

1. Last time

2. Protection and security in Unix

Intro

Setuid

TOCTTOU

Other thoughts

---

2. Protection and security in Unix

A. Intro

UIDs, GIDs (user id, group id)

A process has one user id and one or more group ids

Files and directories are access-controlled

- Saw this in lab 2 (ls)

- System stores w/ each file who owns it

- Where is the info stored?

Special user: uid 0, called root, treated by kernel as the administrator.

(h) | || permissions: can

UID 0 (root) has all permissions

read any file, do anything.

certain things only root can do:

- set clock
- halt machine
- mount filesystems
- change process's user or group id

## B. setuid

ex: how do users change their password?

\$ passwd

/etc/passwd  
/etc/shadow

A prog. can be "setuid"

\$ ls

real uid: mwatfish  
effective uid: mwatfish

\$ passwd

real uid: mwatfish  
effective uid: root

"su": change to new userid if correct passwd is typed.

## Example attacks

(a) `close(2);`  
`exec("/usr/bin/passwd");`

passwd:

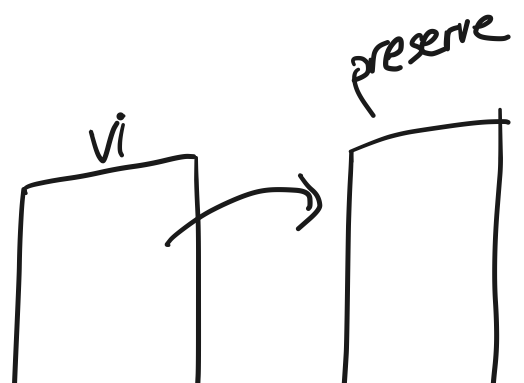
main() *assumption:*

{ *red: 2* *red: 2* *red: 3* `fd = open("/etc/passwd", ...)`

*red: 2* `write(fd, ...)`

*red: 2* `fprintf(stderr, "Err msg\n");`

(b) old days: "preserve"  
`setuid root`



IFS = "/



attacker:  
create "bin, which does:  
→ cp /bin/sh ./attack  
chown root attack  
chmod 4755 attack

preserve:  
system("/bin/mail");

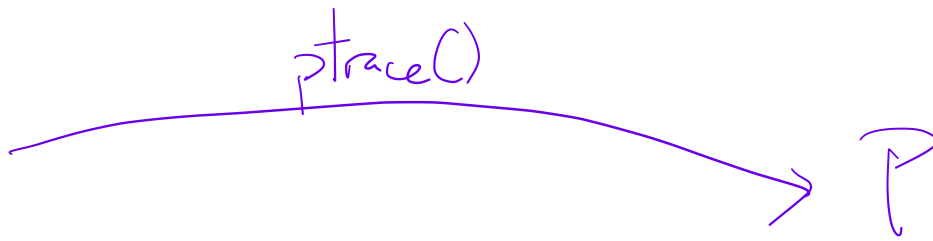
parsed: "bin mail"

---

ptrace()

Attacker

setuid prog: P  
root



---

TOCTTOU

setuid program: P <logfile>

⋮

fd = ~~open~~(logfile, ← user input  
O\_CREAT | O\_WRONLY |  
O\_TRUNC,  
0666)

access(): check whether the real id, not effective id, is allowed to access the file.

```
if (access ("/tmp/X", W_OK) < 0)
    return ERROR;
```

```
fd = open ("/tmp/X", ...) /* as above */
```

attacker runs "P /tmp/X"

P

attacker  
creat("/tmp/X");

checks access("/tmp/X") → OK

unlink("/tmp/X");

symlink("/etc/passwd", "/tmp/X");

```
open("/tmp/X", O_TRUNC | O_WRONLY);
```

---