Authorization

UT LAW396V

SPRING 2023

LECTURE NOTES

Authentication/Authorization

Validating Identity

Permissions Assigned to a Validated Identity

A Framework

Policy

Mechanism

Assurance

Incentives

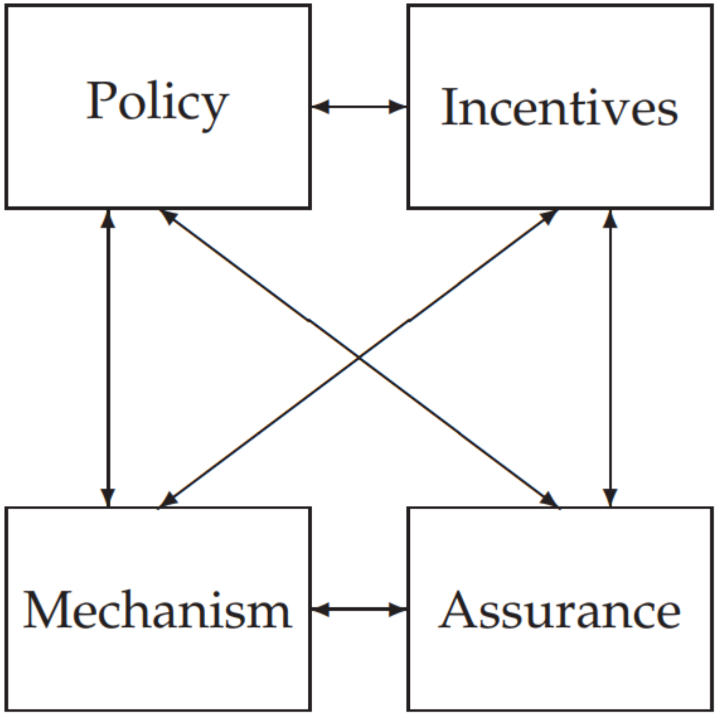


Figure 1.1: Security Engineering Analysis Framework

What a Security Policy is ***NOT***

- Generic platitude statements
- Butt-covering for legal/regulation
- Aspirational, motivational, etc

What a Security Policy *IS*

- Specific, testable properties
- A strategy for security
- Example:

“All checks over $10,000 must be signed by two managers.”

Policy vs Policy Model

- I prefer “policy” for each statement
- I prefer “policy model” for the combination
- Anderson uses them interchangeably
- There are other formulations

Policy and Authorization

- Policy (model) is the security strategy
- Testable security statements
- ***Often an authorization model***
- (Policy defines what is authorized)

What Policy Does/Does Not

- ***Defines*** a way of measuring/testing security
- See previous “signed by 2 managers” example
- Does ***not*** prevent “something bad happening”
- Is ***not*** guaranteed to be “right”
- If policy is right, bad things ***can*** happen
- If policy is wrong, bad things ***will*** happen

Conceptual Building Blocks

- Nothing here is ***implementation***
- There are still conceptual components:
  - ***Permission Models***
  - ***User Models***
  - ***Data Models***
  - ***Enforcement Models***
  - ***Objectives***

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)elete

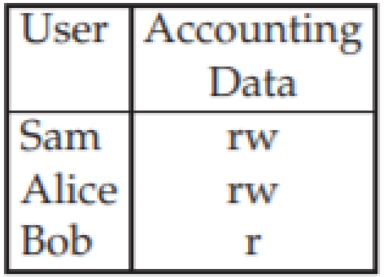
Most other controls can be thought of as a form of one of these

Every-day Approaches

ACCESS CONTROL LISTS

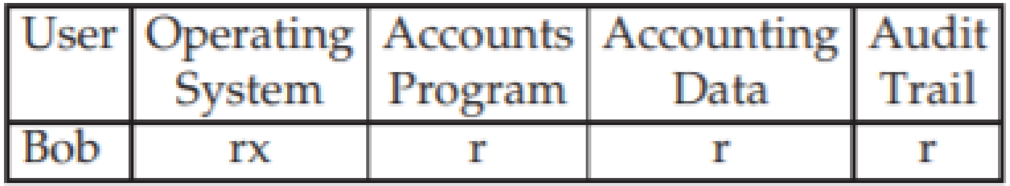
CAPABILITIES

One View of ACL/Capabilities



User	Accounting Data
Sam	rw
Alice	rw
Bob	r

Figure 4.4: Access control list (ACL)



User	Operating System	Accounts Program	Accounting Data	Audit Trail
Bob	rx	r	r	r

Figure 4.5: A capability

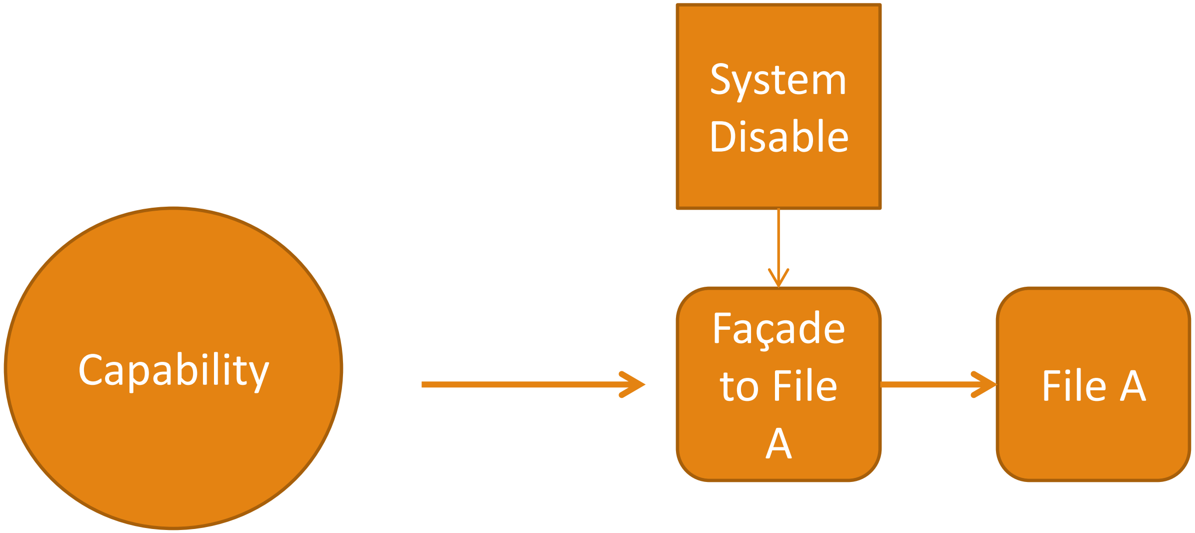
Broader Concept

A ***capability*** is an *enabling* technology for access

An ***access control list*** is a *filtering* technology for access

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)

Users and data are assigned classifications

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

Relationship to MAC:

- Some use it interchangeably
- Some define parts of such a system as MAC (see next slide)
- Anderson does not say the policy is MAC, but the controls that enforce it are
- ***I prefer Anderson’s formulation***

Bell Lapadula Model

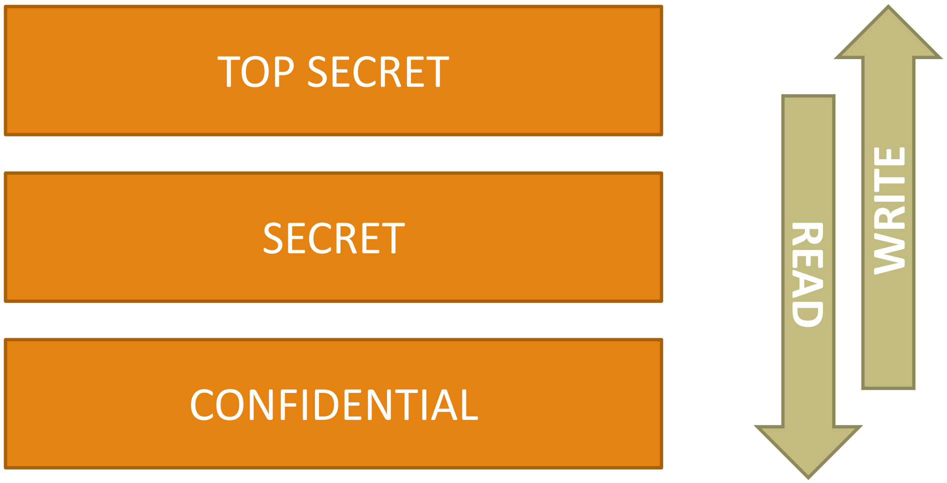
Design emerged from military document classification

Three protection properties

- *Simple Security Property:* No Read Up (NRU)
- **-Property:* No Write Down (NWD)
- Discretionary Access Controls ***within the label***

No Read Up/No Write Down

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



BLP as a Model Security Policy

- BLP is a well defined security policy
- So… is a system with BLP ***secure?***
- BLP itself is relatively easy to understand and enforce, ***BUT:***
  - **Is it the <u>right</u> security policy?**
  - **Is it going to work <u>at the edges</u>?**

BLP Edge #1: Declassifying Data

- What if we ***do*** want to write data down?
- Original BLP “gets around” this by having trusted subjects
- The NWD policy only applies to ***untrusted*** subjects
- But there is no definition for trusted/untrusted
- Trusted subjects introduce ***two*** risks:
  - Risks for any trusted subject
  - Risks that designers will make too many subjects trusted
- Other solutions: security officer, additional policy, etc

BLP Edge #2: Creation of labels

- Model does not say how to create data, subjects, or labels
- Described by the creators of BLP, but not part of the model
- Common solutions are data created by subject at same level
- But how do subjects get their level?

Example of Additional Policy

- **Strong tranquility**: security labels never change during operation
  - Example: put system into offline state to make changes
- **Weak tranquility**: labels never change in a way that violates security policy
  - As subject accesses info that is higher, their level increases
  - At any given time, the NWD policy is enforced

BLP Edge #3: Data Doesn’t Flow

- The model can “work too well.”
- Data becomes compartmentalized
- Data flows upward, duplication, etc., etc., etc.
- In other words, sometimes even working “right” is “wrong”

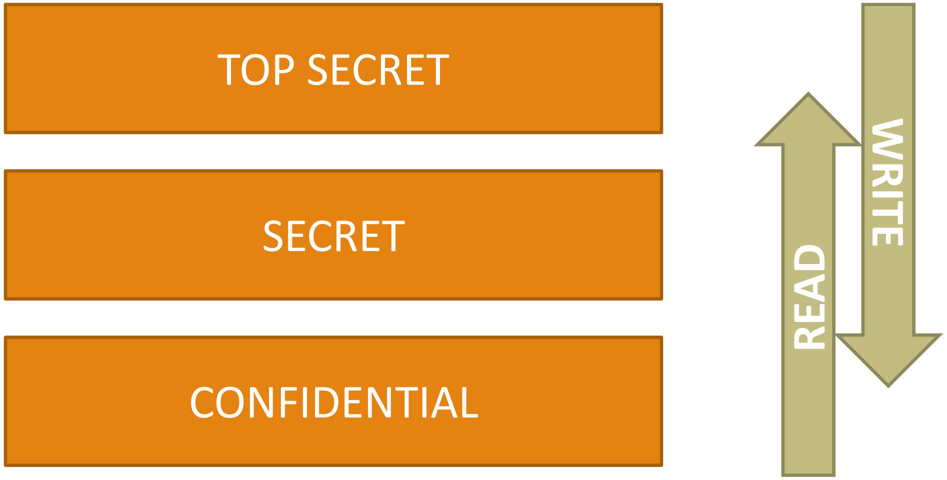
Biba model

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!*



Why BLP or Biba?

- BLP ***primarily enforces*** ***<u>confidentiality</u>***
- Biba ***primarily enforces*** ***<u>integrity</u>***
- Obviously, picking the ***right*** model for a system is crucial
- Remember: many systems fail because the designers ***protect the wrong things*** or protect the right things but ***in the wrong way***.

Domain and Type Enforcement

- DTE assigns a “type” to data objects (e.g., files)
- DTE assigns a “domain” to subjects (e.g., user processes)
- Rules for domain-to-domain and domain-to-type
- Used in SE-Linux and Android
- Powerful, but can be complicated/hard to use
- Perhaps not a real model, but a ***model framework***

Role-Based Access Controls

- RBAC is widely used commercially
- Each user of the system has one or more roles
- Each role has various permissions (can be MAC or DAC)
- Each role’s permissions should be specific/limited
- User may switch roles as needed
- Problems include role-creep, data rot, etc.

Attribute-Based Access Controls

- ABAC includes all of RBAC but adds additional information
- ABAC also includes attributes: time of day, device, etc.
- ABAC is seen as being exceptionally expressive
- Like DTE, can be very complicated and hard to get right

Access Control Principles

- Least privilege
- Separation of duties/concerns
- Accountability/Auditability
- “Conditional” Access

Why Access Controls are Hard

- Hard to model all usage
- For example “Side Channels”
- Another example “Inference Controls”

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