OS Fingerprinting

Using TTL (Time to Live) to fingerprint operating system on a network :

```
66... 39.911470
                           192.168.96.47
                                                                                        23.10.239.251
  66... 39.942093
66... 39.943479
                            23.10.239.251
23.10.239.251
                                                                                        192.168.96.47
192.168.96.47
                                                                                                                      TCP
TCP
  66... 40.164496
                             56:d1:13:db:6e:71
                                                                                        Intel f7:80:7d
                                                                                                                      ARP
                             2409:40c4:2c:2766:3032:4bb7:3f13:a651
  66... 40.201294
                                                                                        2404:6800:4009:82f...
                                                                                                                      TCP
                             2404:6800:4009:82f::2003
                                                                                        2409:40c4:2c:2766
                            192.168.96.47
2405:200:1630:1100::312c:b358
                                                                                        52.35.150.14
2409:40c4:2c:2766
       43 454221
                             2409:40c4:2c:2766:3032:4bb7:3f13:a651
                                                                                        2600:1901:0:38d7::
  Frame 6672: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{612DE01} Ethernet II, Src: Intel_f7:80:7d (9c:29:76:f7:80:7d), Dst: 56:d1:13:db:6e:71 (56:d1:13:db:6e:71) Internet Protocol Version 4, Src: 192.168.96.47, Dst: 23.10.239.251
     0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Time to Live: 128
      Protocol: TCP (6)
Header Checksum: 0x0000 [validation disabled]
[Header checksum status: Unverified]
      Source Address: 192.168.96.47
Destination Address: 23.10.239.251
      [Stream index: 6]
 Transmission Control Protocol, Src Port: 57045, Dst Port: 80, Seq: 461, Ack: 872, Len: 1
```

In this figure we used Wireshark on Windows to see inside a IPv4 traffic from a source to a destination that is 192.168.96.47 to 23.10.239.251, where TTL is 128, which we can infer that 128 TTL is a default value for a Windows OS.

Next, using Nmap to fingerprint operating system on a network :

```
$ nmap -v -0 -Pn 10.10.10.245
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-28 00:44 IST
Initiating Parallel DNS resolution of 1 host. at 00:44
Completed Parallel DNS resolution of 1 host. at 00:44, 0.01s elapsed
Initiating SYN Stealth Scan at 00:44
Scanning 10.10.10.245 [1000 ports]
Discovered open port 22/tcp on 10.10.10.245
Discovered open port 80/tcp on 10.10.10.245
Discovered open port 21/tcp on 10.10.10.245
Completed SYN Stealth Scan at 00:44, 2.21s elapsed (1000 total ports)
Initiating OS detection (try #1) against 10.10.10.245
Nmap scan report for 10.10.10.245
Host is up (0.21s latency)
Not shown: 997 closed top ports (reset)
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
80/tcp open http
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.19
Uptime guess: 29.359 days (since Wed Feb 26 16:06:51 2025)
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=260 (Good luck!)
IP ID Sequence Generation: All zeros
```

Here we can clearly see on IP, 10.10.10.245, we have discovered that there is a Linux Machine running of any version from 4.15 to 5.19, with ports for ftp, ssh, http open on a network.