23...
[+] Scanning 192.168.0.5 on ports 23-23...
[+] Scanning 192.168.0.6 on ports 23-23...
[+] Scanning 192.168.0.7 on ports 23-23...
[+] Scanning 192.168.0.8 on ports 23-23...
[+] Scanning 192.168.0.9 on ports 23-23...
[+] Scanning 192.168.0.10 on ports 23-23...

[+] Final Scan Summary:

  Open Ports:
    - 192.168.0.2:23
    - 192.168.0.3:23
~/w/soluitions/COMP7003/assign-3                                        ✓ 9s
```

## Many hosts, many ports

```
assign-3 : zsh — Konsole
New Tab  Split View                                   Copy  Paste  Find...
~/work/soluitions/COMP7003/assign-3  sudo python3 main.py --show open -p 23  -t 192.168.0.2,192.168.0.3

[+] Starting scan of target(s)...
[+] Scanning 192.168.0.2 on ports 23-23...
[+] Scanning 192.168.0.2:23...
[+] Scanning 192.168.0.3 on ports 23-23...
[+] Scanning 192.168.0.3:23...

[+] Final Scan Summary:

  Open Ports:
    - 192.168.0.2:23
    - 192.168.0.3:23
~/work/soluitions/COMP7003/assign-3                                     ✓
```

# Closed Ports