Academic Year 2023/2024

Internet Performance and Troubleshooting Lab

Politecnico di Torino

Individual Report

Nmap analysis

Group 3

Alessandro Ciullo (s310023)

1 PC and Network Configuration

The following experiments are performed over my local area network. My connection to internet is provided by Tiscali, and the pc used for the laboratory is an HP Laptop connected via Ethernet Cable to the FritzBox Router (modem, switch and wifi access point). The NIC in the pc is a Realtek RTL8111/8168/8411 and the Operating System used is Linux (distribution: Arch 6.6.9).

The network configuration used during the experiments is the following:

Current Speed of eth0	1000Mbps
Current Duplex status	Full
Enabled offloading capabilities	rx-checksumming: on tx-checksumming: on tx-checksum-ipv4: on tx-checksum-ipv6: on gso: off [requested on] generic-receive-offload: on rx-vlan-offload: on tx-vlan-offload: on highdma: on [fixed]

Table 1.1: Configurazione di rete

IP address	192.168.178.36
Default gateway	192.168.178.1
NetMask	255.255.255.0

2 Netscan

The objective of this first part of the experience is the `Host Discovery` over my LAN. To perform an Host Discovery I used the following command:

```
1 nmap -sn 192.168.178.0/24
```

This launch an `nmap` procedure of network scanning over the specified address range without performing a port scan right after.

2.1 Unprivileged mode

By launching this command without root privileges nmap must restrict itself to the functionalities offered by the operating system. In this scenario nmap uses the systemcall `connect` to try perform handshakes with all the host in the address range over the most common ports `80 (http), 443 (https)`. Whatever kind of response to our handshake (both positive and negatives one) is a proof that the investigated host is up and running. By running the command over a LAN, like in my case, packets are only sent to active hosts because the `ARP` protocol does a very reliable host discovery for the operating system and we will see how this will be used to our advantage in the next subsection. If positive response are received nmap closes the connection and a `TCP[RST,ACK]` is sent to the other host, when negative response are received the connection is terminated by the other host and no further action are needed.

At the end of the scansion nmap performs a DNS query for each IP address found online. Since my router hosts a DNS service, as we'll see later, it is able to give human readable names for some of the devices connected in the LAN and put them in the DNS response, than used by nmap.

The terminal output of nmap, with all the host discovered, is shown in the listing 1

2.2 Privileged mode

When we perform the same host discovery over a LAN with root privileges the scansion becomes a lot simpler. Now that nmap has total control over the system, it can construct dummy packets at will. Nmap , as anticipated before, sends ARP request for each host of the LAN and it's able to read the ARP responses (or the ARP table) to find out the active hosts over the network. At the end of the scan is performed a set of DNS queries as in the unprivileged scenario.

The output results are showed in the listing 2. We can immediately notice some differences with user mode scan; the most evident one is the MAC address associated with each host and with it the name of the NIC's producer company (only for known ones). A second difference is the time needed to perform the host discovery: the privileged scan took almost twice as long as the unprivileged one, and this can be caused by the need of computation resources to construct each ARP request.

3 Complete Port Scan

In this section the way nmap operates during a port scan with TCP or UDP and with or without root privileges. The two hosts I have chosen to scan are the router (192.168.178.1) and the Google Home Mini (192.168.178.37).

3.1 Unprivileged default nmap scan

In this section we launch the following command without administrator privileges:

```
1    nmap "IP ADDRESS"
```

The terminal output obtained is showed in listings 3 and 6.

By checking the manual we can deduce that the scan performed is a `TCP connect scan (-sT)`, the default nmap choice when root privileges are not provided. This type of port scan uses the systemcall `connect` to perform an handshake and then closing the connection, if needed, with the host target over the 1000 most common TCP ports (when no port is manually specified), and uses the response to detect the port status:

SYN,ACK The port is open.

RST The port is closed.

No response The port is filtered.

By analysing the `TCP SYN` segments sent we can observe that 1002 packets are captured. The reason behind this 2 more syn segments is the host discovery made before the actual port scan to check that the host is currently active; as seen in the first section when performed in user mode the discovery is made by connecting to the 2 most common port 80 and 443. By observing the graphs 5.1 and 5.4, we can state that the ports are scanned with a random order making the scan, even if only slightly, less conspicuous. We can also notice that no port, besides 80 and 443 for the reason above, is scanned more than one time: the documentation tells us that retransmission are performed only when no answer is received, and since no firewall is placed in my LAN, this never happens. The TCP standard also requires hosts to always respond, even if with a RST flag, to incoming segments, so no doubts are left to the port status.

The only differences between the Router and the Google Home Mini port discovery is the time needed to perform the scan: obviously by being directly connected to my pc with an Ethernet cable the router delay is noticeably less than the IoT device, that is also one hop further over the LAN (the router is also the switch).

3.2 Privileged default nmap scan

By running the same command with root privileges nmap defaults to the `TCP SYN scan`. By having raw packets privileges nmap can writes it's own raw SYN packet and send it to the host target. The response received are treated in the same way discussed before. The difference this time is that the OS doesn't know

about the TCP SYN being sent and will close automatically the connection by sending a `tcp RST` if an, for the OS unrequested, `tcp SYN,ACK` is received. Another difference is that the `source port` used is always the same during the scan: this is due to the nmap `TCP SYN scan` implementation.

The host discovery this time is performed via ARP, so exactly 1000 `TCP SYN segments` are sent (even this time no retrasmission for the same reasons above).

With this scan also the MAC address of the target is provided.

Another difference is in the time needed to perform the scan: this scan is slightly slower because each packet needs to be written by the nmap process directly.

No other difference is noticeable, including port order (still random) and results as shown in listings 4 and 7.

3.3 Privileged UDP nmap scan

For this type of scan no unprivileged version exists, in fact to fully understand the state of a UDP port, reading ICMP packets is necessary, and it's only possible when the process is running as root. This scan is called by the nmap manual `UDP scan`, at the algorithm performs, after a first phase of host discovery, the sending of void UDP packets to the 1000 most common UDP port. The problem with this is that normally no UDP port will respond to an empty packet, and to bypass the issue nmap will send coherently crafted packets for the most popular UDP services (DNS, NTP, DHCP and many other), bypassing the issue. When no response is received some retransmission is performed.

Based on the response nmap will label each port in the following way:

UDP response If nmap receives a response by the interrogated port than that port is `open`.

ICMP Destination Unreachable (type 3) Port Unreachable (code3) The receiving of this message is the only way to be sure that the interrogated port is `closed`. The ICMP message will contain the first 64 bits of the message sent to the host as a way to associate the packets.

No response If no response is received then the port is marked as `open\filtered`. As said before if the payload is not coherently crafted the host can just discard our UDP packet without informing us, but the presence of a firewall would show the same behavior, so we need a double state.

Other ICMP messages If other ICMP messages are received the port is `filtered`.

A very characterizing behavior of this scan is the long time needed to perform the port discovery as we can see on the listings 5 and 8. This is due to the ICMP response rate limitation enforced by a lot of host (in particular Linux ones, like my router). This limitation is usually set to 1 response per second as we can see by the graphs 5.3 and 5.6 that show how to perform a 1000 port scans takes a bit more than 1000 seconds.

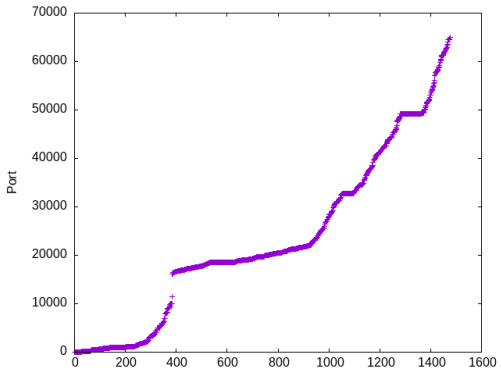

Figure 3.1: reordered UDP scan by port number, performed over google home mini

After plotting the reordered by port number UDP scan shown in figure 3.1 we can notice how a lot of ports are scanned multiple times: this is due to the long wait caused by the ICMP response limitation and

the no obligatory response behavior of UDP that can be perceived like a packet loss, forcing nmap to perform retransmission for good measure.

4 A color-full analysis

I chose as the target the router again, and perfomed against it 2 port scanning already seen `TCP SYN scan (-sS)` and `TCP connect(-sT)` and a new type of port scan `TCP NULL scan (-sN)`, but this time only against the first 300 ports. The graph we generate are a lot more descriptive of the entire scenario because packets are divided by host and by TCP Flag with different icons and colors.

4.1 TCP connect scan

The operation of this type of scanning has already been explained earlier, as the default unprivileged scan made by nmap, but by observing the graph in figure 5.7 something becomes more evident.

The host discovery phase is pretty noticeable when a smaller address range is scanned: even if the scan is limited to the port 300 in this phase nmap still probe for port 443 as shown.

Another thing we can immediately observe is the very balanced distribution of sent and received packets and we will discuss this in the next subsection, in a comparison with the root privileged scan.

4.2 TCP SYN scan

This scan was also discussed earlier, as the default privileged scan made by nmap.

What catches our eye when observing the graph, is the absence of the host discovery phase: the netscan is still there but only ARP traffic is sent and the graphic only traces TCP traffic.

Another thing that immediately catches our attention is the less balanced distribution of the packet sent compared to the TCP connect scan. Packets need to be raw written one by one, consuming computational power. The process, not delegating operations to the operating system or the network card, must alternate between packets generation phases and packets reading phases and the result is a graph divided in bursts of sent and received packets.

In this graph, as mentioned earlier, no `TCP RST-ACK` appears from the nmap host: for the OS the the `TCP SYN-ACK` received are unrequested and the `RST flag` is the standard answer in this scenario.

4.3 TCP NULL scan

The last scan we analyze is the `TCP NULL scan`. This scan, as the name suggests, is performed by sending a TCP segments with all flags set to 0. This scan, together with `TCP FIN scan` and `TCP Xmas scan`, exploit a subtle loophole in the TCP RFC to differentiate between open and closed ports. The RFC states that when a closed port receives a TCP segment with no `RST` flag set, than the receiver must answer with a `TCP RST`, and that if an open port receives a TCP segment with no `SYN` or `ACK` or `RST` flag set the host should just drop the packet and give no answer back.

While the result of this scan can not discern `open` and `filtered` port since in both cases no response is given, the vantages of this scan is that it can sneak through certain non-stateful firewalls and packet filtering routers. Another advantage is that this scan is a little more stealthy than even a SYN scan.

This scan also need root privileges to write raw packets.

By analysing the graph in figure 5.9, we can observe that the sender never set a single flag during the exchange and the only observable packets are the `TCP RST,ACK` received by the target host for each closed port. In fact the number of packets received back is 298, even if 300 ports were scanned, because port 53 and port 80 are open (as known from the previous scans), and packets are dropped when received on those two sockets: the result showed by this scan for those two port is the double state `open\filtered`.

By comparing the graph with the other two we can observe the same pattern of the `TCP SYN scan` (ARP host discovery and packet bursts) and both share as a property the need of root privileges: this further confirms the hypothesis discussed in the last subsection.

5 Appendix

5.1 Netscan

```
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-04 19:51 CET
Nmap scan report for fritz.box (192.168.178.1)
Host is up (0.00047s latency).
Nmap scan report for 192.168.178.28
Host is up (0.0041s latency).
Nmap scan report for 192.168.178.36
Host is up (0.000034s latency).
Nmap scan report for Google-Home-Mini.fritz.box (192.168.178.37)
Host is up (0.0037s latency).
Nmap scan report for 192.168.178.38
Host is up (0.011s latency).
Nmap scan report for 192.168.178.39
Host is up (0.0045s latency).
Nmap scan report for 192.168.178.43
Host is up (0.071s latency).
Nmap scan report for 192.168.178.52
Host is up (0.063s latency).
Nmap scan report for 192.168.178.53
Host is up (0.099s latency).
Nmap scan report for Android.fritz.box (192.168.178.60)
Host is up (0.0027s latency).
Nmap scan report for 192.168.178.61
Host is up (0.030s latency).
Nmap done: 256 IP addresses (11 hosts up) scanned in 2.91 seconds
```

Listing 1: Output of host discovery without root privileges

```
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-04 19:53 CET
Nmap scan report for fritz.box (192.168.178.1)
Host is up (0.00043s latency).
MAC Address: DC:15:C8:CC:57:C8 (AVM Audiovisuelles Marketing und Computersysteme GmbH)
Nmap scan report for 192.168.178.28
Host is up (0.28s latency).
MAC Address: 40:23:43:66:88:65 (Chongqing Fugui Electronics)
Nmap scan report for 192.168.178.37
Host is up (0.093s latency).
MAC Address: 7C:D9:5C:2A:A2:DD (Google)
Nmap scan report for Android-2.fritz.box (192.168.178.38)
Host is up (0.100s latency).
MAC Address: 3C:98:3E:53:3C:54 (Unknown)
Nmap scan report for 192.168.178.39
Host is up (0.070s latency).
MAC Address: B4:07:F9:FB:4C:D3 (Samsung Electro Mechanics)
Nmap scan report for 192.168.178.52
Host is up (0.024s latency).
MAC Address: AE:5D:A1:17:A2:91 (Unknown)
Nmap scan report for Android.fritz.box (192.168.178.60)
Host is up (0.025s latency).
MAC Address: 34:25:BE:08:A0:E8 (Amazon Technologies)
Nmap scan report for ihopeitworks.fritz.box (192.168.178.36)
Host is up.
Nmap done: 256 IP addresses (8 hosts up) scanned in 5.71 seconds
```

Listing 2: Output of host discovery with root privileges

5.2 Port Scan Router

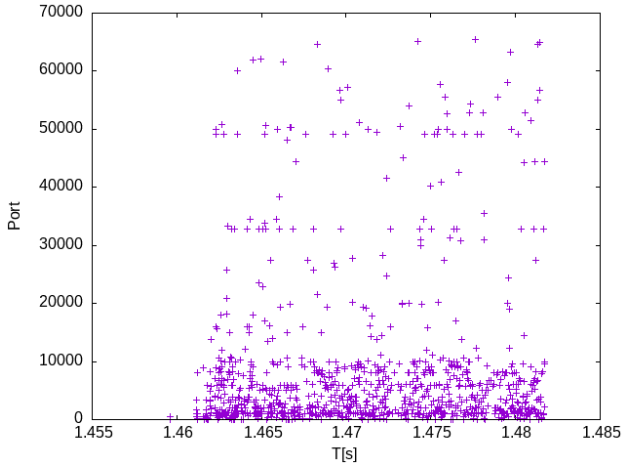

Figure 5.1: nmap router

```
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-04 19:58 CET
Nmap scan report for fritz.box (192.168.178.1)
Host is up (0.0022s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
443/tcp  open  https
5060/tcp open  sip
8089/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.05 seconds
```

Listing 3: Output of nmap router without root privileges

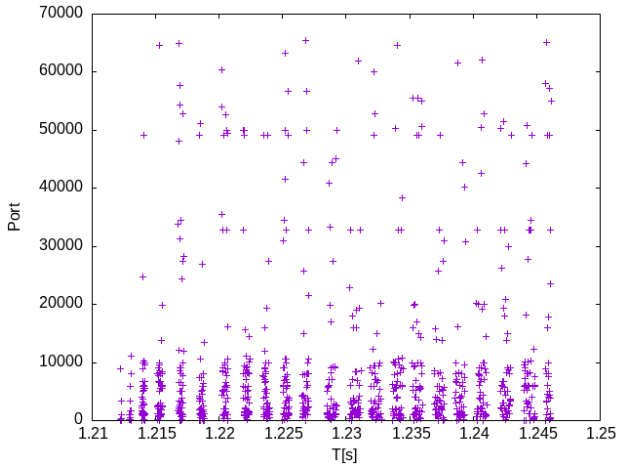

Figure 5.2: sudo nmap router

```
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-04 19:59 CET
Nmap scan report for fritz.box (192.168.178.1)
```

```
Host is up (0.00060s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
443/tcp  open  https
5060/tcp open  sip
8089/tcp open  unknown
MAC Address: DC:15:C8:CC:57:C8 (AVM Audiovisuelles Marketing und Computersysteme GmbH)

Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds
```

Listing 4: Output of nmap router with root privileges

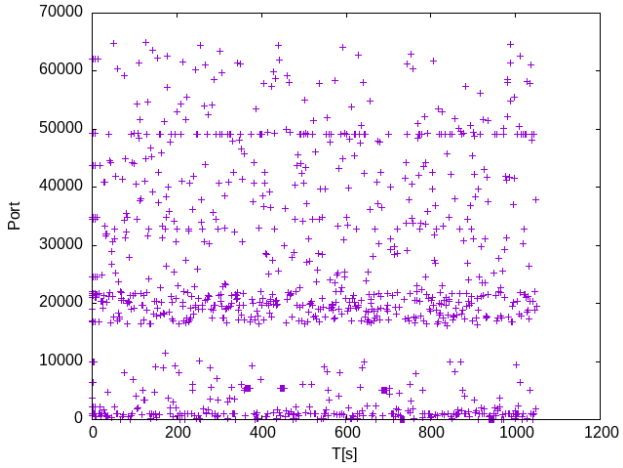

Figure 5.3: sudo nmap-sU router

```
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-05 19:50 CET
Nmap scan report for fritz.box (192.168.178.1)
Host is up (0.00046s latency).
Not shown: 991 closed udp ports (port-unreach)
PORT     STATE         SERVICE
9/udp    open|filtered discard
53/udp   open          domain
67/udp   open|filtered dhcps
123/udp  open          ntp
1900/udp open          upnp
5060/udp open|filtered sip
5351/udp open          nat-pmp
5353/udp open|filtered zeroconf
5355/udp open|filtered llmnr
MAC Address: DC:15:C8:CC:57:C8 (AVM Audiovisuelles Marketing und Computersysteme GmbH)

Nmap done: 1 IP address (1 host up) scanned in 1049.29 seconds
```

Listing 5: Output of nmap -sU router with root privileges

5.3 Port Scan Google Home Mini

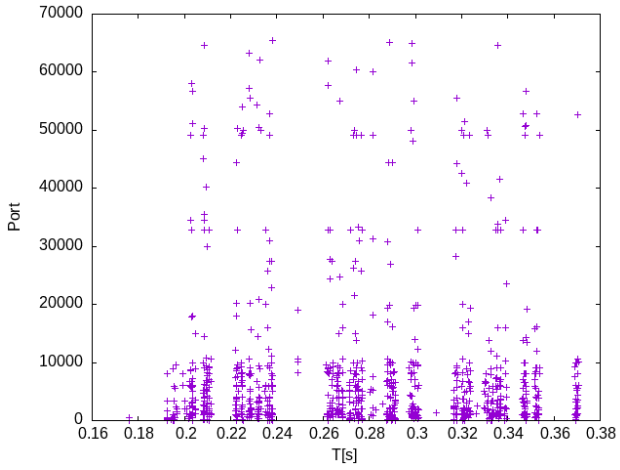

Figure 5.4: nmap ghm

```
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-05 23:37 CET
Nmap scan report for Google-Home-Mini.fritz.box (192.168.178.37)
Host is up (0.0055s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT      STATE SERVICE
8008/tcp  open  http
8009/tcp  open  ajp13
8443/tcp  open  https-alt
9000/tcp  open  cslistener
10001/tcp open  scp-config

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
```

Listing 6: Output of nmap ghm without root privileges

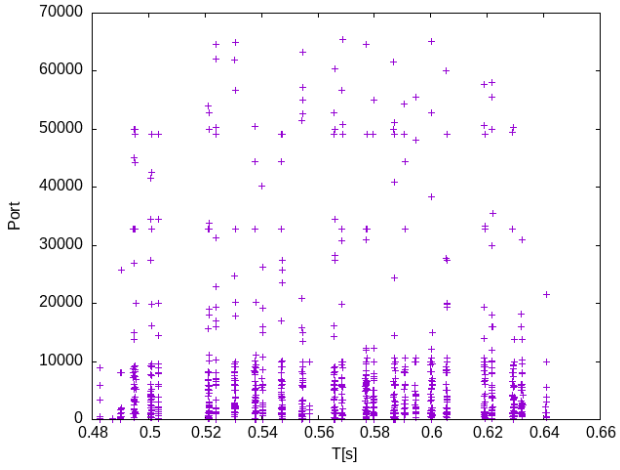

Figure 5.5: sudo nmap ghm

```
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-05 23:37 CET
Nmap scan report for Google-Home-Mini.fritz.box (192.168.178.37)
```

```
Host is up (0.0040s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE
8008/tcp  open  http
8009/tcp  open  ajp13
8443/tcp  open  https-alt
9000/tcp  open  cslistener
10001/tcp open  scp-config
MAC Address: 7C:D9:5C:2A:A2:DD (Google)

Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds
```

Listing 7: Output of nmap ghm with root privileges

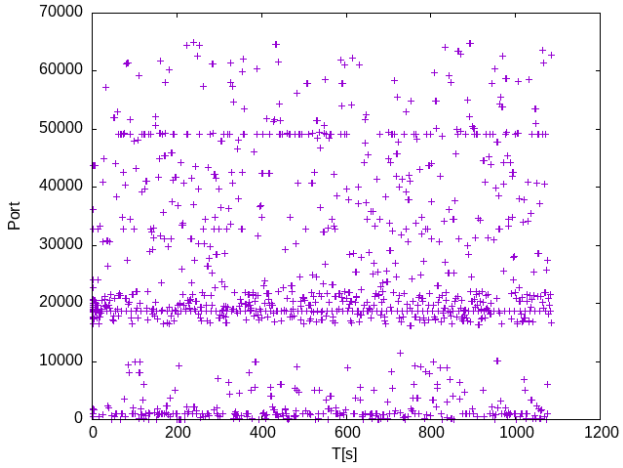

Figure 5.6: sudo nmap-sU ghm

```
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-05 20:19 CET
Nmap scan report for Google-Home-Mini.fritz.box (192.168.178.37)
Host is up (0.0037s latency).
Not shown: 999 closed udp ports (port-unreach)
PORT     STATE SERVICE
5353/udp open  zeroconf
MAC Address: 7C:D9:5C:2A:A2:DD (Google)

Nmap done: 1 IP address (1 host up) scanned in 1083.54 seconds
```

Listing 8: Output of nmap -sU ghm with root privileges

5.4 A colorfull analysis

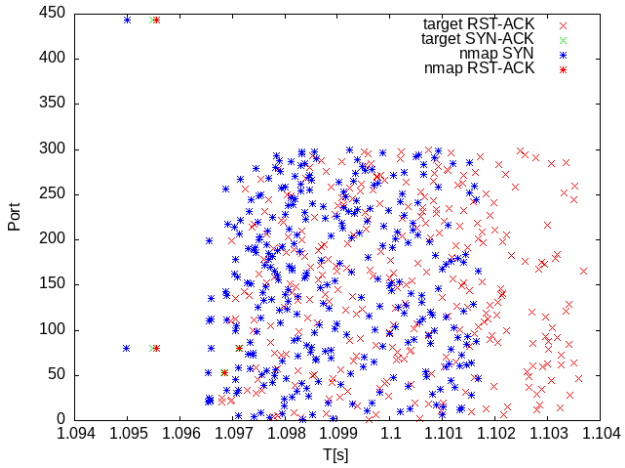

Figure 5.7: nmap -sT -p 1-300 against the router

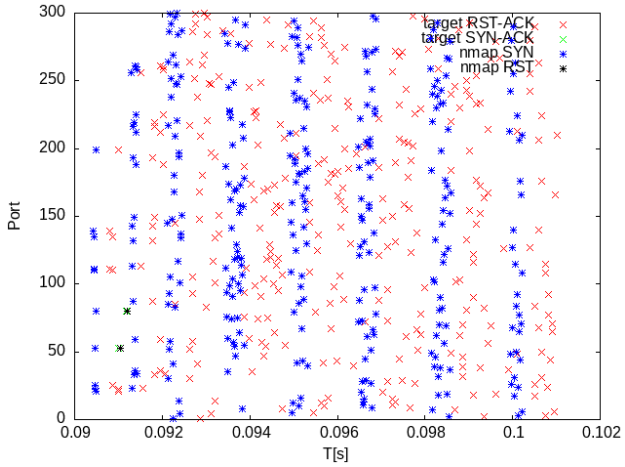

Figure 5.8: sudo nmap -sS -p 1-300 against the router

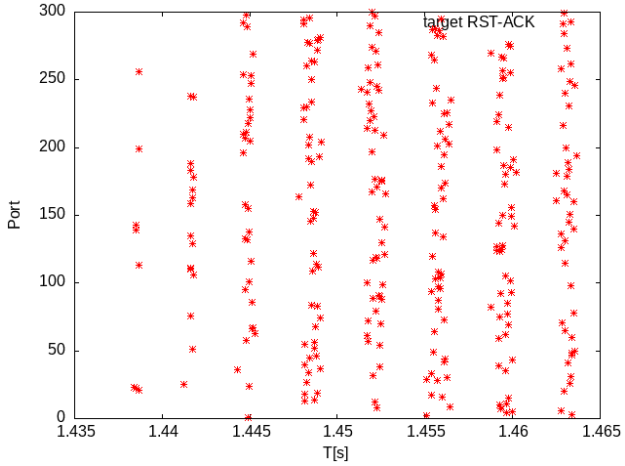

Figure 5.9: sudo nmap -sN -p 1-300 against the router