P1L6 Mandatory Access Control

Discretionary Access Control: who ever creates resource is the owner of the resource and can then determine who has access.

Problems with DAC:

-Information flow problem: You cannot control if someone you share a file with will not further share the data contained in it.

- In many organizations, a user does not get to decide how certain type of data can be shared. Typically the employer may mandate how to share various types of sensitive data.

Mandatory Access Control (MAC)

Is not at the user discretion. It solves the problem of information control.

DAC Quiz

In a certain company, payroll data is sensitive. A file that stores payroll data is created by a certain user who is an employee of the company. Access to this file should be controlled with a MAC model as the company must decide who can access it.

MAC Models

-Although users create data in a company, the company must decide who should be able to share it.

-For example: hospitals own patient records and limit their sharing.
HIPAA is a government regulation that controls medical information.

Implementing MAC

What is needed to implement MAC:

- Labels: a key requirement.
    - indicate sensitivity and/or category of data
    - the clearance/need-to-know requirements

The TCB associates labels with each users and object and checks them when access requests are made.

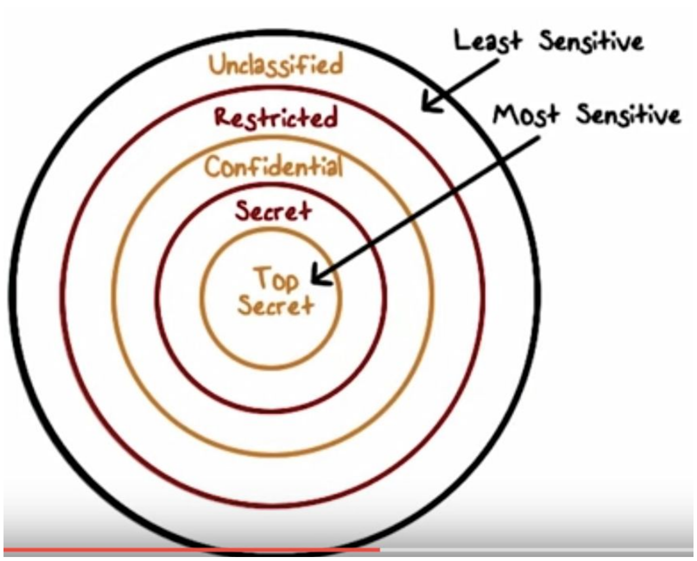
Labels need to be compared.

The exact nature of the labels depends on what kind of model/policy is implemented.

-DoD models include classification/clearance level and a compartment in the label

-Commercial policies are different but use labels to deal with conflict of interest separation.

Implementing MAC



DoD implementation has sensitivity levels for clearance.

Label = (sensitivity level,comparment)

L1 = (TS, {nuclear, chemical})

Providing confidential access to documents (Bell and LaPadula or BLP model).

Comparing Labels

Labels based on clearance: TS > S> C> U

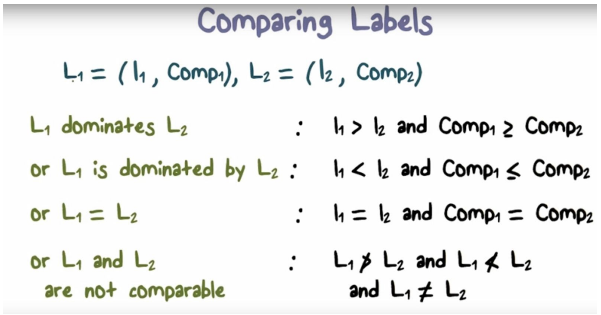
TS > S means top secret is more sensitivity than secret.
When you pick a pair of labels, there is an order between them.

Labels also have a compartment: compartments can be subsets of each other.

Total order: sensitivity
Partial order: compartments

How do we order labels? By sensitivity level and the compartments.

Comparing Labels



Comparing Labels

$L_1 = (l_1, Comp_1)$, $L_2 = (l_2, Comp_2)$

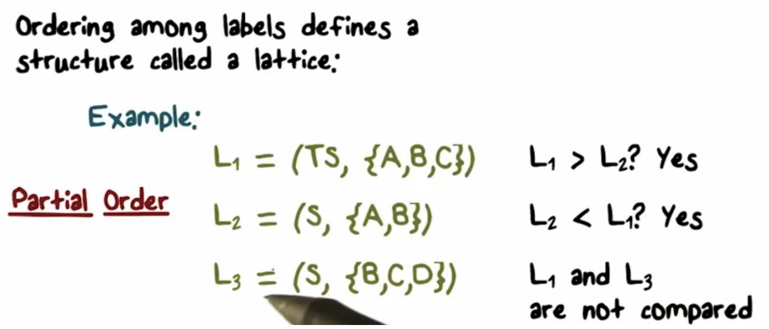
L_1 dominates L_2	:	$l_1 > l_2$ and $Comp_1 \geq Comp_2$
or L_1 is dominated by L_2	:	$l_1 < l_2$ and $Comp_1 \leq Comp_2$
or $L_1 = L_2$	:	$l_1 = l_2$ and $Comp_1 = Comp_2$
or L_1 and L_2 are not comparable	:	$L_1 \ngtr L_2$ and $L_1 \nless L_2$ and $L_1 \neq L_2$

Label domination is based on the level of sensitivity and compartment.

A label has two parts and it is usually the compartments that make it not comparable.

This is a lattice structure with upper bound and lower bound.

Ordering Among Labels



Ordering among labels defines a structure called a lattice:

Example:

Partial Order

$L_1 = (TS, \{A,B,C\})$ $L_1 > L_2$? Yes

$L_2 = (S, \{A,B\})$ $L_2 < L_1$? Yes

$L_3 = (S, \{B,C,D\})$ L_1 and L_3 are not compared

Using Labels for MAC: Confidentiality

Bell and La Padua of BLP Model (developed by DoD)

- Assumes classification of data and clearances for subjects.

Read/write Rules

- Read-down rule (simple security property): User with label L1 can read the document with L2 only when L1 dominates L2

- Write-up rule (star property): User with label L1 can write document with label L2 when L1 is dominated by L2.

Preventing Information Flow with BLP

Information flow can occur in DAC, not in MAC.

Unclassified Documents Quiz

Unclassified documents can be read by anyone, but not written by anyone. The read-down rule applies.

Using the write-up rule BLP allows an unclassified user to write a top-secret document.

Tranquility Principle

The tranquility principle in the BLP model states that classification of a subject or object does not change during a session. This principle is needed to avoid information flow that may violate confidentiality requirements defined by BLP.

Other MAC Models

Biba is dual of BLP :

- Focuses on integrity rather than confidentiality
- Read-up and write-down rules (as opposed to BLP and Read-down and write-up).

Example:

Integrity could be high, medium, or low.
Compartment could be similar to BLP and captures topic(s) of document

Policies for Commerical Environments

- User clearance is not common
- Other requirements exist:
    - data only be accessed by certain applications
    - Separation-of-duty and conflict of interest requirements
        - separation of duty reduces the possibility of fraud

Clark-Wilson Policy

Users should be able to access certain programs.

- The same user cannot execute two programs that require separation-of-duty
- Only the company can decide what files can be accessed by a program.
-

Chinese Wall Policy

Deals with conflict of interest

The user can access any object as long as he/she has not accessed an object from another company in the same conflict class.

Role-based access control (RBAC) is often used in a commercial setting. RBAC is an example of MAC because only the company can decide roles of its employees.

SELinux and SCOMP are both BLP-like model.

Trusted Computing Bases (TCB)

- How do we know TCB can be trusted?
- Secure versus trusted vs high assurance
  - set of all hardware and software are trusted to operate securely
  - required for all other trust in the system security policy

Trusting Software

- Must be functionally correct
    - does what it was designed to do
- Maintains data integrity even for bad input

- Protects disclosure of sensitive data
    - Does not pass data to untrusted software
-Confidence - which means how well does it perform its job
    - This is determined by experts analyzing the program and assure trust
-Statement giving security we expect the system to enforce
    -Do this formally when and where possible

TCB Design Principles

Review of design principles for secure systems also applies here
-Least privilege for users and programs
-Economy
    keep trusted code as small as possible and easier to analyze and test
-Open Design
    security by obscurity does not work
-Complete mediation: every access is checked. All attempts to bypass the security is prevented
-Fail-safe defaults
-Ease of Use
    users avoid security that gets in their way.

Least Privilege Quiz

TCB provides high assurance, but is not a guarantee. So least privilege is useful for damage containment when something goes wrong.

Fail-Safe Default

A home wireless router comes with a setting that does not encrypt traffic unless security settings are explicitly enabled. This violates fail-safe default principle.

How do we Build a TCB:

Key requirements:

- Must implement certain security relevant functions
    - Authentication
    - Access control to files and general objects
    - Mandatory access control (SELinux)
    - Discretionary access control

How to Implement Security Features

-must be tamper proof. The OS must protect itself. How will the OS handle each of the sources are re-allocated
    -security features of trusted OSes
    -object reuse protection
    -disk blocks, memory frames reused
    -process can allocate disk or memory, then look to see what is left behind
    -trusted OS should zero out objects before reuse

- -secure file deletion: overwrite with varying patterns of zeros and ones
- -secure disk destruction: through degaussing, physical destruction

-must have complete mediation of accesses
-must have a trusted path from user to secure system

- -prevents programs from spoofing interface of secure components

-prevents programs from tapping the path (for example: keyloggers)
-Audit log showing object accesses - *only useful if you look at the log*

- -detection of unusual activity of the system

The Gold (Au) Standard

- Authentication
- Authorization
- Audit

Kernel Design

-Security kernel enforces all security mechanisms
-good isolation, small size for verifiability, keeps security code together
-reference monitor controls access to objects (monitors all references to objects)
-Tamperproof (impossible to break or disable)
Un-bypassable (it is always invoked, complete mediation)
Analyzable (small enough to analyze and understand)

Kernel Design

What is included?
-All parts of the OS needed for correct enforcement of security policy

- -handles primitive I/O. clocks, interrupt handling, hardware capabilities, label checking.

-Virtualization - virtual machine provides hardware isolation, logical OS separation.

Revisiting Assurance

Assurance: ways of convincing others that a model, design, and implementation are correct

<u>Methods of assurance validation</u>:

- Testing/penetration testing
    - Penetration testing - try to find vulnerabilities
    - *Demonstrates the existence of problem*
    - *Cannot demonstrate absence of a problem*
    - -Regression Testing: ensure that alterations do not break existing functionality/performance (regression: going backwards)
- Formal verification validation
    - a proof that the system is correct
- Checking that developers have implemented all requirements

Challenges:

-Test case generation

- -shows the existence of a problem
- -make sure there is full code coverage
- -there are an exponential number of different executions
- -different execution environments

Penetration Testing:

-Ethical hackers attempt to defeat security measures

-Cannot demonstrate absence of problem

Formal Verification:

Checking a mathematical specification of program to ensure that security assertions hold.

- Model checking, automated theorem proving
- State variables w/ initial assignment, program specification describing how state changes, boolean predicates over state variables.
- Difficulty: exponential time and space worst case complexity

Security Evaluations

How to do security evaluations:

Trusted Computer System Evaluation Criteria - U.S. Orange Book

Levels of Trust

D<C1<C2<B1<B2<B3<A1

A1 is the highest, D is the lowest.

D: no protection

C: discretionary protection

B: mandatory protection

A: verified protection

C1, C2, B1: security features common to commercial OSes.

Government Security Evaluations

Idea: users specify system needs. Vendors implement solutions. Evaluators determine if vendors met specifications.

EAL (Evaluation assurance level) rates systems:

EAL1 - most basic

EAL7 - most rigorous