

# Authorization

---

UT CS361S

FALL 2020

LECTURE NOTES

# Authentication/Authorization

---

Validating  
Identity

Permissions  
Assigned to a  
Validated Identity

# Access Controls

---

The mechanism by which authorization permissions are managed

Within most information systems, the most common controls:

- (C)reate
- (R)ead
- (U)pdate
- (D)elele

Most other controls can be thought of as a form of one of these

# Every-day Approaches

---



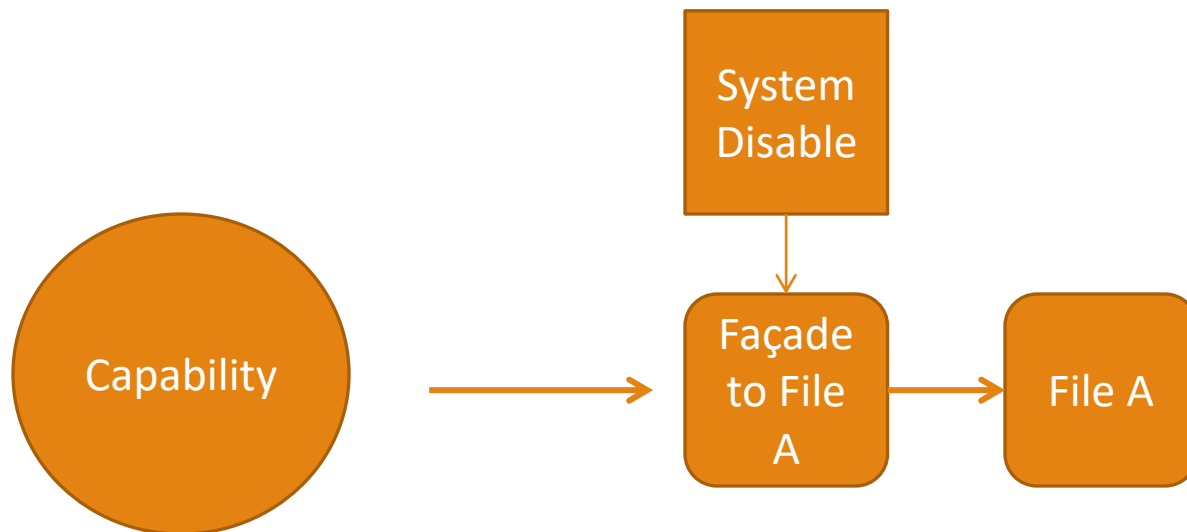
ACCESS CONTROL LISTS



CAPABILITIES

---

Opponents of capabilities argue that you cannot change a file's status  
They just don't understand capabilities



# MAC vs DAC

---



Mandatory Access Controls – what is permitted is determined by policy



Discretionary Access Controls – what is permitted is determined by user

# Multi-Level Security (MLS)

---

Often seen as synonymous with MAC

Users and data are assigned classifications

What users are permitted to do with data depends on both labels

# Bell Lapadula Model

---

Design emerged from military document classification

Enforces two properties

- *Simple Security Property*: No Read Up (NRU)
- *\*-Property*: No Write Down (NWD)

The \*-property was the big innovation of BLP. It *assumed* trojans and buggy code!

This is a well defined security policy

- It is relatively easy to determine if the mechanisms enforce the policy
- If it's the right policy it works great!



# Problems of BLP

---

If the security officer can “temporarily declassify” all of the protections go away

- Strong tranquility: security labels never change during operation
- Weak tranquility: labels never change in a way that violates security policy
  - The idea here is “least privilege”. Even if you have TS, start at unclassified
  - As you access info that is higher, your level increases

The system can get fragmented into pieces that can’t communicate

Also, what do you do with an App that has to straddle?

- A document editor used to redact a TS document to Classified

Doesn’t deal with creation of subjects or objects

# Biba model

---

## Upside-down BLP

- You can only read up and write down
- The goal is *integrity* not *confidentiality*

Partially used in Vista. Uses the NoWriteUp.

- Most files are “medium” or higher. IE is “low”
- So, things downloaded can read most files, *but not write to them!*

This was the first formal model of integrity

- Struggled in real-world because of the exceptions and straddling issues

# Inference

---

Information sharing often involves some kind of “scrubbing”

In MLS, a report is redacted before moving down a security layer

In privacy-preserving systems, data is often *anonymized*

The problem, of course, is inference

- People can often be identified by their medical records even with names removed
- And, of course, we’ve seen this with AOL and Google

# Inference Control

---

Characteristic formula – the query instructions to get some set

Query set – the set produced by a characteristic formula

Sensitive Statistics – stats that deanonymize information:

- For example, if the set is too small, than we've identified an individual by attributes

# Query Size

---

You can limit how small a result is from a query

But you also have to worry about returning  $N-1$ !!

Also, you have to deal with using multiple queries to get a smaller than  $N$  intersection

# Role Based Access Control

---

- RBAC
- Users assigned roles, permissions based on roles
- Is this MAC or DAC?
- Lessons from the field: what goes wrong in RBAC?

# Authorization Principles

---

- Least Privilege
- Separation of Duties