School of Electrical Engineering and Computing

SENG2250/6250
SYSTEM AND NETWORK SECURITY
(S2, 2020)

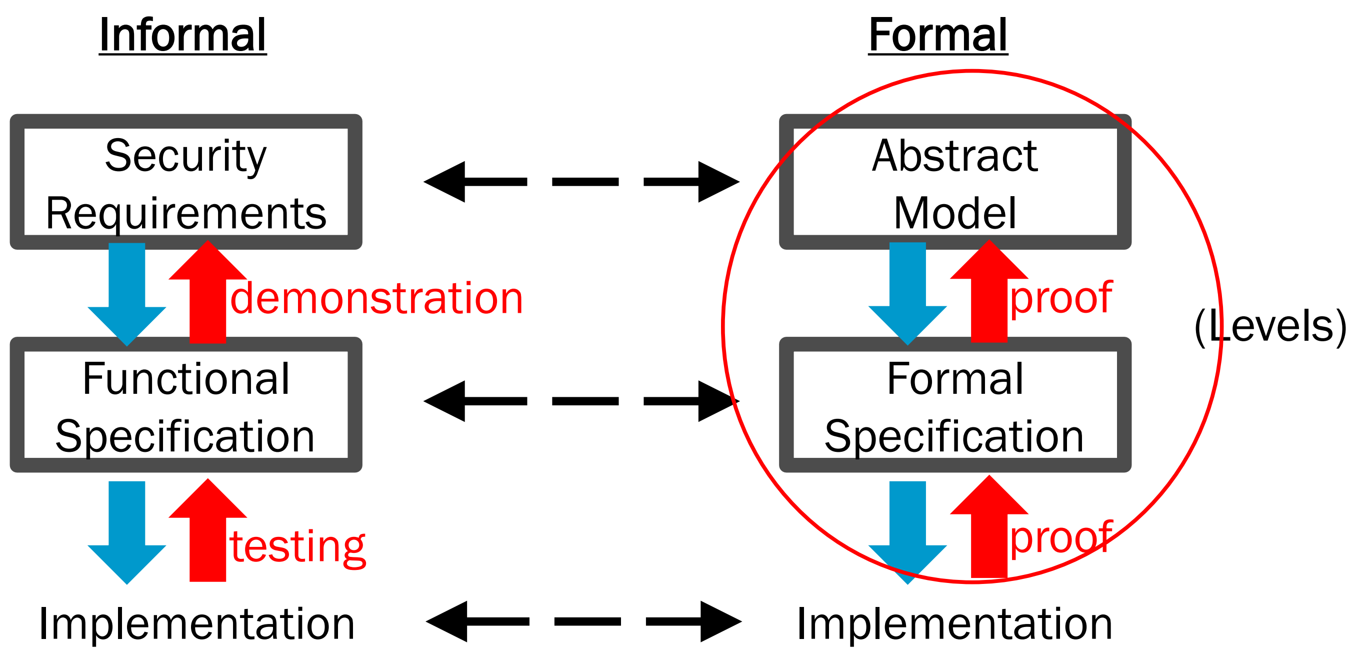
Access Control

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Outline

- Security Models
- Access Control Model
- Access Control Mechanisms
- Access Control Policies

Security System Development Paths



Security Models

- Precise representation of security requirements (Security Policy)
- Characteristics
  - *Simple and Abstract*
  - *Precise and unambiguous*
  - *Deals with security properties*
  - *Does not unduly constrain system functions or implementation*

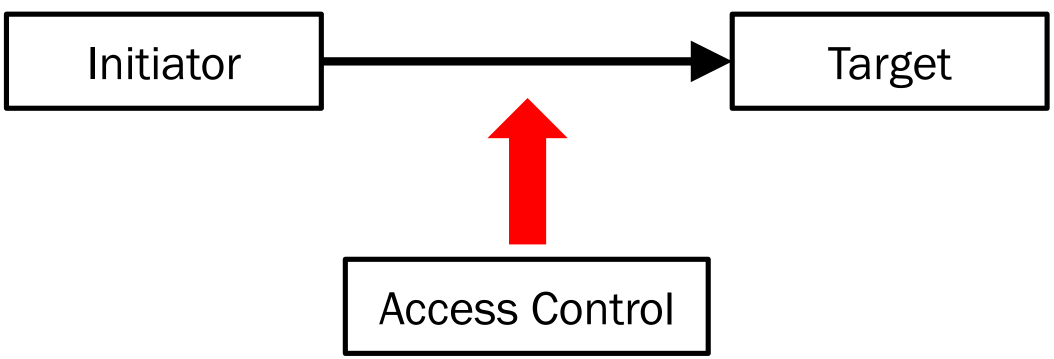
Authentication vs. Authorisation

- Authentication
  - *Is the requesting entity the one it claims to be?*
  - *Who should know about the entity and for what reason?*

- Authorisation
  - *Does the entity have the appropriate privileges for the requested service and operation?*

Access Control

- Access control is a mechanical process, easily implemented by a table and computer process that controls access to information and resources
- Limit who (initiator) can access what (target) in what ways.

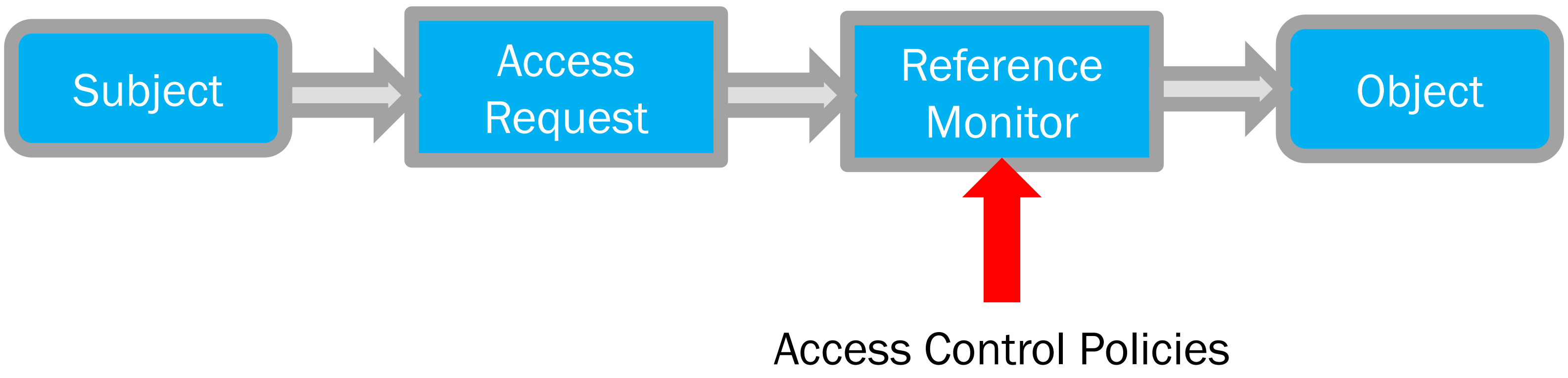


Access Control

- Access Control Information
  - *Individual/Group identities of initiators and targets*
  - *Security labels of initiators and targets*
  - *Roles*
  - *Actions or operations that can be performed*
  - *Contextual information : routing, location, time periods*
- Access Control Policy
  - *Rules that define the conditions under which initiators can access targets*
    - Who can Access What, When and How
- Access Control Authorities
  - *Access Decision and Access Enforcement*

Overview of Access Control Model

- Traditional Access Control Model
  - *Subjects: Users and processes*
  - *Objects: Files, dictionaries, memory, etc.*



Reference Monitor

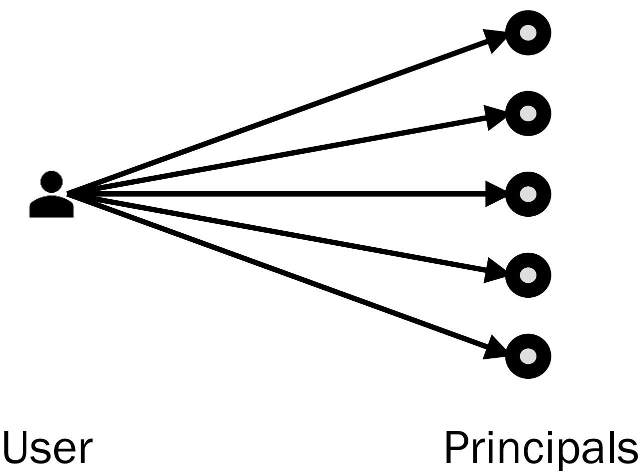
- A set of design requirements which enforce the access control that is always invoked, tamperproof and verifiable.
- It is usually part of operating system.
- Needs effective and efficient translating of access control policies.
  - *Basis for validation*
  - *Policy representation*

Issues with Access Control

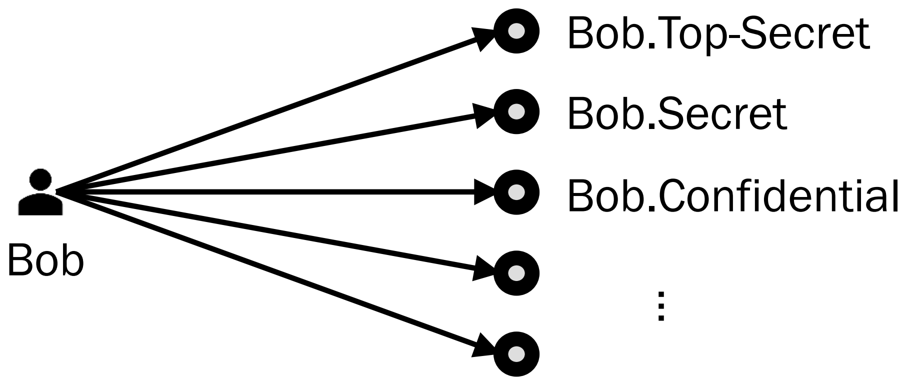
- **Expressiveness:** How to express security policies in terms of access control rules? (at a high level)
- **Efficiency:** Access control decisions occur often, and need to be carried out efficiently
- **Mediation:** How do you know you have not forgotten some access checks?
- **Safety:** How do you know your access control mechanisms match the policy?

Access Control: Users and Principals

- System authenticates the user in context of a particular principal.



Access Control: Users and Principals

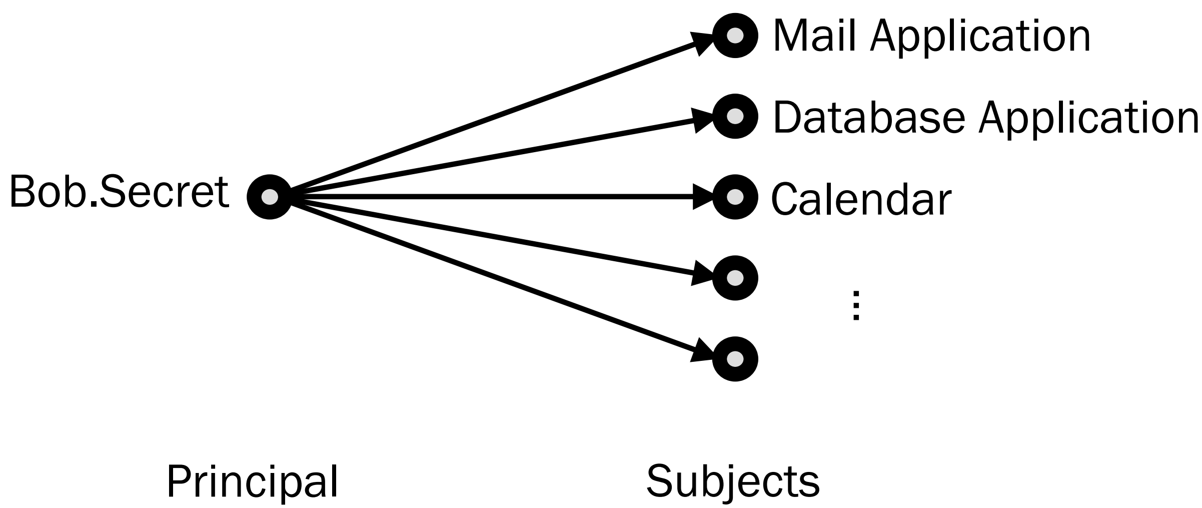


Access Control: Users and Principals

- There should be a one-to-many mapping from users to principals
  - *a user may have many principals, but*
  - *each principal is associated with a unique user*
- This ensures accountability of a user's actions
- Shared accounts (principals) are bad for accountability

Access Control: Subjects and Principals

- A subject is a program (application) executing on behalf of a principal.



Access Control: Subjects and Principals

- Usually (but not always)
  - *each subject is associated with a unique principal*
  - *all subjects of a principal have identical rights (equal to the rights of the invoking principal)*
- This case can be modelled by a one-to-one mapping between subjects and principals

A principal and subject can be treated as identical concepts, while a user should always be viewed as multiple principals.

Objects

- An object is anything on which a subject can perform operations (mediated by rights)
- Usually objects are passive, for example:
  - *File*
  - *Directory (or Folder)*
  - *Memory segment*
- But, subjects can also perform operations on other subjects with operations
  - *Execute, Kill, Suspend, Resume*

Access Control Policy

- A set of rules or statements of access that the system is expected to enforce:
  - *which subject can access which object and how*
  - *which subject can interact with which subject*
  - *Access matrix*
    - Rows : Subjects
    - Columns : Objects (& subjects)
    - Matrix Element : Permissions

Access Control Mechanisms

- Identification
  - *Claim who you are by known identifier (ID).*
- Authentication
  - *Prove yourself by personal secret.*
- Authorisation
  - *Give permission based on ACL.*
- Accountability
  - *Ensure all actions link to authenticated identities.*
  - *E.g., System logs*

Mandatory Access Control

- Use data classification schemes to give users and data owners limited control over access to information resources.

- Access Control Matrix
  - *Access control list (ACL)*
  - *Capability*

Access Control Matrix Model

Objects / Subjects	File 1	File 2	Home Network	Printer
Administrator	rwx	rwx	Allow	Allow
Bob	rw	r	Allow	Deny
Guest	r	-	Deny	Deny
...	...	...	...	...

Access Control List

- Some columns of access control matrix.
- Includes: a list of subjects and the corresponding operations allowed.

Subjects \ Objects	File 1	File 2	Home Network	Printer
Administrator	rwx	rwx	Allow	Allow
Bob	rw	r	Allow	Deny
Guest	r	-	Deny	Deny
...	...	...	...	...

Access control list

Capability

- A row of access control matrix.
- Shows access of a subject to all objects, respectively.

Capability

Subjects \ Objects	File 1	File 2	Home Network	Printer
Administrator	rwx	rwx	Allow	Allow
Bob	rw	r	Allow	Deny
Guest	r	-	Deny	Deny
...	...	...	...	...

ACL vs. Capabilities

- ACL
  - *Good when users manage their own files*
  - *Can set default access right for users.*
  - *Protection is data-oriented.*
  - *Easy to change rights to an object.*
- Capabilities
  - *Easy to delegate.*
  - *Easy to add/delete users.*

Access Control Triples

Subject	Access Right	Object
S_1	rwx	O_1
S_2	r	O_2
S_2	w	O_3
...	...	...

Commonly used in distributed systems, database systems etc.

Access Control Matrix

- Role of the access control matrix:
  - *Manages the rights of subjects to perform actions on objects.*
  - *Manages the rights that subjects can give (or take) to other subjects*

Access Control Matrix

- Contains all relevant information for access control.
- Easy to represent.
- Static representation
- Inefficient
  - *What if there are 2,000 users and 100,000 resources (files, devices, etc), how would be the size of access control matrix?*
  - *Storage and finding issues.*

Access Control

- Discretionary Access Control
  - *Are implemented at the discretion or option of the data user.*
  - *Allows users to pass permissions to any other subjects.*
- Nondiscretionary Access Control
  - *A strictly enforced MACs.*
  - *Managed by a central authority in the organisation.*

Multilevel Security

- Subjects and objects both correspond to security labels.
- Security label for subjects is to set clearances of every user.
- Security label for objects is to set classifications or sensitivity of object.
- The access relationship between security labels of the two types are governed by a series of rules.

Multilevel Security

- A security label binds a set of security attributes to an object or a subject.
  - *Multilevel access control policies.*
- How does it work?
  - *Each object and subject are bound to particular security label (attribute), i.e. clearly defined classifications and clearances.*
  - *When a subject requests to access an object, a label is generated and attached to this request.*
  - *To process a request, an entity (e.g, operating system) compares the request by checking* ***request label*** *and* ***object label*** *with applied policies rules (in access control models) to decide whether the access should be granted or denied.*

Examples

- Classification (sensitive level)
  - *TOP SECRET*
  - *SECRET*
  - *CONFIDENTIAL*
  - *UNCLASSIFIED*
- Multilevel Security Applications
  - *Military*
  - *Government*
  - *Network Firewall*
  - *Database*

Lattice

- A lattice $(L, \leq)$ consists of a set L and a **partial order** $\leq$, for every two element $a, b \in L$, there exists:
  - *A least upper bound* $u \in \mathrm{L}$*.*
  - *A greatest lower bound* $l \in L$*.*
- Formally:
  - *Given* $a \leq u, b \leq u$*, for all* $v \in L$*, we have,*

    $$(a \leq v \wedge b \leq v) \rightarrow (u \leq v)$$
  - *Given* $l \leq a, l \leq b$ *, for all k* $\in L$*, we have,*

    $$(k \leq a \wedge k \leq b) \rightarrow (k \leq l)$$

Example

- $Let\ L = \{0,1,2,3,4,5\}$

- $a = 3, b = 2, u = 3, v \in \{3,4,5\}$
  - u *is the least upper bound.*

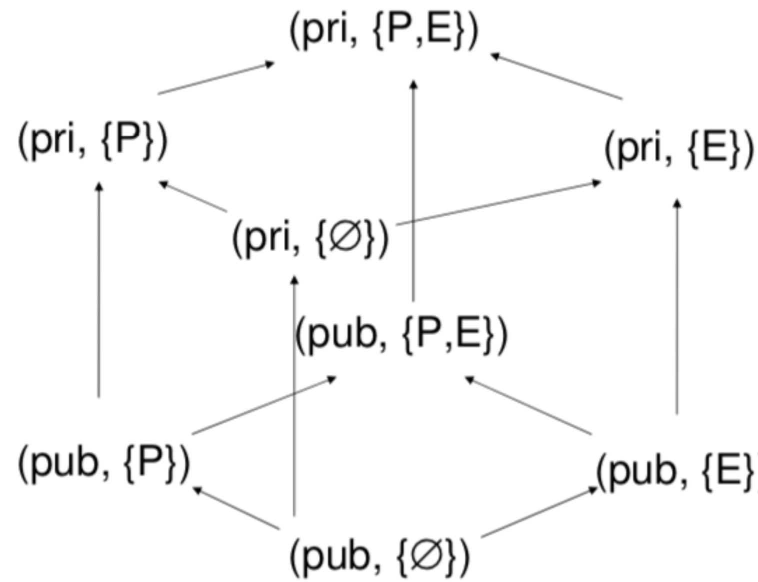
- $a = 3, b = 2, l = 2, k \in \{0,1,2\}$
  - l *is the greatest lower bound.*

Properties of Lattice

- If $a \leq b$, we say that "b **dominates** a" or "a is dominated by b".
    - *Domination can be interpreted as meaning requiring a higher security level.*
    - *If a label (e.g "b") dominates all other labels, it is the **system high**.*
    - *If a label (e.g "a") is dominated by all other labels, it is the **system low**.*
- If $a \leq b$ and $b \leq c$, then $a \leq c$.
- If $a \leq b$ and $b \leq a$, then $a = b$.

Example (Compartments)

- Two levels: {pub, pri}, where pub $\leq$ pri.
- Two Categories: {P, E}
- Some labels may not be comparable.
  - *(pub, {P}) and (pri, {E})*



The Bell-LaPadula (BLP) Model

- The security model works by specifying allowable paths of information flow in a secure system.
- Applies
  - *Lattice with compartments.*
  - *Access control matrix.*
- BLP model is a state machine.
  - *State*
  - *State transitions*
  - *Initial state is secure → all state transitions yield a "secure" state → System is secure if all state transitions are secure.*

The Components

- Let $S = \{s_i\}$ be a set of subjects
  - *Each s_i has a security clearance.*
- Let $O = \{o_i\}$ be a set of objects
  - *Each o_i has a security classification.*
- Let $A = \{a_i\}$ be a set of access operations
  - *E.g,* $A = \{read, write, execute, append\}$.
- Let $C = \{c_i\}$ be a set of classifications/clearances.
  - *E.g.,* $C = \{top\ secret, secret, confidential, unclassified\}$.

The Components

- For all $c_i \in C$, we have $c_i < c_{i+1}$.
    - $E.g., unclassified < confidential < secret < top\ secret$

- For a particular subject s, the security clearance of s is denoted as $\mathrm{c_s} = L(s)$, where $c_s \in C$.

- For a particular subject o, the security classification of o is denoted as $\mathrm{c_O} = L(o)$, where $c_O \in C$.

Read Access

- Information flow from an object o to a subject s.
- Let M be an access control matrix, $\{\mathrm{m_i}\} = M(S, O)$ represents granted permissions.
- Read access is allowed **iff** $L(o) \leq L(s)$ **and** $\mathrm{read} \in \{\mathrm{m_i}\}$.
  - *A subject can read an object if the subject's security label is not **smaller** than that of the object.*
  - *Read permission to the object is contained in your capabilities.*
- It is known as **simple security** condition or "no read up".

Write Access

- Information flow from a subject s to an object o.
- Write access is allowed **iff** $L(s) \leq L(o)$ **and** $\text{write} \in \{\text{m}_\text{i}\}$.
  - *A subject can write to an object if the subject's security label is not larger than that of the object.*
  - *Write permission to the object is contained in subject's capabilities.*
- It is known as *****-property or "no write down".
  - *"Secure" subject cannot write to an "insecure" object.*
  - *E.g., top secret user cannot write to insecure hard drive.*

The Basic Security Theorem

- Let a secure state be one where Simple Security property and *-property hold.
- A state transition is secure if it goes from a secure state to another secure state.
- **The Basic Security Theorem:** Consider a system with a secure initial state and a set of state transitions. If all state transitions are secure, then the system will be always secure.

Example

Top Secret	Alice (read and write)	Personnel Files
Secret	Bob (read)	Email Files
Secret	Coral (write)	Internal Documentation
Unclassified	Eve (read)	Phone Extension Lists

Assume that once given read and write capabilities, they are applied to all files.

- Coral cannot read personnel files, but can write to Email files.
- Bob can read internal documentation, but cannot write to it.

BLP

- A subject may not convey information to a subject at a lower level, *unless* the flow of information accurately reflects the will of an authorized user as revealed by an authorized declassification.

- Prevent unauthorised disclosure of information.

- BLP is a **confidentiality** based access control model.
  - *Protect against unauthorised reading.*

The Biba Model

- A classical **integrity** based access control model.
  - *Protect against unauthorised writing.*
- Much of the basis for Biba is the same as BLP.
- The rules to provide the appropriate policies are, in some sense, the reverse of those for BLP.

No write up, no read down.

- A modified version of Biba was used in Windows 7.

Biba Model

- Let $L(s)_I$ denote the integrity of a subject s.
- Let $L(o)_I$ denote the integrity of a subject o.
- Read access is allowed **iff** $L(s)_I \leq L(o)_I$ **and** $\mathrm{read} \in \{\mathrm{m_i}\}$.
  - *A subject can read an object if the subject's integrity is not greater than that of the object. (No read down)*
  - *Read permission to the object is contained in your capabilities.*
- Write access is allowed **iff** $L(o)_I \leq L(s)_I$ **and** $\mathrm{write} \in \{\mathrm{m_i}\}$.
  - *A subject can write to an object if the subject's integrity is not smaller than that of the object. (No write up)*
  - *Write permission to the object is contained in subject's capabilities.*

Least Privilege

- A subject should have access to the smallest number of objects necessary to perform some task.
- No matter whether the extra information gained would be harmful or useless, additional access should NOT be granted.
- Examples
  - *The access rights should only be granted when they are supposed to.*
  - *Unless a permission is explicitly granted, it should be disallowed.*

References

- C.P. Pfleeger and S.L. Pfleeger. Security in Computing. Prentice Hall , 5th editions, 2015.