

- 1. Last time
  - 2. Protection and security in Unix
    - Intro
    - Setuid
    - TOCTTOU
    - Other thoughts
- 

## 2. Protection and security in Unix

357  
2417

### 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 (1s)
- 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.

(... || 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: mwaldfish  
effective uid: mwaldfish

\$ `passwd`

real uid: mwaldfish  
effective uid: root

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

## Example attacks

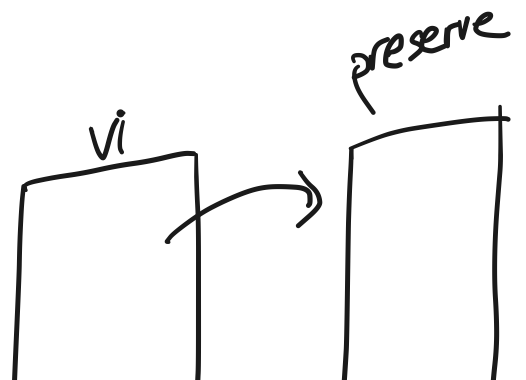
(a)

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

passwd:

```
main()  
{  
    fd = open("/etc/passwd", ...)  
    :  
    :  
    fprintf(stderr, "Err msg\n");  
}
```

(b) old days: "preserve"  
setuid root



IFS = " /"

attacker:  
create bin, which does:

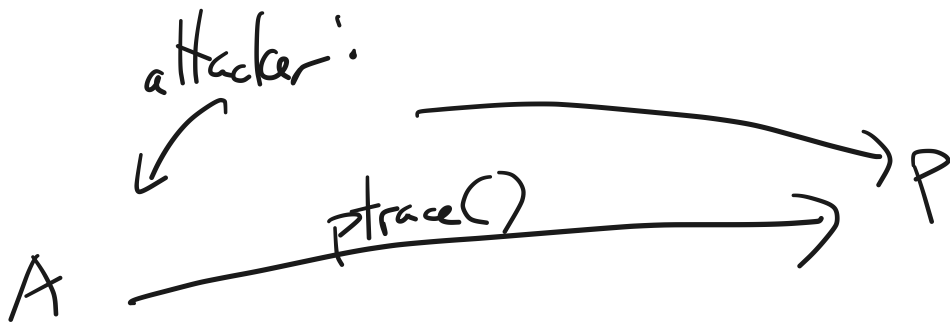
→ `cp /bin/sh ./attack`  
`chown root attack`  
`chmod 4755 attack`

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

bin  
mail

trace()

etw



TOCTTOU

setuid program: P <sup><logfile></sup>

⋮  
fd = ~~open~~(logfile, <sup>user input</sup>  
O\_CREAT | O\_WRONLY |  
O\_TRUNC,  
0666)

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

if (access(logfile, W\_OK) < 0)  
return ERROR;

fd = open(logfile, ... ) /\* 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);

---

OS trasactions