Mandatory Access Control (MAC)

Prof. Ravi Sandhu
Executive Director and Endowed Chair

Lecture 3

ravi.utsa@gmail.com
www.profsandhu.com
Lattice-Based Access Control (LBAC)

Prof. Ravi Sandhu
Executive Director and Endowed Chair

Lecture 3

ravi.utsa@gmail.com
www.profsandhu.com
Denning’s Axioms for Information Flow
Denning’s Axioms

\(< \text{SC}, \rightarrow, \oplus >\)

- **SC** set of security classes
- **\(\rightarrow \subseteq \text{SC} \times \text{SC}\)** flow relation (i.e., can-flow)
- **\(\oplus: \text{SC} \times \text{SC} \rightarrow \text{SC}\)** class-combining operator
Denning’s Axioms

\[ \langle \text{SC}, \rightarrow, \oplus \rangle \]

1. SC is finite
2. \( \rightarrow \) is a partial order on SC
   (i.e., reflexive, transitive, anti-symmetric)
3. SC has a lower bound L such that L \( \rightarrow \) A for all \( A \in \text{SC} \)
4. \( \oplus \) is a least upper bound (lub) operator on SC

Justification for 1 and 2 is stronger than for 3 and 4. In practice we may have a partially ordered set (poset).
Denning’s Axioms Imply

- SC is a universally bounded lattice
- There exists a Greatest Lower Bound (glb) operator \( \bigwedge \) (also called meet)
- There exists a highest security class H
Lattice Structures

Top Secret

Secret

Confidential

Unclassified

can-flow

Hierarchical Classes

reflexive and transitive edges are implied but not shown
Lattice Structures

{ARMY, NUCLEAR, CRYPTO}

{ARMY, NUCLEAR} {ARMY, CRYPTO} {NUCLEAR, CRYPTO}

{ARMY} {NUCLEAR} {CRYPTO}

{}
Lattice Structures

Hierarchical Classes with Compartments

Product of 2 lattices is a lattice
Lattice Structures

The product of 2 lattices is a lattice.
Smith’s Lattice

- With large lattices a vanishingly small fraction of the labels will actually be used
  - Smith's lattice: 4 hierarchical levels, 8 compartments
  - number of possible labels = $4 \times 2^8 = 1024$
    - Only 21 labels are actually used (2%)

- Consider 16 hierarchical levels, 64 compartments which gives $10^{20}$ labels
Extending a POSET to a Lattice

such extension is always possible
BLP Model for Confidentiality
BLP Basic Assumptions

- \( \text{SUB} = \{S_1, S_2, \ldots, S_m\} \), a fixed set of subjects
- \( \text{OBJ} = \{O_1, O_2, \ldots, O_n\} \), a fixed set of objects
- \( \text{R} = \{r, w\} \), a fixed set of rights
- \( \text{D} \), an \( m \times n \) discretionary access matrix with \( D[i,j] \subseteq R \)
- \( \text{M} \), an \( m \times n \) current access matrix with \( M[i,j] \subseteq R \)
BLP Model (Liberal ★-Property)

- Lattice of confidentiality labels $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_p\}$
- Static assignment of confidentiality labels $\lambda: \text{SUB} \cup \text{OBJ} \rightarrow \Lambda$
- $M$, an $m \times n$ current access matrix with
  - $r \in M[i,j] \Rightarrow r \in D[i,j] \land \lambda(S_i) \geq \lambda(O_j)$ simple security
  - $w \in M[i,j] \Rightarrow w \in D[i,j] \land \lambda(S_i) \leq \lambda(O_j)$ liberal ★-property
BLP Model (Strict ★-Property)

- Lattice of confidentiality labels $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_p\}$
- Static assignment of confidentiality labels $\lambda: \text{SUB} \cup \text{OBJ} \rightarrow \Lambda$
- M, an $m \times n$ current access matrix with
  - $r \in M[i,j] \Rightarrow r \in D[i,j] \land \lambda(S_i) \geq \lambda(O_j)$ simple security
  - $w \in M[i,j] \Rightarrow w \in D[i,j] \land \lambda(S_i) = \lambda(O_j)$ strict ★-property
BLP vis a vis Lattices

dominance \geq\ can-flow

Top Secret
Secret
Confidential
Unclassified
BLP vis a vis Lattices

dominance ≥ can-flow

- Unclassified
- Confidential
- Secret
- Top Secret

it is risky to visualize lattices as total orders but it is ok sometimes
BLP vis a vis Lattices

dominance ≥ can-flow

H (High)
L (Low)

often 2 levels suffice to make the main point
-Property

- Applies to subjects not to users
  - Users are trusted (must be trusted) not to disclose secret information outside of the computer system
  - A user can login (create a subject) with any label dominated by the user’s clearance
  - Subjects are not trusted because they may have Trojan Horses embedded in the code they execute
- ★-property prevents deliberate leakage and does not address
  - inference
  - covert channels
- Simple-security and ★-Property do not account for
  - encryption
Dynamic Labels in BLP

- Tranquility: $\lambda$ is static for subjects and objects

- BLP without tranquility may be secure or insecure depending upon the specific dynamics of labeling

- High water mark on subjects: secure
  $\lambda$ is static for objects
  $\lambda$ may increase but not decrease for subjects
  [increase caused by subject trying to read up]

- High water mark on objects: insecure
  $\lambda$ may increase but not decrease for objects
  [increase caused by subject trying to write down]
  $\lambda$ is static for subjects
Biba Model for Integrity
BLP Revisited

dominance $\geq$ can-flow

HS (High Secrecy)

LS (Low Secrecy)
Biba Inverted Flow

HI (High Integrity)

LI (Low Integrity)

dominance \geq \text{can-flow}
Biba and BLP Aligned: BLP Style

One-directional flow is the key point
No need for opposite directions for confidentiality and integrity
Biba and BLP Aligned: Biba Style

- LS (Low Secrecy) ≤ can-flow ≥ HI (High Integrity)
- HS (High Secrecy) ≤ LI (Low Integrity)

One-directional flow is the key point.
No need for opposite directions for confidentiality and integrity.
BLP-Biba Unified Lattice: BLP Style

HS
- LS

HI
- LI

⇒

HS, LI
⇒

HS, HI
⇒

LS, LI
⇒

LS, HI
⇒

Unified

World-Leading Research with Real-World Impact!
BLP versus Biba

- BLP and Biba are fundamentally equivalent and interchangeable

- Lattice-based access control is a mechanism for enforcing one-way information flow, which can be applied to confidentiality or integrity goals

- We will use the BLP formulation:
  - high confidentiality, low integrity at the top
  - low confidentiality, high integrity at the bottom
Lipner’s Lattice

- Lipner's lattice uses 9 labels from a possible space of 192 labels (3 integrity levels, 2 integrity compartments, 2 confidentiality levels, and 3 confidentiality compartments)

- The single lattice shown here can be constructed directly from first principles

- The position of the audit trail at lowest integrity demonstrates the limitation of an information flow approach to integrity

- System control subjects are exempted from the ★-property and allowed to
  - write down (with respect to confidentiality)
  - or equivalently
  - write up (with respect to integrity)
The Chinese Wall Lattice for Separation of Duty
Chinese Wall Policy

- A commercial security policy for separation of duty driven confidentiality
- Mixture of free choice (discretionary) and mandatory controls
- Requires some kind of dynamic labelling
A consultant can access information about at most one company in each conflict of interest class.
Chinese Wall Example

BANKS

A

B

OIL COMPANIES

X

Y
Chinese Wall Lattice

SYSHIGH

A, X
A, Y
B, X
B, Y

A, -
- , X
- , Y
B, -

SYSLOW
Conclusion
MAC or LBAC or BLP (or Biba)

- BLP enforces one-directional information flow in a lattice of security labels
- BLP can enforce one-directional information flow policies for
  - Confidentiality
  - Integrity
  - Separation of duty
  - Combinations thereof
MAC or LBAC or BLP (or Biba)

- BLP enforces one-directional information flow in a lattice of security labels
- BLP can enforce one-directional information flow policies for
  - Confidentiality
  - Integrity
  - Separation of duty
  - Combinations thereof
Covert Channels
Covert Channels

A covert channel is a communication channel based on the use of system resources not normally intended for communication between subjects (processes)
Covert Channels

High User → High Trojan Horse Infected Subject → Low Trojan Horse Infected Subject → Low User

Information is leaked unknown to the high user
Covert Channels

- Information is leaked unknown to the high user

- Property prevents overt leakage of information and does not address covert channels
Side Channels

User 1

User 1’s Subject

User 2

User 2’s Trojan Horse Infected Subject

Information is leaked unknown to the User 1

SIDE CHANNEL
Covert channels require a cooperating sender and receiver.

Side channels do not require a sender but nevertheless information is leaked to a receiver.
Coping with Covert/Side Channels

- Identify the channel
  - close the channel or slow it down
  - detect attempts to use the channel
  - tolerate its existence
Storage Channels

- Also known as Resource Exhaustion Channels

- Given 5GB pool of dynamically allocated memory
  
  - **HIGH PROCESS (sender)**
    
    - bit = 1 \(\Rightarrow\) request 5GB of memory
    
    - bit = 0 \(\Rightarrow\) request 0GB of memory
  
  - **LOW PROCESS (receiver)**
    
    - request 5GB of memory
    
    - if allocated then bit = 0 otherwise bit = 1
Also known as Load Sensing Channels

Given 5GB pool of dynamically allocated memory

- **HIGH PROCESS (sender)**
  
  bit = 1 ⇒ enter computation intensive loop
  
  bit = 0 ⇒ go to sleep

- **LOW PROCESS (receiver)**
  
  perform a task with known computational requirement
  
  if completed promptly then bit = 0 otherwise bit = 1