SYN flood exercise

Offensive Technologies 2021
Matteo Franzil <matteo.franzil@studenti.unitn.it>

October 19, 2021

1 Solution

As usual, I logged in with SSH: `ssh attacker.franzil-nmap.offtech`
This is the map of network:

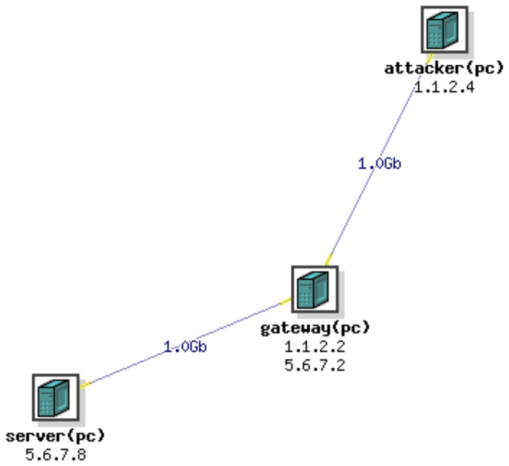

Figure 1 The network.

1.1 Reconnaisance

We already know that the range is `5.6.7.0/24`. So we don't have to scan randomly. Our first request is to do an ICMP scan. The bash script is the following:

```
#!/bin/bash

i=1
end=254

while [ $i -le $end ]; do
  ping -c 3 5.6.7.$i
  i=$(($i+1))
done
```

Code 1 Bash script for pinging all hosts.

Since it's a `/24` subnet we can just iterate from 1 to 254 (we skip 0 and 255 for the usual network reasons). Either way, nobody answered to our pings - probably the hosts have disabled ICMP on their end.

With nmap, we can blast our way through with `sudo nmap -sn -PE 5.6.7.0/24`. Still, nobody cared about our pings.

We then tried to do a TCP ACK ping: `sudo nmap -sn -PA 5.6.7.0/24`. This type of ping sends an empty TCP packet with the ACK flag to port 80 (by default, but it's changeable). This usually allows to ignore firewalls. This time, `5.6.7.8` answered!

```
otech2af@attacker:~$ sudo nmap -sn -PA 5.6.7.0/24 --traceroute

Starting Nmap 7.60 ( https://nmap.org ) at 2021-10-19 13:28 PDT
Nmap scan report for server-link1 (5.6.7.8)
Host is up (0.00027s latency).

TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   0.11 ms gateway-lan1 (1.1.2.2)
2   0.25 ms server-link1 (5.6.7.8)

Nmap done: 256 IP addresses (1 host up) scanned in 1.44 seconds
```

Figure 2 Executing a TCP ACK ping.

Finally, we had to probe the network with different probes. A quick visit to the nmap manual yields some useful information:

- PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
- PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
- PO[protocol list]: IP Protocol Ping

To get started, we launched a multi-probe scan: `sudo nmap -sn 5.6.7.0/24`. This does an ICMP echo request, a TCP SYN packet to port 443, a TCP ACK packet to port 80, and an ICMP timestamp request. The result do remain still the same, though. In other cases, such scans can be very useful in order to trick firewalls or avoid protections. For exampke, ICMP PP and PM scans are very useful if PE is blocked - for example, if the server blocked ICMP ping responses.

1.2 Port scanning

Our first request is to perform a TCP half open scan. We can do it with `sudo nmap -sS 5.6.7.8`.

First, we launch it with no flags, then we set the port flag to `-p 1-1500`, in order to scan the first 1500 ports. When no flag is set, nmap defaults to searching on the top 1000 ports - where top refers to the most used, using a nmap-specific metric obtained with experimental results.

So, we launch another scan with `sudo nmap -sS -p 1-1500 5.6.7.8`.

```
otech2af@attacker:~$ sudo nmap -sS 5.6.7.8

Starting Nmap 7.60 ( https://nmap.org ) at 2021-10-19 13:30 PDT
Nmap scan report for server-link1 (5.6.7.8)
Host is up (0.00024s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 16.68 seconds
```

Figure 3 Executing a TCP half open scan with no flags.

The difference is evident: on our second scan, there is port 1212/tcp open, which is apparently not on the top 1000 ports (being quite an obscure one actually). Indeed, doing:

```
otech2af@attacker:~$ sudo nmap -sS -p 1-1500 5.6.7.8

Starting Nmap 7.60 ( https://nmap.org ) at 2021-10-19 13:30 PDT
Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Ping Scan Timing: About 100.00% done; ETC: 13:30 (0:00:00 remaining)
Nmap scan report for server-link1 (5.6.7.8)
Host is up (0.00024s latency).
Not shown: 1497 filtered ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
1212/tcp open  lupa

Nmap done: 1 IP address (1 host up) scanned in 14.05 seconds
```

Figure 4 Executing a TCP half open scan with the first 1500 ports.

```
sort -r -k3 /usr/share/nmap/nmap-services | head -n 1000 | grep " 1212"
```

confirms our suspects.

Next on the list, is performing a XMas scan. It sets the FIN, PSH, and URG flags, lighting the packet up like a Christmas tree. We gest started by launching `sudo nmap -sX 5.6.7.8`.

```
otech2af@attacker:~$ sudo nmap -sX 5.6.7.8

Starting Nmap 7.60 ( https://nmap.org ) at 2021-10-19 13:34 PDT
Nmap scan report for server-link1 (5.6.7.8)
Host is up (0.00023s latency).
Not shown: 997 closed ports
PORT    STATE         SERVICE
22/tcp  open|filtered ssh
80/tcp  open|filtered http
111/tcp open|filtered rpcbind

Nmap done: 1 IP address (1 host up) scanned in 96.69 seconds
```

Figure 5 Executing a XMas scan.

From the results, it looks like we found another `open|filtered` one, rpcbind @ 111/filtered. But why are ports being shown as `open|filtered`?.

As a clarification, from the standpoint of this scan, if a RST packet is received, the port is considered closed. The port is marked surely filtered if we receive an ICMP error. Finally, no response means it is open—filtered.

But now, we have no way of distinguishing a filtered port from an open one - we finished our possible cases - and since open ports are inferred via no responses being generated, we have no way of actually distinguishing them and we have to do additional scans to confirm our findings.

Now, we are required to perform a TCP ACK scan. We can do it with `sudo nmap -sA 5.6.7.8`.

```
otech2af@attacker:~$ sudo nmap -sA --top-ports 3328 5.6.7.8

Starting Nmap 7.60 ( https://nmap.org ) at 2021-10-19 13:39 PDT
Nmap scan report for server-link1 (5.6.7.8)
Host is up (0.00016s latency).
All 3328 scanned ports on server-link1 (5.6.7.8) are unfiltered

Nmap done: 1 IP address (1 host up) scanned in 3.47 seconds
```

Figure 6 Executing a TCP ACK scan.

All ports seem to be unfiltered - indeed, this is the use of this type of scan: verifying the capabilities of the firewall. This allows us to assume that the firewall is stateless (since the ACKs did correctly go through).

1.3 OS detection

Finally, we run a scan to determine both the O.S. version running on the host and the versions of the services:

```
sudo nmap -A -O -sV -sC --traceroute 5.6.7.8
```

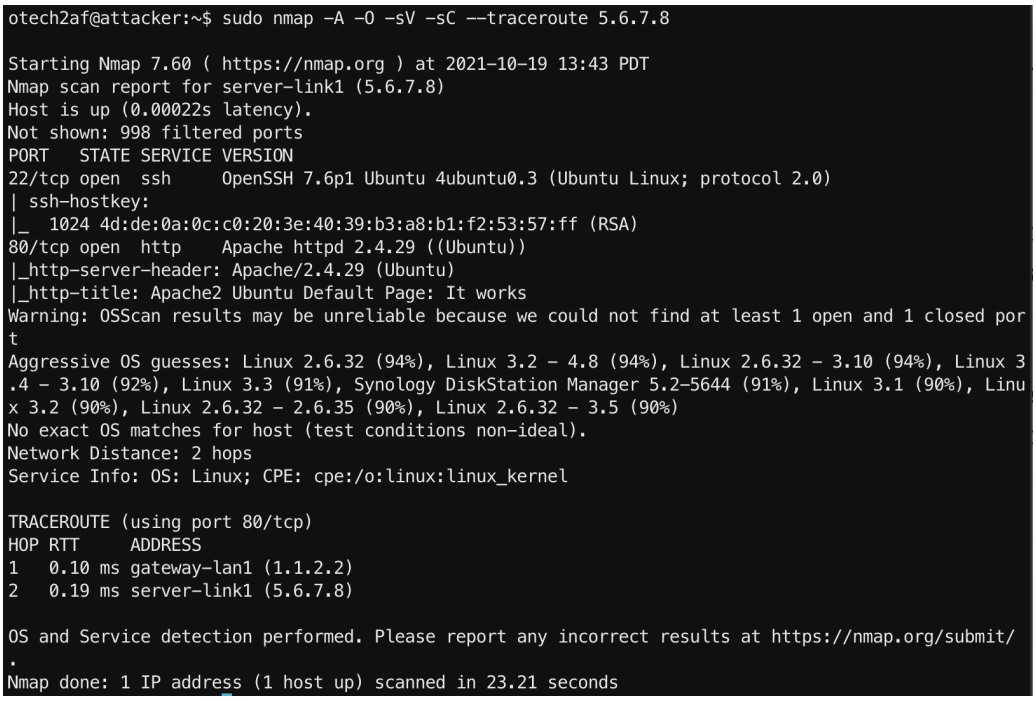

Figure 7 Executing the final OS discovery scan.

From this large message we can gather a lot of information. First, we learn the versions of SSH and Apache (quite an old one). Then, we learn almost surely that our target is running some version of Ubuntu, although the Nmap guesses are wild (the server is actually running Ubuntu 18.04, which is quite newer than the predicted Linux 2.6.32 kernel, whose release dates back to 2012).

Finally, we learn that our path to the target goes through an additional machine, called `gateway`.