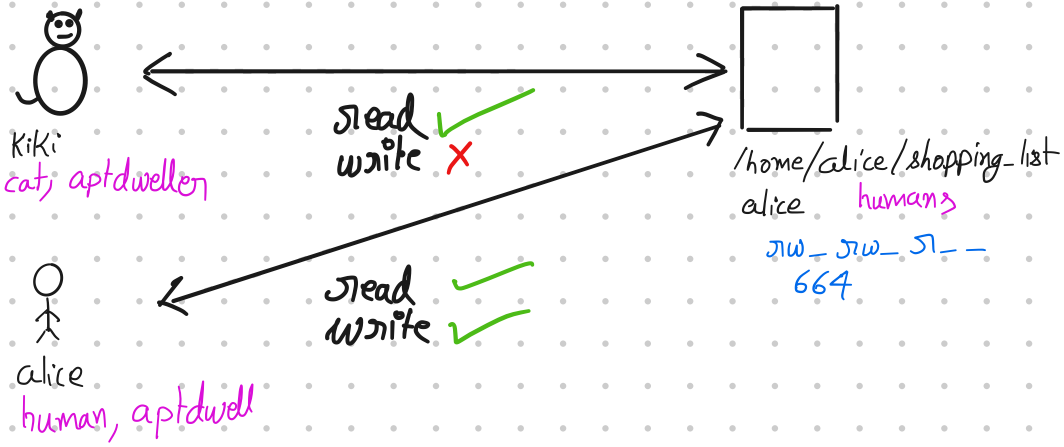
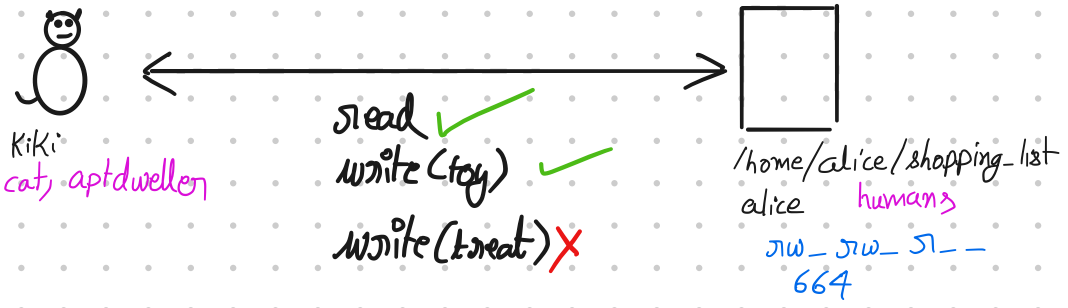
CS202 (-002) More security

- Last class
  - Users, groups
  - File permissions
- Where we left off
  - File permissions allow us to limit whether a user can read, write, or execute a file

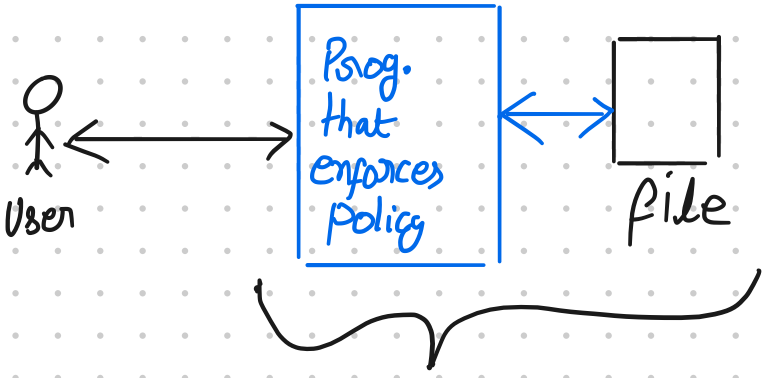


No way to allow only some writes



Restricting what a user can read from/write to a file is often what we want

- Albert
- Google Docs
- ...

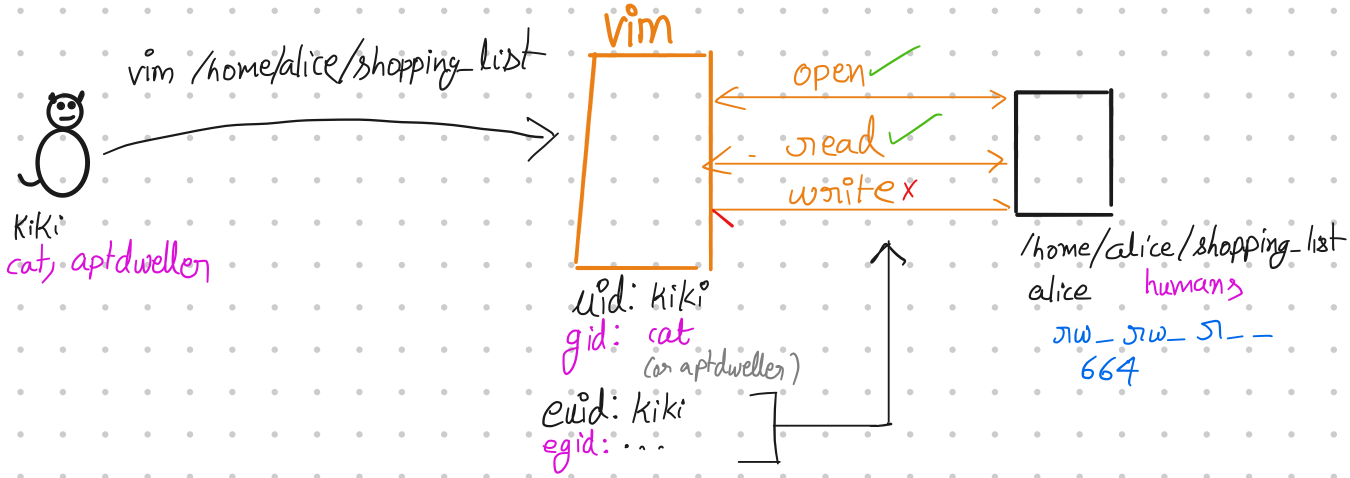


Primitive: Only programs that enforce desired access

How

① Put user and file on different computers, force user access through program
- Albert
- Google Docs
- ...

② Setuid — Computers were once expensive

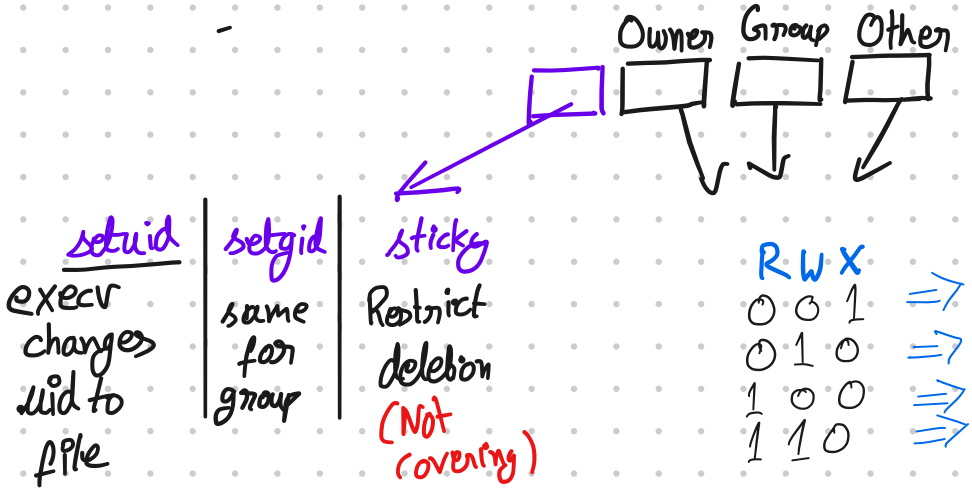


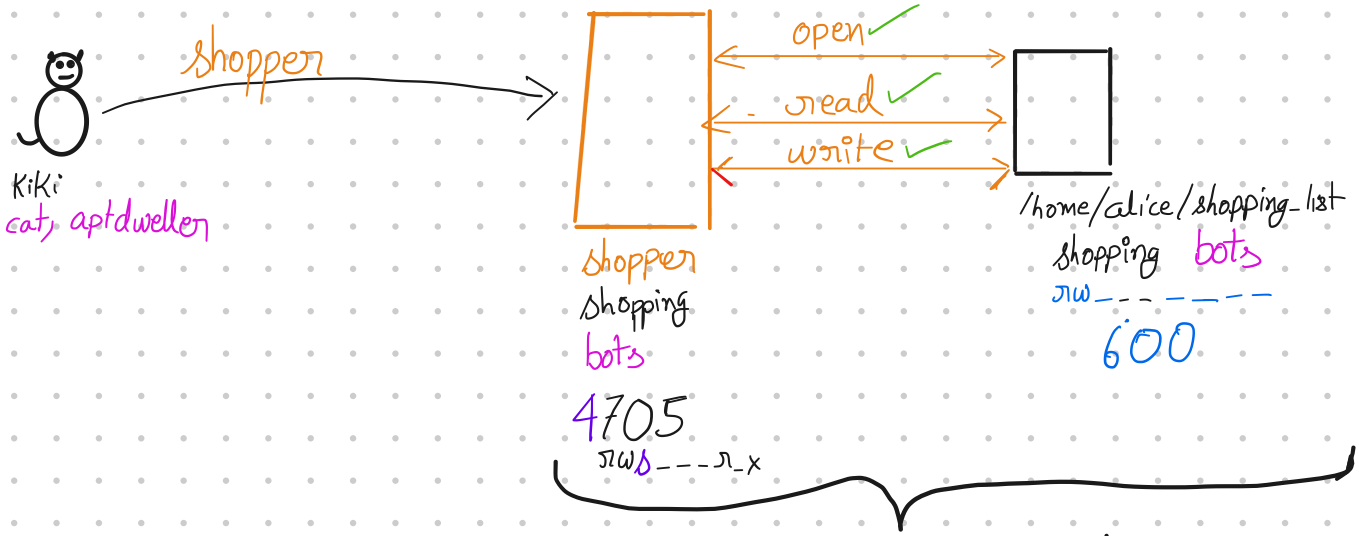
- Processes have an associated uid, gid
  (also associated effective UID, GID)
  For now, just assume uid ≡ euid
- Can read this
  getuid(2)
  getgid(2)
  geteuid(2)
  getegid(2)
- Can also change them — within reason. Can decrease privilege
  e.g., root → alice
  setuid
  setgid

setuid, setgid → To help with the privilege thing.

- Initial value
  - Usually inherited on fork
    ↳ Child has same uid, gid as parent ★

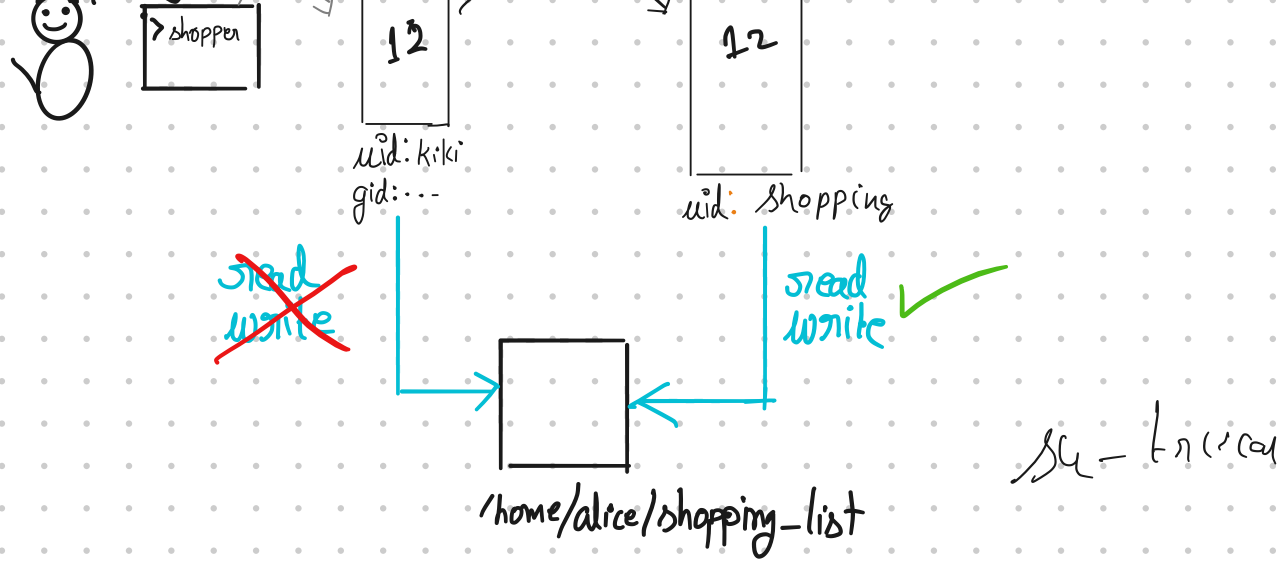
-





Observe, only shopping has access to /home/shopping_list

What is happening



su - typical

Used by several tools

- su
- passwd
- sudo
- ...

all owned by root

↳ Call setuid/setegid

root wheel
rwsr-x--- r-x---

But can also be a source of security problems

↳ Why? Run program ⇒ another user's identity (e.g., root)

Must make sure there is no way to trick program to do unsafe things

→ e.g., Output to the wrong file (by redirecting files)

⇒ Using a debugger

↳ Notes mention ptrace, ptrace(2) is the syscall most commonly used to build debuggers.

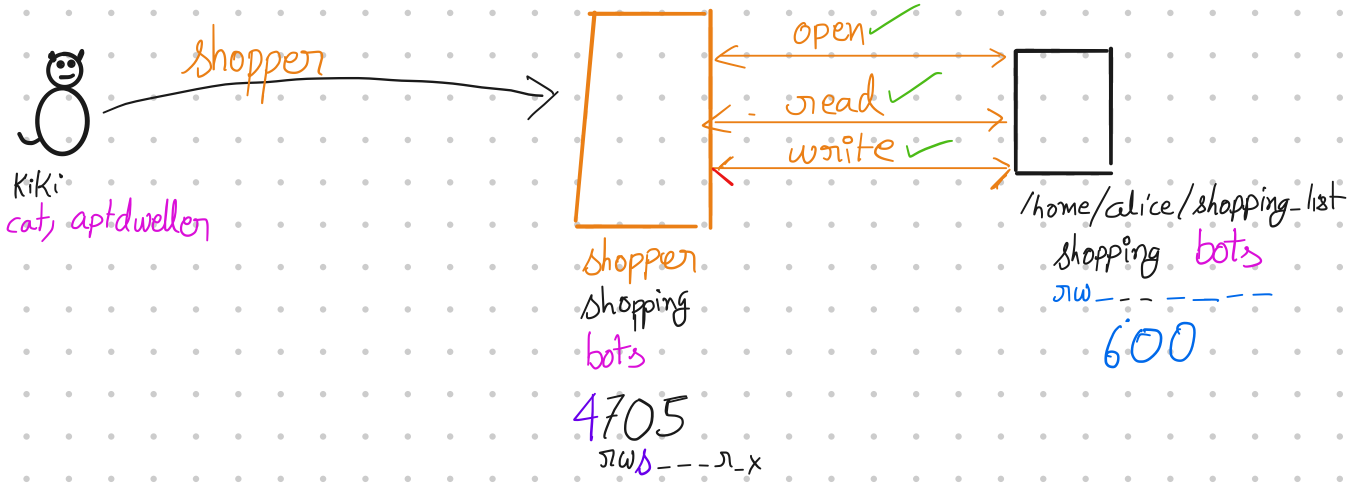
→ ...

Time of check to time of use (TOCTOU)

→ Last bit of security: a common class of bugs

→ Generally, a common pattern for sec.

```
if (condition) {
    Do Thing
} else {
    Error out
}
```




```
int fd = open("/home/alice/shopping_list", Read || write)

① if (!contains(fd, "toy")){
②         write (fd, "toy");
③ } else {
④         printf("nope");
⑤ }
```

Q: How many times can toy show up in /home/alice/shopping_list ?

> ./shopper &; ./shopper &; ./shopper

General problem

```
① if (!contains(fd, "toy")) {
②         write(fd, "toy");
③ } else {
④         printf("nope!");
⑤ }
```

Contents of file here ≠ Contents here

More generally

```
if (condition) {
    Do Thing
} else {
    Error out
}
```

Condition might not hold here.

Time of check to time of use

- You saw how to deal with similar bugs before.

Where?

- How would we fix the problem if we had concurrency control tools

```
lock(&m);          <-- Protects state determining condition
if (condition) {
    Do Thing
} else {
    Error out
}
unlock(&m);
```

Unfortunately, cannot do this here.

↳ Locks taken & released by processes don't suffice — Why?

Solutions? No good solutions — mostly lots of careful hacks.

Shopper 1

```
lock(&m);
if (x == 23){
    x += 100;
}
unlock(&m);
```

Shopper 2

```
x = 3;
```