



# Project: Scanning Tool

A tool to demonstrate host and port scanning to students.

## Summary:

Create a tool that will help students better understand the concept of host and port scanning. The tool will only be limited to -

- ICMP scanning - to detect live host/s
- TCP scanning - to detect open port/s

## Details:

1. The tool should run on our **labtainer iptables2** environment. The tool will run inside the **client** workstation to probe the network within that environment.
2. The tool should also run on the "**student**" environment to scan the Proxmox subnet.
3. You can choose your own programming language, preferably something that is already installed on the "client" and the "student" environment - or - a programming language that can easily be installed on those environment.
4. The tool should be able to do ICMP scanning, in particular ICMP Echo Request.
5. The tool should be able to do the following TCP port scans:
  - TCP Connect Scan
  - Stealth Scans
    - TCP SYN (Half-open) Scan
    - Xmas Scan
    - FIN Scan
    - Null Scan
  - TCP ACK Scan

## Notes:

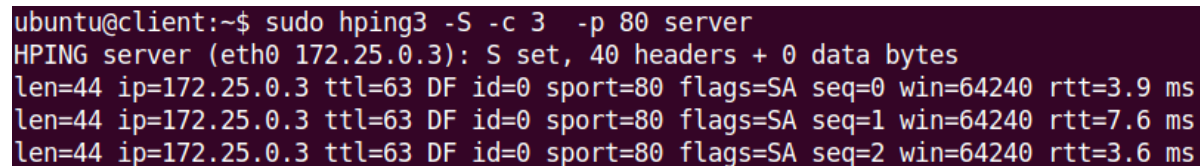
There are a lot of available materials/sources/guides on how to craft packets to produce these scans (ICMP and TCP). The challenge will be on how to “visually represent” these tests so that the student (using your tool) can easily understand the different concepts.

For example:

[1] If I run this command from the **client** workstation with the **server** as my target (inside the **labtainer iptables2**):

```
hping3 -S -c 3 -p 80 server
```

I will get these responses from hping3:

A terminal window with a dark purple background. The prompt is 'ubuntu@client:~\$'. The command entered is 'sudo hping3 -S -c 3 -p 80 server'. The output shows 'HPING server (eth0 172.25.0.3): S set, 40 headers + 0 data bytes' followed by three lines of packet details: 'len=44 ip=172.25.0.3 ttl=63 DF id=0 sport=80 flags=SA seq=0 win=64240 rtt=3.9 ms', 'len=44 ip=172.25.0.3 ttl=63 DF id=0 sport=80 flags=SA seq=1 win=64240 rtt=7.6 ms', and 'len=44 ip=172.25.0.3 ttl=63 DF id=0 sport=80 flags=SA seq=2 win=64240 rtt=3.6 ms'.

```
ubuntu@client:~$ sudo hping3 -S -c 3 -p 80 server
HPING server (eth0 172.25.0.3): S set, 40 headers + 0 data bytes
len=44 ip=172.25.0.3 ttl=63 DF id=0 sport=80 flags=SA seq=0 win=64240 rtt=3.9 ms
len=44 ip=172.25.0.3 ttl=63 DF id=0 sport=80 flags=SA seq=1 win=64240 rtt=7.6 ms
len=44 ip=172.25.0.3 ttl=63 DF id=0 sport=80 flags=SA seq=2 win=64240 rtt=3.6 ms
```

“-c 3” means, send only 3 packets

“-p 80” means, check port 80

“-S” means, send a SYN flag to the **server**, then the **server** will reply back (see image above).

From the above image, the **server (172.25.0.3)** replied with the flags = “SA” which means it acknowledges(ACK) the **client**’s SYN, and the **server** sends its own SYN to the **client**.

What would be better:

If we can show that the **client** sends out the packet to the **server**.

The **server** responded back.

Showing only the appropriate information we need for the student. No need for len, DF, id, etc.

[2] If I run these 2 commands on the **client** machine:

[client-terminal-1]:

```
sudo nmap -sS -p 80 -Pn server
```

Using nmap to do a TCP SYN port scan to the **server**.

```
ubuntu@client:~$ sudo nmap -sS -p 80 -Pn server
Starting Nmap 7.01 ( https://nmap.org ) at 2020-11-02 04:09 UTC
Nmap scan report for server (172.25.0.3)
Host is up (0.00017s latency).
PORT      STATE SERVICE
80/tcp    open  http
Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds
```

[client-terminal-2]:

```
sudo tcpdump -i eth0 ip --immediate-mode -l -n |cut -d' ' -f 3-7
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

172.24.0.3.50718 > 172.25.0.3.80: Flags [S],
172.25.0.3.80 > 172.24.0.3.50718: Flags [S.],
172.24.0.3.50718 > 172.25.0.3.80: Flags [R],
```

The command will capture the conversation between the **client** and **server** which is important for the student to understand the concept.

As you can see in the image above, the conversation is a little bit clearer between the **client** and the **server**.

The challenge: How can we make this conversation “more visually engaging” for the student to learn the concept?

6. The tool should have the following “options”

-v: Version/about.

-h: Help file/documentation.

-c X: Number of packets to send.

-t: The time spent by the tool.

-p <>: Port/s (single, range)

-<?>: Different options for the different ICMP and TCP scans. You can choose your own. You may follow the option switches used by hping3 (for example).

[host/s]: Target hosts, can be a single IP or a range of IPs.

### **Important Note re your environment:**

I suggests that you develop your code using the "**student**" environment first. You can use the Proxmox subnet for testing. Once running or if there is a need to test in your "**labtainer iptables2**" environment, you transfer the code and install to the **client** machine.

The reason am recommending this approach is because - when you or someone will do a "**labtainer iptables2 -r**" it will reset everything on those machines. It will erase your programming environment and your files. **Take note also that you are responsible for the proper upkeep of your codes(proper backups outside of Proxmox).**

### **Deliverables and points distribution:**

#### **[1](10 points)**

Completeness, quality, and how well the documentation is written for the target audience (student). This is the **-h** option (how to use the tool).

#### **[2](50 points)**

- 50: All the required features are present.
- 40: 1 to 3 features are missing.
- 30: >3 to 5 features are missing.
- 0: >5 features are missing.

#### **[3](30 points)**

- How visually engaging is the output of the tool?
- 20: Better than what the 2 examples (shown above; see Details #4) can provide, to help the student understand the concepts.
- 10: At par with the 2 examples (shown above; see Details #4).
- 0: Was not able to meet the level of examples (shown above; see Details #4).

#### **[4](10 points)**

- 10: Code review:
  - Is the code properly documented and commented?
  - Error handling - does the code handle errors properly?
  - Using appropriate techniques for each requirement to get the desired results.

## How will the tool be tested?

Minimum checks:

I will access your VM in Proxmox and run the tool within the

[1] **client** machine to scan the **server** machine

[2] "**student**" machine to scan the Proxmox subnet

And will check the required features and quality of output.

-v: Version/about.

-h: Help file/documentation.

-c X: Number of packets to send.

-t: The time spent by the tool.

-p <>: Port/s (single, range)

-<?>: Different options for the different ICMP and TCP scans.

You can choose your own. You may follow the option switches used by hping3 (just a recommendation). Must be able to perform:

- ICMP Scan
- TCP Connect Scan
- Stealth Scans
  - TCP SYN (Half-open) Scan
  - Xmas Scan
  - FIN Scan
  - Null Scan
- TCP ACK Scan

[host/s]: Target hosts, can be a single IP or a range of IPs.

## What to submit?

Submit in Canvas 2 files:

**[File 1]** Short instruction (PDF) on where I can find your tool within your Proxmox VM. Students sharing a VM should have their own respective folders for their tools. This document should also contain the instruction on how to install the tool on a new machine.

**[File 2]** Source code.

**Dates to remember:**

**[2020-11-24] Project Milestone #1: Update on the status of your project.**

**Submit a PDF document containing the following:**

- What name will you give for your tool?
- What programming language are you using for this project?
- Have you started coding?
- In your estimate, what percent have you completed so far?

**[2020-12-15] Project Milestone Final: Submission.**

**Resubmission Practice for this Project:**

I encourage you that you work on your project as early as possible. You can submit your project for early checking and if we need to modify/improve something about the code/output - you will have enough time. This will help assure that you will likely get the maximum number of points. Resubmission practice will on be provided until Dec. 5. All submissions between Dec. 6 to Dec. 15 will be considered as final (will not be entitled to resubmission).