CTF REPORT 3
EPUNA STANLEY ONYEDIKA
STUDENT ID: 45731365
31st May 2020

ABSTRACT

Capture the Flag (CTF) competitions allow students to learn cybersecurity skills in a fun and engaging way. It is an effective platform to increase students' interest in cybersecurity and prepare them for defending against real cyber attackers. For this particular CTF, we were required to attack the 172.31.0.10/24 network range for vulnerabilities using an open SSH port

Introduction
The CTF 3 commenced on the 26th of May 2020, 10:00 am with its finish time set to noon. We were provided with 6 flags to complete. We had to input the flag results on the mobi.science.mq.edu.au server to correlate our result.
This CTF challenge was a Network CTF which comprises of 6 flags.
This report contains some technical writeups of the CTF 3 task. It's a well written technical report.

Connection Methods:

1. The entry point to the network was through Macquarie University VPN
2. The website where the flags would be submitted: https://moby.science.mq.edu.au
3. The server where we would find our Task and Flags: https://10.46.225.208

Note: it is stated clearly in the Moby website that in other to get the flags, this network portion: 172.31.0.10/24 needs to be scanned for vulnerabilities.

We were given the following to connect using the Secure Shell (SSH) remote login to get the CTF tasks

1. Username: Alice
2. Server Address: 10.46.225.208
3. Password: greensand34

In this scenario, we used Kali interface to remotely connect to the server

Using SSH
ssh: ssh user@remotehostip -p xxx
Alice – Username
10.46.225.208 – Remote Host Ip

for me to create a ssh command
type the following command in the terminal:
ssh alice@10.46.225.208
password: greensand34

To get the vulnerabilities, we need to first connect to 10.46.225.208 using ssh.
So, in the terminal:
ssh alice@10.46.225.208
password: greensand34
now we are in the network, its time to find vulnerabilities in network range.

nmap -sV -Pn 172.31.0.0/24 -p22,80,443,8080,8443
I scanned for well-known ports: 80. 443, 8080. 8443

Nmap scan report for team16 (172.31.0.208)
Starting Nmap 7.60 (https://nmap.org) at 2020-05-26 01:46 UTC
Nmap scan report for team16_buzzer.team16_back (172.31.0.10)
Host is up (0.000017s latency).
PORT STATE SERVICE VERSION
22/tcp closed ssh
80/tcp closed http

443/tcp closed https
8080/tcp closed http-proxy
8443/tcp closed https-alt
MAC Address: 02:42:9E:1A:50:1F (Unknown)

Nmap scan report for team16_tftp.team16_back (172.31.0.30)
Host is up (0.000027s latency).
PORT   STATE SERVICE   VERSION
22/tcp   closed ssh
80/tcp   closed http
443/tcp closed https
8080/tcp closed http-proxy
8443/tcp closed https-alt
MAC Address: 02:42:78:45:C1:BF (Unknown)

Nmap scan report for team16_dns.team16_back (172.31.0.40)
Host is up (-0.045s latency).
PORT   STATE SERVICE   VERSION
22/tcp   closed ssh
80/tcp   closed http
443/tcp closed https
8080/tcp closed http-proxy
8443/tcp closed https-alt
MAC Address: 02:42:26:5C:48:7C (Unknown)

Nmap scan report for team16_haystack.team16_back (172.31.0.50)
Host is up (-0.051s latency).
PORT   STATE SERVICE   VERSION
22/tcp   closed ssh
80/tcp   closed http
443/tcp closed https
8080/tcp closed http-proxy
8443/tcp closed https-alt
MAC Address: 02:42:5F:C8:99:BF (Unknown)

Nmap scan report for team16_pong.team16_back (172.31.0.60)
Host is up (0.000019s latency).
PORT   STATE SERVICE   VERSION
22/tcp   closed ssh
80/tcp   closed http
443/tcp closed https
8080/tcp closed http-proxy
8443/tcp closed https-alt
MAC Address: 02:42:39:75:2E:72 (Unknown)

Nmap scan report for team16 (172.31.0.208)
Host is up (0.000022s latency).
PORT   STATE SERVICE   VERSION
22/tcp   open   ssh   OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp   closed http
443/tcp closed https

8080/tcp closed http-proxy
8443/tcp closed https-alt
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
Host is up (0.000022s latency).

Nmap scan report for team16 (172.31.0.208)
Network 172.31.0.208 was the only open ssh port from al the networks in the 172.31.0.0/24 range. So, it is best we used the network.

Flag 2- X11(Mark the spot)
After scanning with nmap and found vulnerabilities in 172.31.0.208 network, the next step is to access the vulnerabilities
ssh 172.31.0.208
password: greensand34
ls -a

The above command prints the directories available:
.Xauthority .bash_history .bashrc .profile enum4linux.pl offsec.txt
...Xdefaults .bash_logout .cache .ssh offsec.ctf x11flag
We should check each directory and files to look for good information related to the task. one seems to get our attention (x11flag). let us view the properties of the file
File x11flag
It is shown that it is a shared object (eb 64-bit lsb). It seems to be an encrypted file. We viewed the file
cat x11flag
this had some encryption information but was not that readable. I read some lines and found that it had something to do with SHA1 encryption. I researched some way to decrypt the file but got some errors, but I am quite sure this x11flag should be decrypted before getting the flag.
Also, I found another file that could be of great interest and that is enum4linux, but I could not deduce what it means, so I researched it and found out that **enum4linux** is used to extract information from Windows and samba hosts. So, it is used to get information from a windows server machine. I was trying to deduce if this enum4linux file had anything to do with the x11flag but could not find anything.

Flag 4 – Respect my Authority
Here, we used the vulnerable network 172.31.0.208 using ssh
ssh 172.31.0.208
password: greensand34
ls -a

The above command prints the directories available:
.Xauthority .bash_history .bashrc .profile enum4linux.pl offsec.txt
.. .Xdefaults .bash_logout .cache .ssh offsec.ctf x11flag
I noticed two files which could be of interest, offsec.txt, and offsec.ctf.
Tried to view the file properties
Ls -al
-rw-r--r-- 1 alice alice 0 May 26 01:30 offsec.ctf
-rw-r--r-- 1 alice alice 0 May 26 01:31 offsec.txt
After viewing the properties, I found it was an empty file with 0 bytes.
I also tried opening the files to be sure it is empty, and it really was.
cat offsec.ctf

cat offsec.txt
I also tried concatenating both of the files and exporting them to a file
cat offsec.ctf offsec.txt > filezip
I opened the zip file and it was still empty
Cat filezip

Reflection

During the final minutes of the CTF task, I figured out that Wireshark would have been used to get some flags since Wireshark deals with data packets coming from a network. Wireshark would have benefited us so well in this task since we did not get any flags. I felt that flag2, flag4, flag5 consist of details related to the spoofing of packets. I researched some CTF challenges that had to deal with Wireshark but all of them had to deal with a pcap file that we could open on Wireshark to determine the packet flow. I paid too much attention related to the pcap file. But this our task does not need the pcap file since there is already a communication going on the Linux terminal. What was needed was to just turn on the Wireshark application and click on the ssh remote capture: sshdump.

I learned so much in this CTF even though I did not get any flags. It was an interacting session, we cracked our brains to figure out something, at the late hour we realized that most of these flags could be gotten using Wireshark application.

Conclusion

This capture the flag task provides valuable experience with network exploitation and also different approaches needed to solve the problems. We as a group had difficulties in getting the flags but I am happy that the research I did before the CTF helped me in reasoning deep.

In this CTF there were not any ethical implications since it was hosted by Macquarie University. If it was carried out in a real situation there could be some legal breach and repercussion. I am glad I could be able to practice this CTF challenge which could help me build up myself in the task ahead.