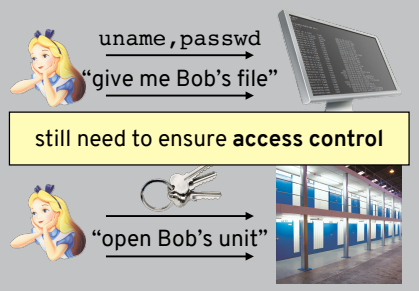
SECURITY (COMP0141): ACCESS CONTROL

UCL

ACCESS CONTROL



As we've seen, authentication gets you in the door, but it's important to think about what happens afterwards. What are you allowed to do then?

SECURITY DESIGN

define

How to ~~design~~ a secure system?

one that meets a specific security policy

How to define a security policy?

use threat model and build policy to address it

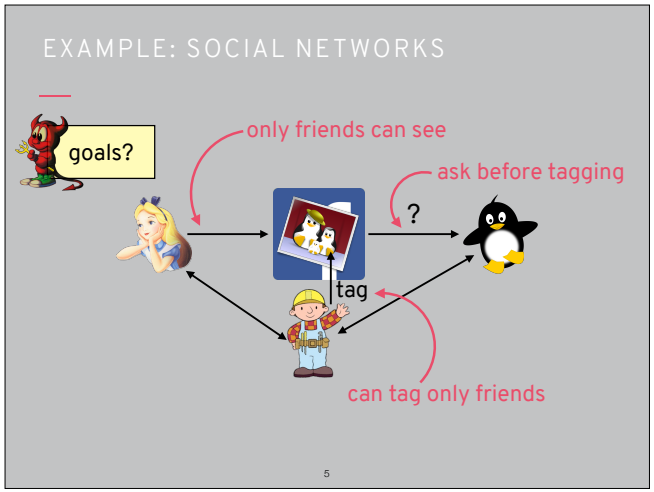
As we'll see, access control is really all about defining policies

ACCESS CONTROL

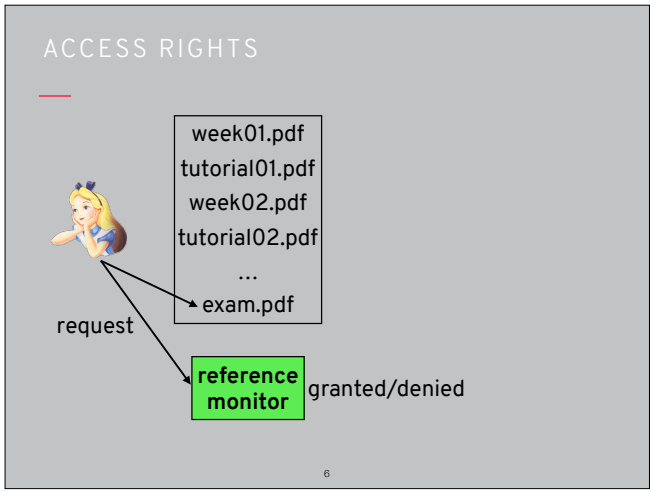
Access control is the ability of one entity to permit or deny the use of a particular resource to another

Informal: "We don't want people wandering in off the street"
Formal: "Only UCL staff and students can enter that area"

Authentication is already a (coarse) form of access control



Social networks are a good example of issues in access control that we'll see later – complex set of interactions with different and potentially contradictory permissions for each one



An even easier example is reading a file in a folder

TYPES OF FILE ACCESSES

subjects (s)
objects (o)
access rights (r/p)

	non-ALT	ALT
non-OBS	execute	append
OBS	read	write

Subjects are the users of the system

Objects are the different files

Access rights: execute, read, write, append (some combination of ALTeration and OBServation)

To deal with accesses we need to consider subjects, objects, and rights (or permissions)

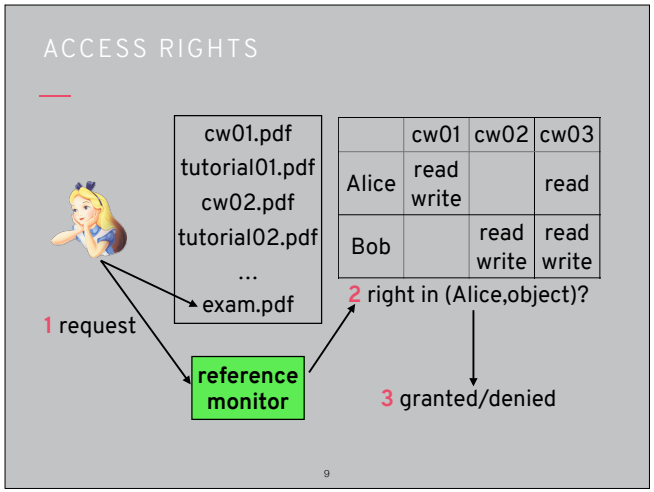
ACCESS CONTROL MATRIX

S: Alice, Bob
O: cw01,cw02, cw03
R: read, write

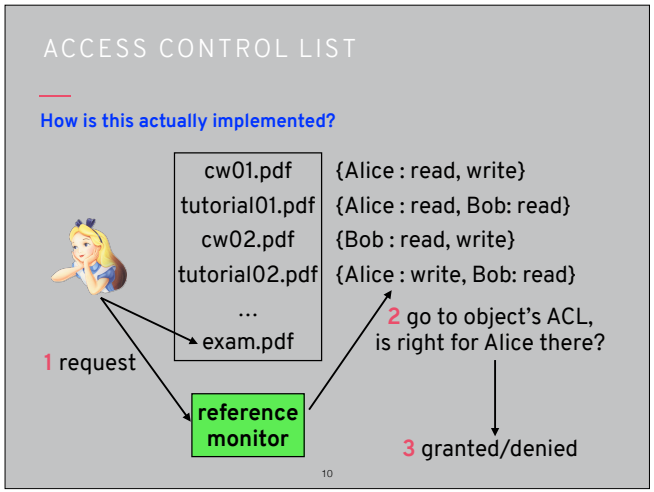
	cw01	cw02	cw03
Alice	read write		read
Bob		read write	read write

can Alice read cw01?
can Bob write cw01?

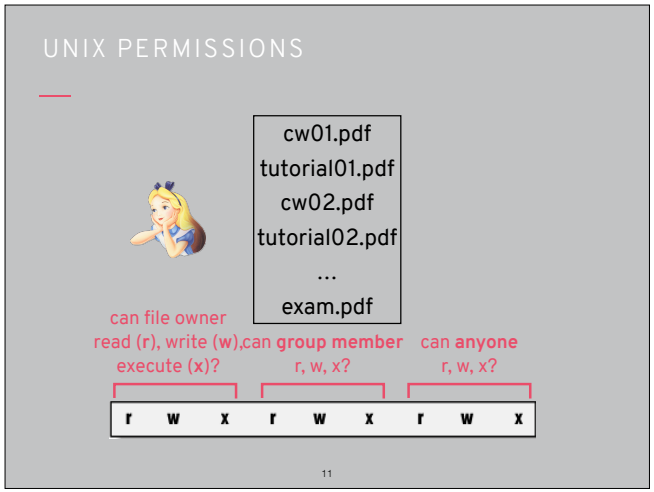
An access control matrix says which subjects have which access rights for which objects. Here Alice (a subject) can read cw01 (an object) but Bob has no access rights for cw01 so can't write to it



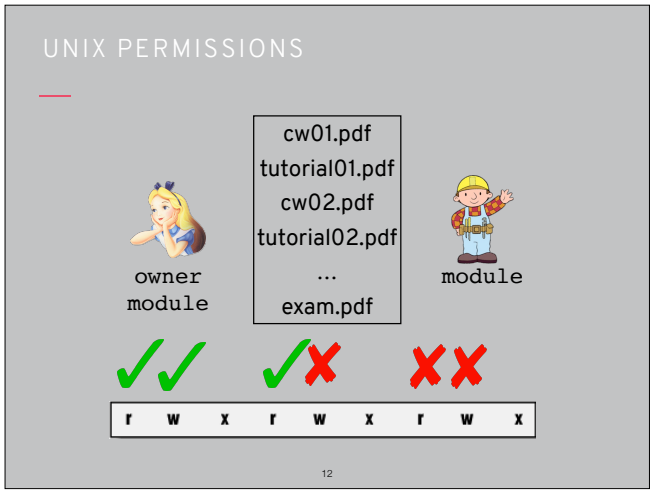
Alice's request looks like the pair (right,object), and the reference monitor checks to see if Alice has that right for that object



Access control lists are an alternative to an access control matrix in which a separate list is maintained for each object. This is what we do in practice because trying to maintain a matrix would require too much memory

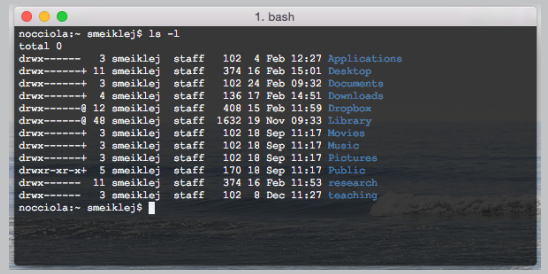


We briefly mentioned UNIX permissions back in Week 2, now we'll see them in more detail since they are an important security mechanism for file-based access control



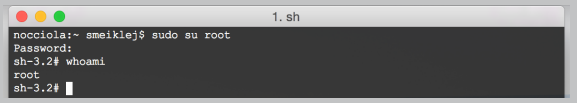
For example, if Alice creates an exam for a module, she should be able to read and write the exam file, but students in the module should only be able to read it (and everyone else can't see it at all)

UNIX PERMISSIONS: DEMO



Demo of how to find and update permissions on a Unix-based file system (works on Mac or Linux)

ROOT USER

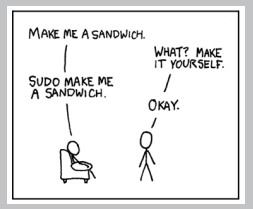


default owner of all system files
protects users from themselves!
especially important in multi-user systems

but what if I want to execute certain tasks?

Root is also known as a “superuser”

SUDO



allows one user to temporarily run things with privileges of another (often `root`)

accountability: `sudo` usage is logged

Sudo gives special temporary permissions

PERMISSIONS FOR DIRECTORIES

cw01.pdf
tutorial01.pdf
cw02.pdf
tutorial02.pdf
...
exam.pdf

read = list contents
execute = traverse
write = **modify** files

can file owner read (r), write (w), execute (x)?

can **group member** r, w, x?

can **anyone** r, w, x?

r	w	x	r	w	x	r	w	x

One small thing we didn't cover yet: what do permissions mean for directories rather than just files?

STICKY BIT

Can do this even if you don't have write permissions on the individual files!

The **sticky bit** (T) for a directory changes write privileges, can rename or delete files only if you are the owner (or `root`)

read = list contents
execute = traverse
write = **modify** files

= create, rename, or delete

```
nocciola:~ smeiklej$ chmod 1700 research
nocciola:~ smeiklej$ ls -l
total 0
drwx------+ 10 smeiklej  staff   320 12 Mar 16:29 Desktop
drwx------+  4 smeiklej  staff   128  6 Mar  2015 Documents
drwx------+  4 smeiklej  staff   128 13 Mar 09:30 Downloads
drwx------@ 18 smeiklej  staff   576 11 Mar 21:18 Dropbox
drwx------@ 74 smeiklej  staff  2368  5 Nov 11:59 Library
drwx------+  3 smeiklej  staff    96 18 Sep  2014 Movies
drwx------+  6 smeiklej  staff   192 20 Nov  2017 Music
drwx------+  6 smeiklej  staff   192 21 Jun  2017 Pictures
drwxr-xr-x+  5 smeiklej  staff   160 18 Sep  2014 Public
drwx---T--  21 smeiklej  staff   672 17 Dec 10:17 research
drwx------   8 smeiklej  staff   256 20 Dec 13:58 teaching
drwx------   6 smeiklej  staff   192  5 Jan 21:11 writing
```

17

Sticky bit prevents one type of misbehaviour in a multi-user system

---

UNIX PERMISSIONS: POP QUIZ!

permissions	owner	group	filename
rwx-----x	bob	eng	week01.pdf
rwxrwxrwx	bob	eng	cw01.pdf
rwx--x--x	alice	alice	week02.pdf
rw-r-----	alice	cs	cw02.pdf
rw-r--r--	bob	cs	week03.pdf
rw--wxr--	root	cs	exam.pdf

alice,cs

which files can Alice write?

18

Alice can write cw01.pdf (world), week02.pdf (owner), cw02.pdf (owner), and exam.pdf (group)

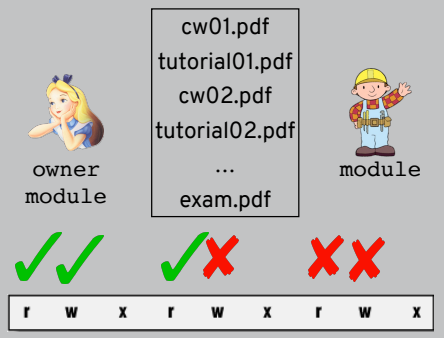
DESIGN PRINCIPLES

Least privilege
Separation of responsibilities
Complete mediation
Fail-safe default
Defence in depth
Open design
Psychological acceptability
Economy of mechanisms

19

Going back to design principles, we can see that Unix permissions exemplify many of them

LEAST PRIVILEGE



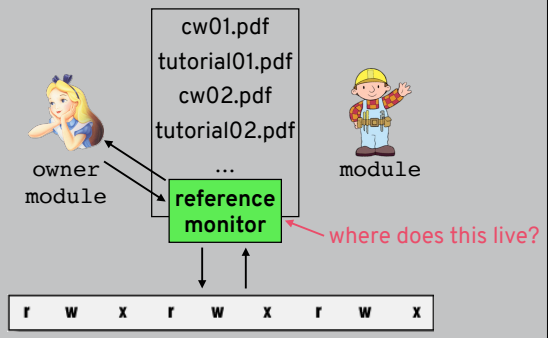
20

Permissions clearly demonstrate least privilege (if they're set properly)

DESIGN PRINCIPLES

Least privilege
Separation of responsibilities
Complete mediation
Fail-safe default
Defence in depth
Open design
Psychological acceptability
Economy of mechanisms

COMPLETE MEDIATION



Complete mediation works if reference monitor checks permissions every time and can't be corrupted

TRUSTED COMPUTING BASE (TCB)

Trusted computing base (TCB) refers to every component of the system upon which the security policy relies (could be hardware, software, etc.)

In other words, if something goes wrong then the security policy may be violated

This needs to be kept small!

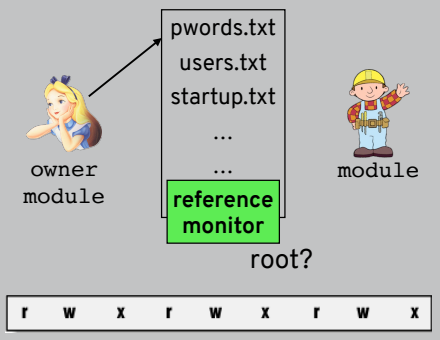
This is an example of economy of mechanisms (could just think of entire system as TCB but this is very unrealistic)

We first saw the idea of a TCB back in Week 2, and it's important for complete mediation that the reference monitor lives in the TCB to ensure it can't be corrupted

DESIGN PRINCIPLES

Least privilege
Separation of responsibilities
Complete mediation
Fail-safe default
Defence in depth
Open design
Psychological acceptability
Economy of mechanisms

FAIL-SAFE DEFAULT



Root helps to achieve fail-safe default (protects the user from themselves)

DESIGN PRINCIPLES

Least privilege
Separation of responsibilities
Complete mediation
Fail-safe default
Defence in depth
Open design
Psychological acceptability
Economy of mechanisms

Permissions also satisfy open design, psychological acceptability, and economy of mechanisms

NEXT TIME

subjects (s)
objects (o)
access rights (r/p)

	non-ALT	ALT
non-OBS	execute	append
OBS	read	write

Subjects are the users and processes of the system

Objects are the different files

Access rights: execute, read, write, append (some combination of ALTeration and OBServation)

Users are not the only one interacting with files, there are also many processes running behind the scenes