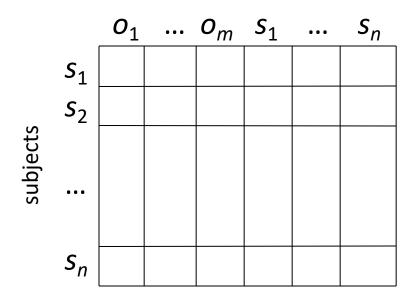
# ECS 235B Module 4 Access Control Matrix

#### Description

#### objects (entities)



- Subjects  $S = \{ s_1, ..., s_n \}$
- Objects  $O = \{ o_1, ..., o_m \}$
- Rights  $R = \{ r_1, ..., r_k \}$
- Entries  $A[s_i, o_j] \subseteq R$
- $A[s_i, o_j] = \{r_x, ..., r_y\}$  means subject  $s_i$  has rights  $r_x, ..., r_y$ over object  $o_i$

- Processes p, q
- Files *f*, *g*
- Rights *r*, *w*, *x*, *a*, *o*

	f	g	p	q
p	rwo	r	rwxo	W
q	а	ro	r	rwxo

- Host names telegraph, nob, toadflax
- Rights own, ftp, nfs, mail

talograph

telegraph nob toadflax

telegruph	1100	ισααίταχ
own	ftp	ftp
	ftp, mail, nfs, own	ftp, nfs, mail
	ftp, mail	ftp, mail, nfs, own

nah

toadflay

- Procedures inc\_ctr, dec\_ctr, manage
- Variable counter
- Rights +, -, call

	counter	<u>inc_ctr</u>	<u>dec_ctr</u>	manage
inc_ctr	+			_
dec_ctr	_			
manager		call	call	call

#### UNIX/Linux Access Controls

• Files: A is ~bishop/a.out (0755), B is /etc/passwd (0644), H is /home/bishop (0711), S is /bin/su (4711)

bishop zheng root

$\overline{A}$	D	3	11
rwxo	r	X	rwxo
rx	r	X	X
rwx	rwo	rwxo	rwx

#### UNIX/Linux Access Controls

- Access control matrices are dynamic:
- After bishop executes chmod 700 /home/bishop:

bishop muwei root

A	В	S	Н
rwxo	r	X	rwxo
	r x		
rwx	rwo	rwxo	rwx

#### Boolean Expression Evaluation

- ACM controls access to database fields
  - Subjects have attributes
  - Verbs define type of access
  - Rules associated with objects, verb pair
- Subject attempts to access object
  - Rule for object, verb evaluated, grants or denies access

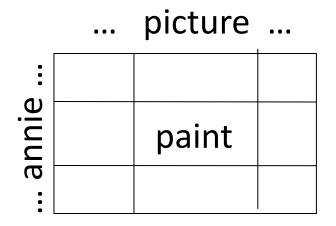
- Subject annie
  - Attributes *role* (artist), *group* (creative)
- Verb paint
  - Default 0 (deny unless explicitly granted)
- Object picture
  - Rule:

```
paint: 'artist' in subject.role and 
'creative' in subject.groups and 
time.hour ≥ 0 and time.hour ≤ 4
```

#### ACM at 3AM and 10AM

At 3AM, time condition met ACM is:

At 10AM, time condition not met ACM is:





#### History

- Problem: what a process has accessed may affect what it can access now
- Example: procedure in a web applet can access other procedures depending on what procedures it has already accessed
  - S set of static rights associated with procedure
  - C set of current rights associated with each executing process
  - When process calls procedure, rights are  $S \cap C$

## Example Program

```
// This routine has no filesystem access rights
// beyond those in a limited, temporary area
procedure helper proc()
      return sys kernel file
// But this has the right to delete files
program main()
      sys_load_file(helper_proc)
      tmp file = helper proc()
      sys delete file(tmp file)
```

- sys\_kernel\_file contains system kernel
- tmp\_file is in limited area that helper\_proc() can access

## Before helper\_proc Called

• Static rights of program

main helper\_proc

sys_kernel_file	tmp_file
delete	delete
	delete

• When program starts, current rights:

main helper\_proc process

sys_kernel_file	tmp_file
delete	delete
	delete
delete	delete

## After helper\_proc Called

• Process rights are intersection of static, previous "current" rights:

	sys_kernel_file	tmp_file
main	delete	delete
helper_proc		delete
process		delete

#### Quiz

In an access control matrix, do the rights "r", "w", and "x" represent "read", "write", and "execute" permissions, respectively?

- Yes, because the permission symbols are tied to those permissions ("r" for "read", "w" for "write", "x" for "execute").
- Possibly; the meanings of the permission symbols depends upon the instantiation.
- No, because the meanings of the permission symbols changes as the matrix evolves, so you cannot say what the symbols mean; you can only manipulate them based on the given commands.