

TA: George Iordache

CC all reviews to: georgei+509r@cs.sunysb.edu

ACM

- undecidable
- ACL lists, capabilities

Bell-Lapudula

- read
- write
- MAC vs DAC

Bell-Lapudula

ex. (Top Secret, $\{Area\ 513\}$)
(Confidential, $\{3\}$)

- ordering on labels
"Read down, write up"

For read:

Process with l can read process with l' if $l' \leq l$

For write:

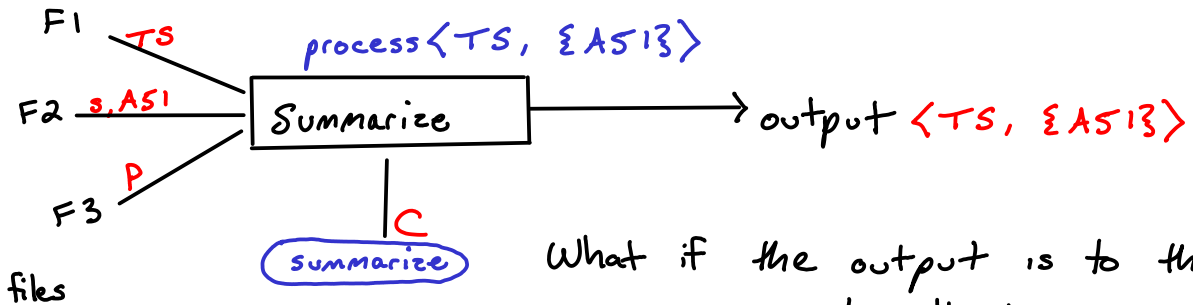
Process with l can read process with l' if $l \leq l'$

Basic Security Theorem

If Σ starts in secure state and we use read down / write up, then the system can never reach an insecure state

Two strategies:

- ① fixed process label
- ② process label = lab(labels of files read)



What if the output is to the screen, and the user doesn't have clearance?

MAC - Mandatory Access Control (system tells you)

DAC - Discretionary Access Control (you tell system)

Declassification

- special, trusted declassification process that is allowed to break the read down/write up rule

Combine policy and privilege

ex: ~~secret~~ ^{declassifier} → averager → average salary ^{public}

ex: separation of privilege

- declassifier shouldn't have bugs
- no theorems :(

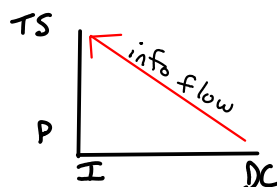
Biba (Bell-Laputula upside down) - Read up, write down

Double Checked (DC)

Reliable Witness

Anonymous Tips

Internet (I)



Over time, info is

- less reliable
- more secret

Role-Based Access Control (RBAC)

- rights with roles in organization
- each user is assigned set of roles

<u>University</u>	
<u>role</u>	<u>rights</u>
student	submit hw, view grade
professor	read hw, write grade
dean	change grade

	submit HW	read HW	write grade	change grade
student				
professor				
dean				

Static separation of duties

No user is both professor and student

hierarchical role:

prof \leq dean

Capabilities & the Confused Deputy

- Process possesses a set of rights
- OS automatically allows access if any one of the process' rights allows it
- Process can't decide which write is used for access

Ex: Computer that stores billing info in special file /var/cc/txns

writable only by processes in CC group

is setgid cc

% CC hello.c -o hello.exe \leftarrow should only write to the file if user can

% CC hello.c -o /var/cc/txns

Fixes:

- Check args for `"/var/cc/txns"`
 - doesn't scale as number of "special" files grows
 - can get out of sync
 - symlink } canonicalization
 - `/var/./`
- Separate privilege management from privilege enforcement.
 - App manages privileges
 - OS checks them
 - App must provide proof that access is ok to the OS

Compiler source:

```
// write billing info
open("/var/cc/txns" CCtoken)
...
open(outfile, user token)
↑ fails if outfile = /var/cc/txns
```

- Compiler Holds
 - two tokens: CC token, user token

Capability Implementations

- Cryptography
- Unforgable Pointers - unix file handles