UNIVERSITY OF TRENTO

Security Engineering

Lecture 11 – MAC - Security Models
Fabio Massacci

Recaps: Types of Access Control

UNIVERSITY OF TRENTO

- **Discretionary Access Control**
  - Policy decided by individual subjects
  - Access based on identity of subjects
- **Role based Access Control**
  - Policy decided by system
  - Subjects assigned to Roles,
  - (Action,Objects) assigned to Roles
  - Access based on roles activated by subjects
- **Mandatory Access Control**
  - Policy decided by system
  - Subject assigned to security levels (clearance),
  - Object assigned to security labels
  - Access based on matching objects' labels to subjects' clearances
- **Credential based Access Control**
  - Access based on attributes qualifying a subject
    - Essentially "self-service" PIP signed by accredited PAPs

Mandatory Access Control

UNIVERSITY OF TRENTO

- **Organization Access Policy is always MAC**
  - I do not decide who can read the grades of my course
- **Implements**
  - Legislation
  - Commercial Confidentiality – Integrity requirements
  - Paranoia of Board of Directors
  - Pet projects of the above (security holes)
- **Any policy can be specified → enough to have gigantic tables**
  - Objects → Labels
  - Subject → Labels
  - Match: Action x Object x Subject → {True/False}
- **Example on RedHat Security Enhanced Linux**
  - "TE uses a matrix of domains and object types derived from the policy. "
  - `allow httpd_t net_conf_t:file { read getattr lock ioctl };` gives the domain associated with **`httpd`** [=subject] the permissions to read data out of specific network configuration files [=object] such as **`/etc/resolv.conf`**.
- **Example on TSA for flying armed [=object]**
  - Subject [=subject] must be Federal Law Enforcement Officer AND ….
  - Be commissioned to enforce criminal statutes or immigration statutes AND
  - Be authorized by the employing agency to have the weapon in connection with assigned duties:
  - provision of protective duties… OR control of a prisoner… OR …

Security Models

UNIVERSITY OF TRENTO

- **MAC is complicated…**
  - "For Red Hat Enterprise Linux 4 the policy has been designed to restrict **only** a specific list of daemons. **All other processes run in an unconfined state**. This policy is designed to help integrate SELinux into your development and production environment. It is possible to have a much more strict policy, which comes with an increase in maintenance complexity."
- **Security Model = MAC with specific focus**
  - Policy encodes some "default" action in the match function
- **Security Models allows**
  - Simplification of matching process (essential for humans, less for computers)
  - Simplification od administration
  - Formal verification of security

Bell-LaPadula Confidentiality Model

- **BLP is a model that covers the confidentiality aspects of access control**
  - Initially invented for the military
  - OS Multics Operating Systems
  - Implemented in physical security
    - Eg photocopier won't copy document with a "Top Secret" mark
- **Prevents low-security level subjects to read high-security level objects**
- **Consider information flows when a subject reads or alters an object**

Bell-LaPadula Components

- ***S* - set of subjects**
- ***O* - set of objects**
- ***A* - set of access operations**
  - read, write, append, execute
- ***L* - set of partially ordered security levels**
  - Top secret > secret > confidential > unclassified

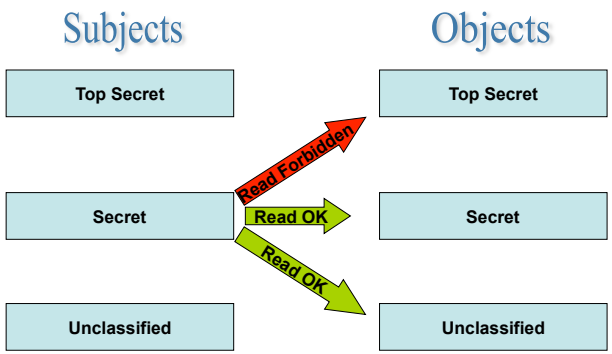
Bell-LaPadula State: assign security levels

- **f_s: S ➔ L**
  - Assign to a subject the maximum security level
- **f_c: S ➔ L**
  - Assign to a subject the current security level
- **f_o: O ➔ L**
  - Assign to an object its security level
- **The security level assigned to a subject is also called security clearance**

Bell-LaPadula properties – ss property

- **A subject can only read an object of less or equal security level**
- **Formally**
  - A system satisfy the simple security property if for every granted read access the security level of the subject *s* dominates the security level of the object *o*
  - $f_o(o) \leq f_s(s)$
- **Also known as *no read-up* security policy**

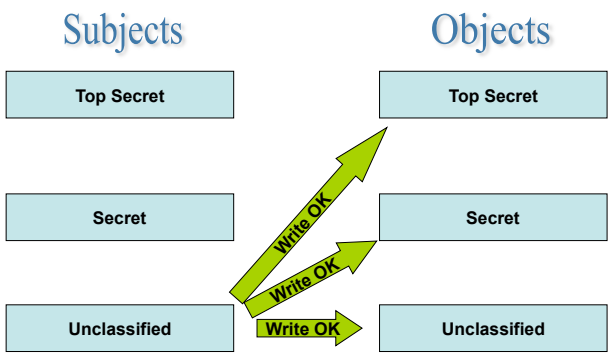
Bell-LaPadula properties: ss property



Bell-LaPadula properties - * property

- **A subject can only write objects of greater or equal security level**
- **Formally**
  - A system satisfies the * property if for every granted write/modify request the security level of the subject *o* dominates the security level of the object *o*
  - $f_s(s) \leq f_o(s)$
- **Also known as no write-down policy**

Bell-LaPadula properties - * property



Bell-LaPadula properties - * property limitation

- **The ∗ - property implies that a high level subject is not able to send messages to a low level subject**
  - How can a general send an email to the secretary?
- **There are two ways to escape from this restriction**
  - Temporarily downgrade a high level subject. This is the reason for the current security level *fc*.
  - Identify a set of trusted subjects, which are permitted to violate the ∗ - property.

The Basic Security Theorem

UNIVERSITY OF TRENTO

- **A state is secure, if all current assignment of permissions to subjects satisfies the ss-property, * - property, and ds-property.**
- **A state transition is secure if it goes from a secure state to a secure state**
- **Basic Security Theorem**
  - If all the transitions are secure and the intial state is secure all the subsequent states will be secure regardaless the input

Tranquillity

UNIVERSITY OF TRENTO

- **McLean: consider a system with an operation *downgrade*:**
  - downgrades all subjects to system low
  - downgrades all objects to system low
  - enters all access rights in all positions of the access control matrix
- **The resulting state is secure according to BLP**
- **Should such a system be regarded as secure?**
  - McLean: no, everybody is allowed to do everything
  - Bell: yes, if *downgrade* was part of the system specification
- **Fact: BLP assumes tranquility, i.e. access control data do not change.**

Limitations of Bell-LaPadula

UNIVERSITY OF TRENTO

- **Restricted to confidentiality**
- **No policies for changing access rights**
  - A general and complete downgrade is secure
  - However, BLP is intended for systems with static security levels
- **BLP contains covert channels**
  - Information flow that is not controlled by the model

Covert Channels

UNIVERSITY OF TRENTO

- ***Covert channels* are information channels that are not controlled by the security mechanism of the system**
- **Information can flow (leak) from a high security level to a low security level**
  - A subject assigned to a low-security level can detect the existence of an high-security level object when it is denied access
  - Sometimes, it is not sufficient to hide only the content of objects. Also their existence may have to be hidden.
- **Telling a subject that a certain operation is not permitted constitutes information flow**

Bell-LaPadula Example

- ESSE3 Clearances
  - Students' Secretariat > Professor > Assistant > Student
- Kate is a teacher for the Security Engineering course → clearance A
  - She can login into the esse3 system as teacher and as student
- Andrea is student enrolled in the Security Engineering course → clearance S
  - He can only login as student

Bell-LaPadula Example

- Kate creates file f1 with P security level
- Andrea creates file f2 with S security level
- Is Kate authorized to read f2?
- Is Kate authorized to write f2?
- Kate creates an exam file f3 with A security level
- Is Andrea authorized to read the f3?

Biba Integrity Model

- State-machine model similar to BLP which focuses on integrity aspects of access control
- Focus on preventing unauthorized modifications of data
- Access permission based on
  - Assignment of subjects and objects to integrity levels
- Prevents information flow from low-integrity levels to high-integrity levels

Biba Integrity Model Components

- S – set of subjects
- O – set of objects
- A – set of access operations
  - modify, observe, execute, invoke
- f_s: S → L
  - Assign to a subject the integrity level
- f_o: O → L
  - Assign to an object its integrity level

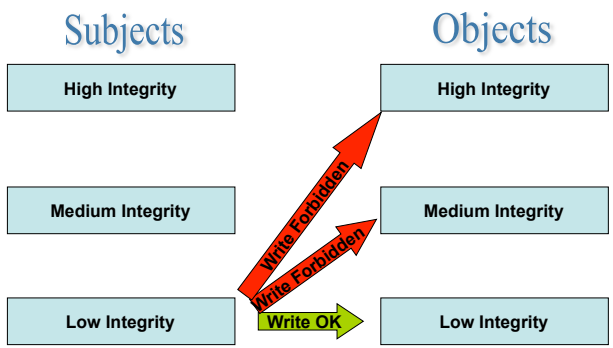
Biba Integrity Model properties: si property

UNIVERSITY OF TRENTO

- **A subject can modify an object only if the integrity level of the subject dominates the integrity level of the object**
- **Formally**
  - A subject *s* can modify (alter) an object *o* if $f_s(s) \geq f_o(s)$
- **Also known as no write-up policy**

Biba Integrity Model properties: si property

UNIVERSITY OF TRENTO



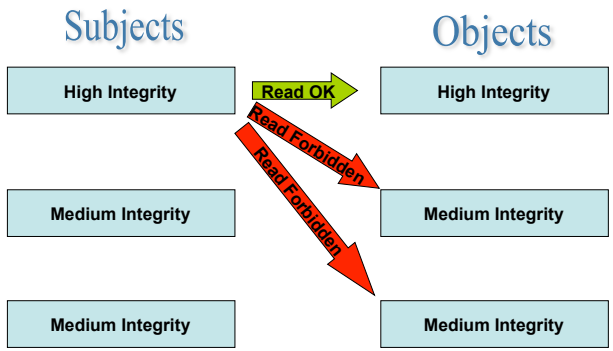
Biba Integrity Model properties: * property

UNIVERSITY OF TRENTO

- **A subject can read an object only if the integrity level of the subject is dominated by the integrity level of the object**
- **Formally**
  - A subject *s* can read (observe) an object *o* if $f_s(s) \leq f_o(s)$
- **Also known as no read-down policy**

Biba Integrity Model properties: * property

UNIVERSITY OF TRENTO



Biba Integrity Model: dynamic integrity properties

- **Automatically adjust subjects and objects assigned integrity levels**
- **Subject Low Watermark Security Policy**
  - A subject s can read (observe) an object o at any integrity level. The new integrity level of the subject s is the greatest lower bound of f_s (s) and f_o (o).
- **Object Low Watermark Security Policy**
  - A subject s can modify (alter) an object o at any integrity level. The new integrity level of the subject s is the greatest lower bound of f_s (s) and f_o(o).

Biba Integrity Model properties: invoke and ring property

- **Invoke Property**
  - A subject is only authorized to invoke subjects (tools) at lower integrity levels
  - Formally
    - A subject s_1 can invoke a subject s_2 if $f_s(s_2) \leq f_s(s_1)$
- **Ring property**
  - A subject s can read objects at any integrity level. It can only modify objects o with $f_o(o) \leq f_s(s)$; it can invoke a subject s' only if $f_s(s) \leq f_s(s')$

Biba Implementation in Vista

- **Vista marks files with an integrity level**
  - Low, Medium, High and System
  - Critical files are assigned System integrity level
  - Other objects are assigned Medium integrity level
  - Internet Explorer is assigned Low integrity level
- **Vista implements the no write-up policy**
  - Files downloaded form IE can read most of the files in Vista file system but cannot write them
  - Limit the damage done by viruses and malwares

Clark-Wilson Integrity Model

- **This model attempts to capture security requirements of commercial applications**
- **Emphasis on integrity**
  - internal consistency: properties of the internal state of a system
  - external consistency: relation of the internal state of a system to the outside world
- **Access permission based on**
  - the assignment of subjects to trusted programs

Clark-Wilson Integrity Mechanisms

UNIVERSITY OF TRENTO

- **Well-formed transactions**
  - A user should only access data through trusted programs
- **Separation of duty**
  - Any person permitted to create or certify a well-formed transaction should not be permitted to perform it

Clark-Wilson Integrity Model Components

UNIVERSITY OF TRENTO

- **Constrained Data Items (CDIs)**
  - Data items subject to strict integrity controls
- **Unconstrained Data Items (UDIs)**
  - Unchecked data items
- **Transformation Procedures (TPs)**
  - System transactions that transforms CDIs from a consistent state to another
- **Integrity Verification Procedures (IVPs)**
  - Check integrity of data items

Clark-Wilson Integrity Model: Certification Rules

UNIVERSITY OF TRENTO

1. **IVPs must ensure that all CDIs are in a valid state at the time the IVPs is run**
2. **TPs must be certified to be valid**
   - Valid CDIs must always be transformed in valid CDIs
   - TPs must be certified to access a specific set of CDIs
3. **Access rules must satisfy any separation of duty requirement**
4. **All TPs must write to an append-only log**
5. **Any TPs taking a UDI as input must either convert it to a CDI or reject the UDI**

Clark-Wilson Integrity Model: Enforcement Rules

UNIVERSITY OF TRENTO

1. **maintain and protect list of TPs and CDIs each TP is certified to access**
   - $(TP_1:CDI_{a1},CDI_{b1},...)$, $(Tp_2:CDI_{a2},CDI_{b2},...)$, $(Tp_3:CDI_{a3},CDI_{b3},...)$
2. **system must maintain and protect the list of UserIDs and TPs each user can execute.**
   - $(UId_1TP_{a1},Tp_{a2},,Tp_{a3})$
   - Maybe further refined by restricting also CDI on a per-user basis
3. **must authenticate each user wishing to execute a TP.**
4. **Only a subject that may certify an access rule for a TP may modify the respective entry in the list.**
   - This subject must not have execute rights on that TP

Credit Card Example

UNIVERSITY OF TRENTO

- **Data (which is CPI, which is UDI?)**
  - Name, Surname
  - Address
  - Credit Card Number
  - PIN Code
  - Account Balance
- **Which is TP?**
  - Issue card (send card to customer's address)
  - Issue PIN
  - Change Name
  - Change address
  - Check credit history
  - Allow debit operation on cc number

Reading Material

UNIVERSITY OF TRENTO

- **Chapters 11 and 12. D. Gollman. Computer Security**
- **Chapter 10 . W. Stallings and L. Brown. Computer Security. Principles and Practices**
- **Chapters 8 and 9. R. Anderson. Security Engineering**
- **David D. Clark and David R. Wilson. A Comparison of Commercial and Military Computer Security Policies in IEEE SSP 1987**