463.6.2 Trusted Computing

Computer Security II
CS463/ECE424
University of Illinois
Based on Slides by Stalling and Brown
Outline

- Bell-LaPadula (BLP)
- Biba
- Clark-Wilson
- Chinese Wall
Multilevel Security (MLS)

An MLS system

• Has system resources (particularly stored information) at more than one security level (i.e., has different types of sensitive resources) and

• permits concurrent access by users who differ in security clearance and need-to-know, but

• is able to prevent each user from accessing resources for which the user lacks authorization.

IETF RFC 2828
Bell-LaPadula (BLP) Model

- Formal model for access control
  - developed in 1970s
- Subjects and objects are assigned a security class
  - a subject has a security clearance
  - an object has a security classification
  - form a hierarchy and are referred to as security levels
    - top secret > secret > confidential > restricted > unclassified
  - security classes control the manner by which a subject may access an object
BLP Model Access Modes

• READ
  – the subject is allowed only read access to the object

• APPEND
  – the subject is allowed only write access to the object

• WRITE
  – the subject is allowed both read and write access to the object

• EXECUTE
  – the subject is allowed neither read nor write access to the object but may invoke the object for execution
No Read Up and No Write Down

- No read up
  - subject can only read an object of less or equal security level
  - referred to as the simple security property (ss-property)

- No write down
  - a subject can only write into an object of greater or equal security level
  - referred to as the *-property
Threat Intuition

- Malicious subject with high-level security clearance
- Flow of information:
  - High-level object-1
  - Low-level object-1
An individual (or role) may grant to another individual (or role) access to a document – based on the owner’s discretion, but – these are constrained by the MAC rules

site policy overrides any discretionary access controls

This is called the ds-property
Current state of system: \((b, M, f, H)\)

- current access set \(b\): triples of \((s, o, a)\)
  - subject \(s\) has current access to object \(o\) in access mode \(a\)
- access matrix \(M\): matrix of \(M_{ij}\)
  - access modes of subject \(S_i\) to access object \(O_j\)
- level function \(f\): security level of subjects and objects
  - \(f_o (O_j)\) is the classification level of object \(O_j\)
  - \(f_s (S_i)\) is the security clearance of subject \(S_i\)
  - \(f_c (S_i)\) is the current security level of subject \(S_i\)
- hierarchy \(H\): a directed rooted tree of objects
the three BLP properties:

- **ss-property**: every \((S_i, O_j, \text{read})\) has \(f_c(S_i) \geq f_o(O_j)\)

- ***-property**: every \((S_i, O_j, \text{append})\) has \(f_c(S_i) \leq f_o(O_j)\) and every \((S_i, O_j, \text{write})\) has \(f_c(S_i) = f_o(O_j)\)

- **ds-property**: every \((S_i, O_j, A_x)\) has \(A_x \in Mij\)

These are used to define the concepts of secure state and secure system.
• The state \((b, M, f, H)\) is **secure** if every element of \(b\) satisfies the three properties.

• A **system** defines a set of transitions that allow changes to the four components of the system, \((b, M, f, H)\).

• A system is **secure** if system transitions on secure states result only in secure states.
1. Get access: Add a triple (subject, object, access-mode) to the current access set b.

2. Release access: Remove a triple (subject, object, access-mode) from the current access set b.

3. Change object level: Change the value of \( f_o(O_j) \) for some object \( O_j \).

4. Change current level: Change the value of \( f_c(S_i) \) for some subject \( S_i \).

5. Give access permission: Add an access mode to some entry of the access permission matrix M.

6. Rescind access permission: Delete an access mode from some entry of M.

7. Create an object: Attach an object to the current tree structure H as a leaf.

8. Delete a group of objects: Detach from H an object and all other objects beneath it in the hierarchy. This renders the group of objects inactive.
BLP Example

*property
BLP Example

ss-property and *-property
BLP Example

“downgrade” in a controlled and monitored manner

“classification creep”: information from a range of sources and levels

Integrity?

“covert channels”: (untrusted) low classified executable data allowed to be executed by a high clearance (trusted) subject
Figure 13.3 Multics Data Structures for MLS

- \( L_u \) = segment security level
- \( L_u \) = user security level
Limitations to the BLP Model

• BLP does not address integrity issues.

• The *-property is difficult to implement
  – Inferences from ordinary actions of higher-level subjects (side channels)
  – Deliberate communications by higher-level subjects (covert channels)

• The BLP formalism does not include de-classification protocols.
Biba Integrity Model: Actions

- **Modify**: To write or update information in an object
- **Observe**: To read information in an object
- **Execute**: To execute an object
- **Invoke**: Communication from one subject to another
Simple integrity: A subject can modify an object only if the integrity level of the subject dominates the integrity level of the object: $I(S) \geq I(O)$.

Integrity confinement: A subject can read an object only if the integrity level of the subject is dominated by the integrity level of the object: $I(S) \leq I(O)$.

Invocation property: A subject can invoke another subject only if the integrity level of the first subject dominates the integrity level of the second subject: $I(S_1) \geq I(S_2)$. 
Clark-Wilson Integrity Model

• Models commercial operations
  – Well-formed transactions
    • A user should not manipulate data arbitrarily
  – Separation of duty among users
    • A person who creates or certifies a well-formed transaction is not allowed to execute it

• Aims to model two key concepts from common commercial practice
  – Well-formed objects and transactions
  – Separation of duty between users
• Constrained data items (CDIs)
  – Subject to strict integrity controls
• Unconstrained data items (UDIs)
  – Unchecked data items
• Integrity verification procedures (IVPs):  
  – Intended to assure that all CDIs conform to some application-specific model of integrity and consistency
• Transformation procedures (TPs):
  – System transactions that change the set of CDIs from one consistent state to another
CDI = constrained data item
IVP = integrity verification procedure
TP = transformation procedure
UDI = unconstrained data item

C1: IVP validates CDI state
C5: TPs validate UDI

E1: CDIs changed only by authorized TP
E2: Users authenticated for TP
E3: Users are authenticated
E4: Authorization lists changed only by security officer
C2: TPs preserve valid state
C3: Suitable separation of duty
C4: TPs write to log

System in some state

Users
Chinese Wall Model

• Use discretionary and mandatory access to address integrity and confidentiality concerns
  – **Subjects**: Active entities that may wish to access protected objects
  – **Information**: Information organized into a hierarchy
    • **Objects**: Individual items of information, each concerning a single corporation
    • **Dataset** (DS): All objects that concern the same corporation
    • **Conflict of interest** (CI) **class**: All datasets whose corporations are in competition
  – **Access rules**: Rules for read and write access
Simple security rule: S can read O only if
• O is in the same DS as an object already accessed by S, OR
• O belongs to a CI from which S has not yet accessed any information

*-property rule: S can write O only if
• S can read O according to the simple security rule, AND
• All objects that S can read are in the same DS as O.
Discussion

• What might be the shape of the next generation of policy models?
  – Don’t be evil?

• Consider new applications and architectures
  – Social media (Facebook v. Twitter etc.)
  – Mobiles (laptops, tablets, smartphones) (BYOD etc.)
  – Internet of Things (IoT)
  – Clouds
  – Crowdsourcing (Waze etc.)