Senko Salihbegovic - Week 8 Homework

Phase 1:

The IP addresses (CIDR) to scan are:

15.199.95.91/28
15.199.94.91/28
11.199.158.91/28
167.172.144.11/32
11.199.141.91/28

The command used is “fping” and looks like this:

fping -g 15.199.95.91/28
fping -g 15.199.94.91/28
fping -g 11.199.158.91/28
fping -g 167.172.144.11/32
fping -g 11.199.141.91/28

Using -g runs fping through all of the IP addresses automatically without having to type the whole range of IP addresses manually

After running those commands I got only one live ip which gave a response and that is:

167.172.144.11

This is a problem and vulnerability since Rockstar doesn’t want any request responses.

After running the command fping -s -g 167.172.144.11/32 we get this summary:
167.172.144.11 is alive

    1 targets
    1 alive
    0 unreachable
    0 unknown addresses

    0 timeouts (waiting for response)
    1 ICMP Echos sent
    1 ICMP Echo Replies received
    0 other ICMP received

 237 ms (min round trip time)

237 ms (avg round trip time)
237 ms (max round trip time)
0.237 sec (elapsed real time)

Mitigation step would be to restrict allowing ICMP echo requests against this IP address.

This occurred on the network layer (layer 3).

Phase 2

In this phase I'm going to run a SYN SCAN against 167.172.144.11 using nmap.

The command I ran:

sudo nmap -sS 167.172.144.11 (This scan type requires root privileges)

This is the result that came back:

Nmap scan report for 167.172.144.11
Host is up (0.24s latency).
Not shown: 995 closed ports
PORT STATE SERVICE
22/tcp open ssh
135/tcp filtered msrpc
139/tcp filtered netbios-ssn
593/tcp filtered http-rpc-epmap
646/tcp filtered ldp

Nmap done: 1 IP address (1 host up) scanned in 485.73 seconds

As we can see from this result, port 22 is open and ports 135, 139, 593 and 646 are filtered. When ports are filtered it means that only specific machines are allowed to connect to those ports.

The port 22 (SSH) is open and it shouldn't be which makes this a very dangerous vulnerability. An attacker can freely connect to the port 22 and crack the password with no issues (especially if the password is simple).

The recommended step here is to close the port 22 or filter it to only accept the connection from machines approved by Rockstar.

Nmap mainly uses the transport layer (layer 4).

Phase 3

In this step I'm going to ssh to the ip address. I have been provided with username: jimi and password: hendrix.

Command to ssh is:

sudo ssh jimi@167.172.144.11 -p 22

In the assignment we need to figure out why Rockstar can't access the domain rollingstone.com in their Hollywood office.
They have reported that when trying to access rollingstone.com they get redirected to a different, unusual website which tells us that hosts file has probably been tampered with.

To check this I have run the command:

cat /etc/hosts

That returned this result:

```
# Your system has configured 'manage_etc_hosts' as True.
# As a result, if you wish for changes to this file to persist
# then you will need to either
# a.) make changes to the master file in /etc/cloud/templates/hosts.tmpl
# b.) change or remove the value of 'manage_etc_hosts' in
#     /etc/cloud/cloud.cfg or cloud-config from user-data
#
127.0.1.1 GTscavengerHunt.localdomain GTscavengerHunt
127.0.0.1 localhost
98.137.246.8 rollingstone.com

oooooooollowing lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

We can see from this result that when trying to access rollingstone.com, the request would be redirected to 98.137.246.8.

Now we need to figure out where this IP leads. To do that I have terminated the ssh session and used the command nslookup to find the domain associated with the strange IP address.

The command looks like this:

nslookup 98.137.246.8

The result that came back is this:

8.246.137.98.in-addr.arpa name = unknown.yahoo.com.

Authoritative answers can be found from:

But if we run nslookup on the rollingstone.com we get this result:

Server: 8.8.8.8
Address: 8.8.8.8#53

Non-authoritative answer:
Name: rollingstone.com
Address: 151.101.64.69
Name: rollingstone.com
Address: 151.101.0.69
Name: rollingstone.com
Address: 151.101.128.69
Name: rollingstone.com
Address: 151.101.192.69

From these results we can see that an attacker managed to get in through the ssh port and change the hosts file to redirect traffic from rollingstone.com to another domain. Once the attacker is in, they have the whole system in their hands. Definitely a major vulnerability.

Mitigation recommendation stays the same as in Phase 2. One additional thing can be done and that is to do a regular backup of the hosts file (or just have a copy of the original file) so it can be restored in case an attack like this happens.

This is a perfect example of taking advantage of network layer (layer 3).
SSH is an application layer protocol, which is the 7th layer of the OSI model.

Phase 4

In this phase we need to ssh back onto the open IP and find a note that the “hacker” left. After some digging I have found the file called packetcaptureinfo.txt in the /etc folder. After that I opened the file with the “less” command (less packetcaptureinfo.txt) and this link was written inside:

https://drive.google.com/file/d/1ic-CFFGrbruloYrWaw3PvT71elTkh3eF/view?usp=sharing

After opening the link I've found and downloaded secretlogs.pcapng (according to the instructions, these are packets that were captured from the activity in the Hollywood Office) which I have opened in Wireshark to inspect and see if there was any suspicious activity that could be attributed to an attacker.

After analyzing the packets I have found that the IP 192.168.47.200 is pointing to two different mac addresses (packets 4 and 5):

ARP 192.168.47.200 is at 00:0c:29:0f:71:a3
ARP 192.168.47.200 is at 00:0c:29:1d:b3:b1

This would probably be the result of the attacker redirecting traffic or backdooring into the server but there is one issue. The date of the ARP packets say that they are from 2014. Either the attacker spoofed these packets or this incident has no connection whatsoever with the current one.

After further analyzing the packets I have found a HTTP POST packet (packet 16) in which we have evidence of the attacker contacting Got The Blues Corp to sell login credentials for the port 22 (SSH) for 1 million dollars.

I have copied just a piece of the HTML Form:

Hypertext Transfer Protocol
POST

Form item: "0<text>" = "Mr Hacker"
Form item: "0<label>" = "Name"
Form item: "1<text>" = "Hacker@rockstarcorp.com"
Form item: "1<label>" = "Email"
Form item: "2<text>" = ""
Form item: "2<label>" = "Phone"
Form item: "3<textarea>" = "Hi Got The Blues Corp! This is a hacker that works at Rock Star Corp. Rock Star has left port 22, SSH open if you want to hack in. For 1 Milliion Dollars I will provide you the user and password!"
Form item: "3<label>" = "Message"
Form item: "redirect" = "http://www.gottheblues.yolasite.com/contact-us.php?formI660593e583e747f1a91a77ad0d3195e3Posted=true"
Form item: "locale" = "en"
Form item: "redirect_fail" = "http://www.gottheblues.yolasite.com/contact-us.php?formI660593e583e747f1a91a77ad0d3195e3Posted=false"
Form item: "form_name" = ""
Form item: "site_name" = "GottheBlues"

From this we can also see that the “hacker” is also an employee of RockStar Corp. Instead of acting for the best interest of the company and closing the port or alerting the security team of the vulnerability, the “hacker” decided to use the ssh vulnerability for personal gain. This is also a big vulnerability.

As in previous phases, the mitigation steps include closing/filtering port 22 as well as actively monitoring the traffic on their network. If they do, they will notice this event taking place. Also, in general, checking file integrity is a good idea (like /etc/hosts file).

ARP works between layers 2 and 3 (MAC - layer 2, IP - layer 3).
HTTP is application layer (layer 7).