Advanced Linux stuff

Zack Orndorff
Previous edition: Chris and Zack

Agenda

- Linux Permissions review + advanced - I can’t let you do that Dave
- Symlinks!
- Pipes! (with cool examples!)
- System Administration 101

CSAW Update

We finished 10th in the Undergraduate / North America bracket

That qualifies us for the North American finals at NYU Tandon in New York

Thanks to those of you that helped out, and hope to see the rest of you playing next year!

Linux Perms (review)

read = 4
write = 2
execute = 1

d = directory
l = symlink
b = block device
c = character device

```
:)-22:18-zack@sperfari:56551:0:~/ctf/linux_talk
$ ls -l
total 28K
drwxr-xr-x 2 zack zack 4.0K Sep 12 22:13 a_directory/
-rw-r--r-- 1 zack zack    0 Sep 12 22:17 afile
-rw-r--r-- 1 zack zack    3 Sep 12 22:17 bfile
crw-r--r-- 1 root root 1, 3 Sep 12 22:15 devicefile
brw-r--r-- 1 root root 1, 3 Sep 12 22:15 devicefile2
-rwxr-xr-x 1 zack zack 8.5K Sep 12 22:16 hello*
-rw-r--r-- 1 zack zack   64 Sep 12 22:16 hello.c
drwxr-xr-x 2 zack zack 4.0K Sep 12 22:18 other_files/
lrwxrwxrwx 1 zack zack   11 Sep 12 22:13 some_symlink -> a_directory/
```

SUID/SGID

```
:)-22:25-zack@sperfari:56602:0:~/ctf/linux_talk/other_files
$ ls -l
total 44K
-rwsr-sr-x 1 zack zack 8.5K Sep 12 22:17 hello_crazy*
-rwxr-sr-x 1 zack zack 8.5K Sep 12 22:17 hello_sgid*
-rwsr-xr-x 1 zack zack 8.5K Sep 12 22:16 hello_suid*
drwxr-sr-x 2 zack adm  4.0K Sep 12 22:22 somedir/
drwxrwxrwt 2 pair pair 4.0K Sep 12 22:25 someother/
```

```
:)-22:25-zack@sperfari:56603:0:~/ctf/linux_talk/other_files
$ ls -l somedir
total 0
-rw-r--r-- 1 zack adm 0 Sep 12 22:22 afile
-rw-r--r-- 1 zack adm 0 Sep 12 22:22 bfile
-rw-r--r-- 1 zack adm 0 Sep 12 22:22 cfile
-rw-r--r-- 1 zack adm 0 Sep 12 22:22 dfile
```

```
:)-22:25-zack@sperfari:56604:0:~/ctf/linux_talk/other_files
$ ls -l someother
total 0
-rw-r--r-- 1 junk junk 0 Sep 12 22:25 afile
```

```
:)-22:25-zack@sperfari:56605:0:~/ctf/linux_talk/other_files
$ rm someother/afile
rm: remove write-protected regular empty file 'someother/afile'? y
rm: cannot remove 'someother/afile': Operation not permitted
```

There are 3 more permission bits

setuid (4)
setgid (2)
Sticky bit (1)

Files:

- setuid/setgid mean the file runs as owning user/group
- Sticky bit is useless

Directories:

- setuid is useless
- setgid means that new files are owned by that group
- Sticky bit means only owner can create files

So you can set a file suid with:

chmod 4755 /usr/local/bin/suidfile

setcap

The Linux kernel has a concept of “capabilities”. Idea is to split **root** up into pieces so a process can have just part of the privs.

Unfortunately, most of them are equivalent to full root

But they’re useful in a few cases, such as ping, which needs to send ICMP packets

```
:)-22:58-zack@sperfari:56610:0:/bin
$ ls -l | grep ping
-rwxr-xr-x 1 root root  60K Nov 10  2016 ping*
lrwxrwxrwx 1 root root    4 Nov 10  2016 ping4 -> ping*
lrwxrwxrwx 1 root root    4 Nov 10  2016 ping6 -> ping*
```

```
:)-22:58-zack@sperfari:56610:0:/bin
$ getcap ping
ping = cap_net_raw+ep
```

CAP_AUDIT_CONTROL
CAP_AUDIT_READ
CAP_AUDIT_WRITE
CAP_BLOCK_SUSPEND
CAP_CHOWN
CAP_DAC_OVERRIDE
CAP_IPC_LOCK
CAP_IPC_OWNER
CAP_KILL
CAP_LEASE
CAP_LINUX_IMMUTABLE
CAP_MKNOD
CAP_NET_ADMIN
CAP_NET_BIND_SERVICE
CAP_NET_RAW
CAP_SETGID
CAP_SETFCAP
CAP_SETPCAP
CAP_SETUID
CAP_SYS_ADMIN
… and more

Linux attributes

These are Linux specific, not to be confused with POSIX extended attributes

Change them with chattr, see them with lsattr

Most are useless, the two important ones are immutable and appendonly

Immutable makes the file unchangable, even for root!

Appendonly makes it so you can only append stuff to the file, again, even for root!

```
root@sperfari:~/linux_talk# echo "hello" >> myfile
root@sperfari:~/linux_talk# chmod 000 myfile
root@sperfari:~/linux_talk# ls -l
total 4
---------- 1 root root 6 Sep 12 23:23 myfile
root@sperfari:~/linux_talk# echo "goodbye" >> myfile
root@sperfari:~/linux_talk# cat myfile
hello
goodbye
root@sperfari:~/linux_talk# chattr +i myfile
root@sperfari:~/linux_talk# echo "nope" >> myfile
-bash: myfile: Operation not permitted
root@sperfari:~/linux_talk# lsattr myfile
----i---------e---- myfile
root@sperfari:~/linux_talk# cat myfile
hello
goodbye
root@sperfari:~/linux_talk# chattr -i myfile
root@sperfari:~/linux_talk# echo "yes" >> myfile
root@sperfari:~/linux_talk# cat myfile
hello
goodbye
yes
root@sperfari:~/linux_talk#
```

Symbolic Links (Symlinks)

Symlinks are literally my favorite filesystem feature

Basically a reference to another file. Inherits permission, content, etc from the file it points to.

Created with the ln command (ln -s /path/to/file /path/to/link)

Can’t remember the order? It’s like cp

Since it has the same permissions as the file it points to but a different name, it can help you bypass a blacklist or whitelist in a SUID program. (hint for wargame)

Also, if you have a program that expects a file in a certain location, but you want it somewhere else, symlinks help with that

Think “Portal”

Pipes! | | | | | | |

Pipes sends the output of one program to another program. Pipes can be chained together to form pipelines, which can do some pretty powerful stuff.

Common pipe recipients:

grep - search input for some pattern

cut - extract text from input based on some delimiters

tail/head - Get only the last few/first few lines of the input

sed - text processing tool, generally useful for substituting things

awk - another text processing tool, has a bit of a learning curve

Pipe demos

Find top 10 commands

cat .bash_history | grep -v "^#" | sed "s/sudo //" | sed -E "s/^\s+//" | cut -f 1 -d " " | sort | uniq -c | sort -nr | head -40

Defcon CTF Qualifiers 2017: crackme2000 solutions

objdump -M intel -d $1 | grep “cmp” | grep “rdi,0x” | cut -d”,” -f2 | xxd -r -p

objdump -M intel -d $1 | grep -1 “e.x,BYTE PTR \[rax” | grep cmp | grep -v “eax” | grep -v “al” | cut -d”,” -f2 | xxd -r -p

Fork bombs!

:(){ :|: & };: