Findings for Capture the Flag (CTF) Exercise 3

Mohammed Sadiq Abuwala: 45921407
Team Number: 16

02 June 2020

Abstract

The capture the Flag (CTF) exercise is a security hacking game organized for COMP 6320 (Offensive Security). For the purpose of this particular exercise, we targeted an application with an open SSH port and we describe our experience and results here in detail. Every flag was concealed in such a way that it required some sort of operations before they could be revealed. Instead of dividing flags between team members, our team collectively decided to approach the exercise on a best effort basis.

Introduction

Hardware and Software used during the exercise:

Hardware:
Processor: Intel ® Core ™ i7-8550U CPU @ 1.80GHz
Installed memory (RAM): 8.00 GB
Primary Storage: 512GB SSD M.2 2280 PCIe NVMe
Note: Hardware specifications above may not be relevant as we are using a Virtual Machine (VM) as noted below.

Software:
Oracle virtual Machine running Kali Linux with following configuration:
System Base Memory (RAM): 2048MB
Processors: 2

In total, this exercise consisted of capturing six flags over a time duration of two hours. The **connection and target details** are provided below:

1. VPN Access via the Virtual Intranet Access application.
2. Moby dashboard to enter flags and view scoreboard.
3. Login credentials:
   a. Username: alice@10.46.255.208
   b. Password: greensand34

Now, let us probe the given target IP address further to find further network properties like open ports by using the network scanning utility **nmap**. This shows the following open ports:

- 22/tcp   open   ssh
  80/tcp   closed http
  443/tcp closed https
  8080/tcp closed http-proxy
  8443/tcp closed https-alt

This confirms that the only open port is SSH. Hence, we can now try connecting to the address by inputting the login credentials provided into the Kali terminal via the **ssh** command:

- ssh alice@10.46.255.208

Upon entering the above command, the server responds back asking for the password. Upon entering the given password, connection to the target IP address is successful. In the sections below, Flags have been recorded in the order that they were discovered by the individual.

Flag: X11-Mark the spot

Now that I have used ***nmap*** to find open ports and upon using that information to successfully login to the target IP address, let us see what directories are available by using the ***ls*** command:

- 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

Let us check each directory and files to look for any interesting information. One file seems to be of particular interest - ***x11flag***. Let us try to view file properties:

- file x11flag

The server responds that it is an ***eb 64 bit lsb shared object***. On further research, it seems to be an encrypted file. Let us try to view file contents to gather further information:

- cat x11flag

There seems to be some encrypted information in the file. It is mostly unreadable. The readable text does mention SHA1 encryption. Perhaps we could find a way to decrypt this file. Another interesting file can also be seen within the available directories - ***enum4linux***. On further research, we have deduced that it is an enumeration tool for windows and samba systems. Finally, we made attempts to find ways to use the enum41linux.pl file to decrypt the information provided in the ***x11flag*** file but no further result was achieved.

Flag: Respect my Authority

As we were given a range of IP addresses that could be in use, we first tried to find which IP addresses were in use. This was done with the following commands:

- /sbin/ifconfig -a

The above command prints out the already given IP address (10.46.255.208) and an additional interesting IP address : ***172.31.0.208***

We try to confirm that there are no additional IP addresses in use by doing further probing using the ***nmap*** and ***netstat*** command:

- netstat -a
- nmap -nP

Results from both commands above print out similar results which show that ***172.31.0.208*** is the only other IP address being used. We try to use the ***ssh*** command to gain access to this network address:

- ssh alice@123.31.0.208

Upon inputting the above command, the server responds back requesting a password. We enter the same password (greensand34) that was provided to us earlier to login to 10.46.255.208. Login was successful. Again, we try using probing methods when finding vulnerabilities while accessing an ssh port. Let us first check directories:

- ls -al

Upon investigating the results, we come upon interesting files **offsec.ctf** and **offsec.ctf**:

```
- 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
```

Next, we try to view the contents of the files (offsec.ctf, offsec.txt) using the ***cat*** commands:

- cat offsec.ctf
- cat offsec.txt

Both files are empty. Let us try to view file properties to see if we can gain any more information:

- file offsec.ctf
- file offsec.txt

Results show that both files are 0 bytes. No further results could be achieved.

Summary of CTF Results

FLAG NUMBER	**RESULTS**	**SOLVED BY / ATTEMPTED BY**
FLAG 1	UNSOLVED	N/A
FLAG 2	UNSOLVED	N/A
FLAG 3	UNSOLVED	N/A
FLAG 4	UNSOLVED	N/A
FLAG 5	UNSOLVED	N/A
FLAG 6	UNSOLVED	N/A

Conclusion

In the first CTF, we conducted penetration tests on a linux server through its open SSH port. In the second CTF, we targeted an open HTTP port. In this CTF, we were able to expand our knowledge further by probing open ports through the use of packet capture and investigation via wireshark. This shows that it is also important to protect and maintain how data is being communicated from the server to the client as data packets can also be captured and leak critical information.

Personal Reflection

On further reflection, WE should have made better use of Wireshark to investigate data packets being captured. It would have most likely provided us with the necessary pieces of information required to capture the flags. While I had attempted several capture the flag exercises publicly available online, all of them involved downloading a pcap file and opening it via the Wireshark application for investigation. Hence, I was expecting to find a pcap file somewhere throughout the exercise so that I could investigate packets through the Wireshark application. At the end of the exercise, I realized that since we already had access to the system and were communicating with the system in real time, we did not need a pcap file to investigate packets and simply needed to turn on the wireshark application to view packets being captured.

Furthermore, although I had a running VPN configuration throughout every CTF exercise, it would not connect during this particular exercise. By the time this issue was successfully fixed with the support of the tutor, there were only 1.3 hours of the total 2 hours remaining. Considering that this particular CTF was considerably more difficult than previous CTFs, a further reduction in available time considerably impacted how much I could contribute to this CTF.

Finally, these CTF exercises have induced our interest in learning. We would consider this CTF exercise as an interactive and interesting tool in enabling us to further our knowledge in penetration testing. Additionally, this exercise helped us understand the concepts at a more practical level by allowing us to apply the concepts learned.

Disclaimer

It is important to note that this CTF was conducted in a sandboxed environment solely for educational purposes. All team members have signed an Acceptable Usage agreement and understand the responsibilities and consequences while conducting penetration tests.