Chapter 5: Confidentiality Policies

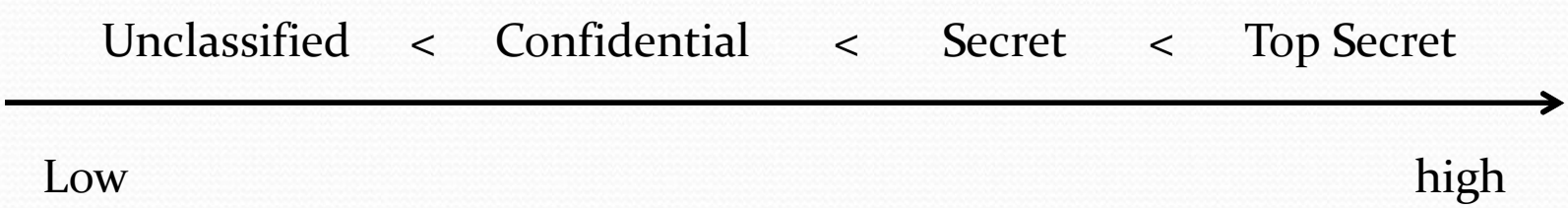
- Overview
  - What is a confidentiality model
- Bell-LaPadula Model
  - General idea
  - Informal description of rules
- A case study (DG/UX)

Confidentiality Policy

- Also called an *information flow policy*
- Goal: prevent the unauthorized *disclosure* of information
  - Deals with information flow
  - Integrity is secondary
  - E.g., military security policy
- Multi-level security models are best-known examples
  - Bell-LaPadula Model basis for many, or most, of these

Bell-LaPadula Model (Preliminary Version)

- Security *levels* arranged in linear ordering, e.g.



- Levels consist of *security clearance*
  - Subjects have *security clearances* $L(s)$
  - Objects have *security classification* $L(o)$

Example

security level	*subject*	*object*
Top Secret	Tamara	Personnel Files
Secret	Samuel	E-Mail Files
Confidential	Claire	Activity Logs
Unclassified	Ulaley	Telephone Lists

- Tamara can read all files
- Claire cannot read Personnel or E-Mail Files
- Ulaley can only read Telephone Lists

Reading Information

- Information flows *up*, not *down*
  - "Reads up" disallowed, "reads down" allowed

- Simple Security Condition (Preliminary Version)
  - Subject *s* can read object *o* iff $L(o) \leq L(s)$ and *s* has permission to read *o*
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no reads up" rule

Writing Information

- Information flows up, not down
  - "Writes up" allowed, "writes down" disallowed
- *-Property (Preliminary Version)
  - Subject *s* can write object *o* iff $L(s) \leq L(o)$ and *s* has permission to write *o*
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no writes down" rule

Basic Security Theorem, Preliminary Version

- If a system is initially in a secure state, and every transition of the system satisfies the simple security condition, preliminary version, and the *-property, preliminary version, then every state of the system is secure

Bell-LaPadula Model, Full Version

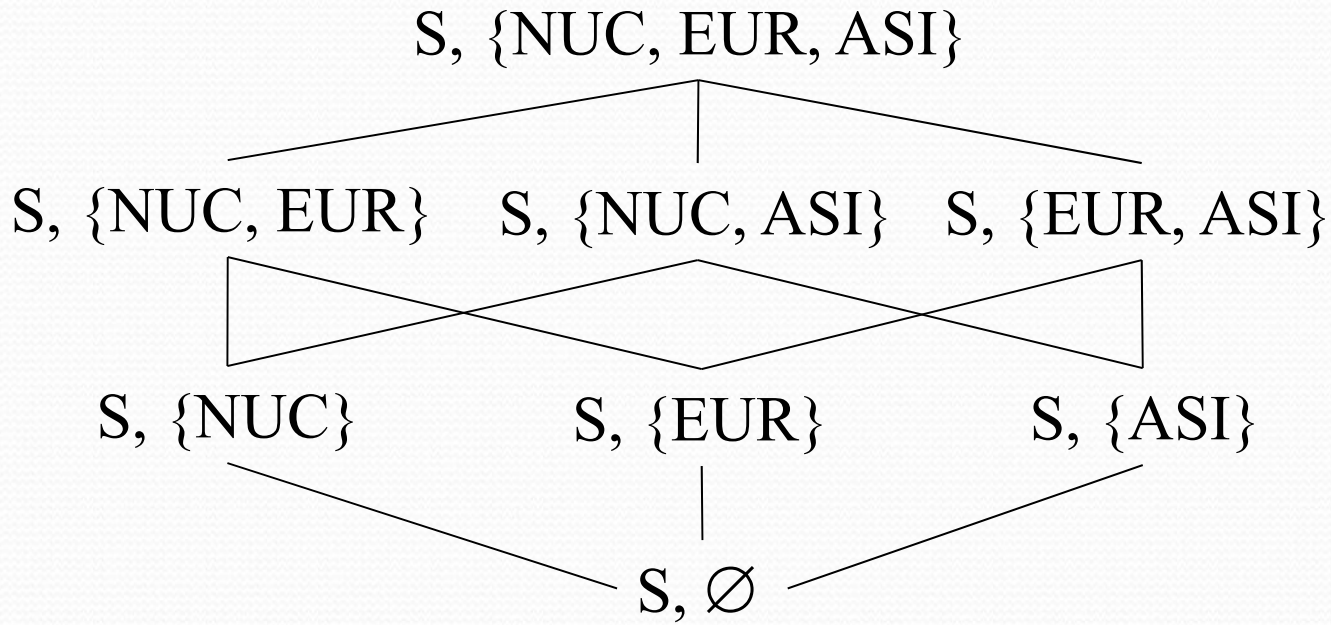
- *Expand* notion of security level to include *categories*
  → the "need to know" principle

- Security level is now (*clearance*, *category set*)
- Examples
  - (Top Secret, { NUC, EUR, ASI })
  - (Confidential, { EUR, ASI })
  - (Secret, { NUC, ASI })

Levels and Lattices

- How do we compare the new security levels?
- The *dom* relationship
  - (A, C) *dom* (A', C') iff $A' \leq A$ and $C' \subseteq C$
- Examples
  - (Top Secret, {NUC, ASI}) *dom* (Secret, {NUC})
  - (Secret, {NUC, EUR}) *dom* (Confidential,{NUC, EUR})
  - (Top Secret, {NUC}) $\neg$*dom* (Confidential, {EUR})
- The *dom* relationship forms a lattice

Lattice Example



- S represents Secret
- The categories are NUC, EUR, and ASI
- The relationship is *dom*

Levels and Ordering

- Security levels partially ordered
  - Any pair of security levels may (or may not) be related by *dom*
- "dominates" serves the role of "greater than" in the preliminary version
  - "greater than" is a total ordering, though

Reading Information

- Information flows *up*, not *down*
  - "Reads up" disallowed, "reads down" allowed
- Simple Security Condition (Full Version)
  - Subject *s* can read object *o* iff *L*(*s*) *dom L*(*o*) and *s* has permission to read *o*
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no reads up" rule

Writing Information

- Information flows up, not down
  - "Writes up" allowed, "writes down" disallowed
- *-Property (Full Version)
  - Subject *s* can write object *o* iff *L*(*o*) *dom* *L*(*s*) and *s* has permission to write *o*
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no writes down" rule

Basic Security Theorem, Full Version

- If a system is initially in a secure state, and every transition of the system satisfies the simple security condition, full version, and the *-property, full version, then every state of the system is secure

A Practical Problem

- Colonel has (Secret, {NUC, EUR}) security level
- Major has (Secret, {EUR}) security level
- Major can talk to colonel ("writes up" or "reads down")
- Colonel cannot talk to major ("no reads up" or "no writes down")
- Clearly does not make sense

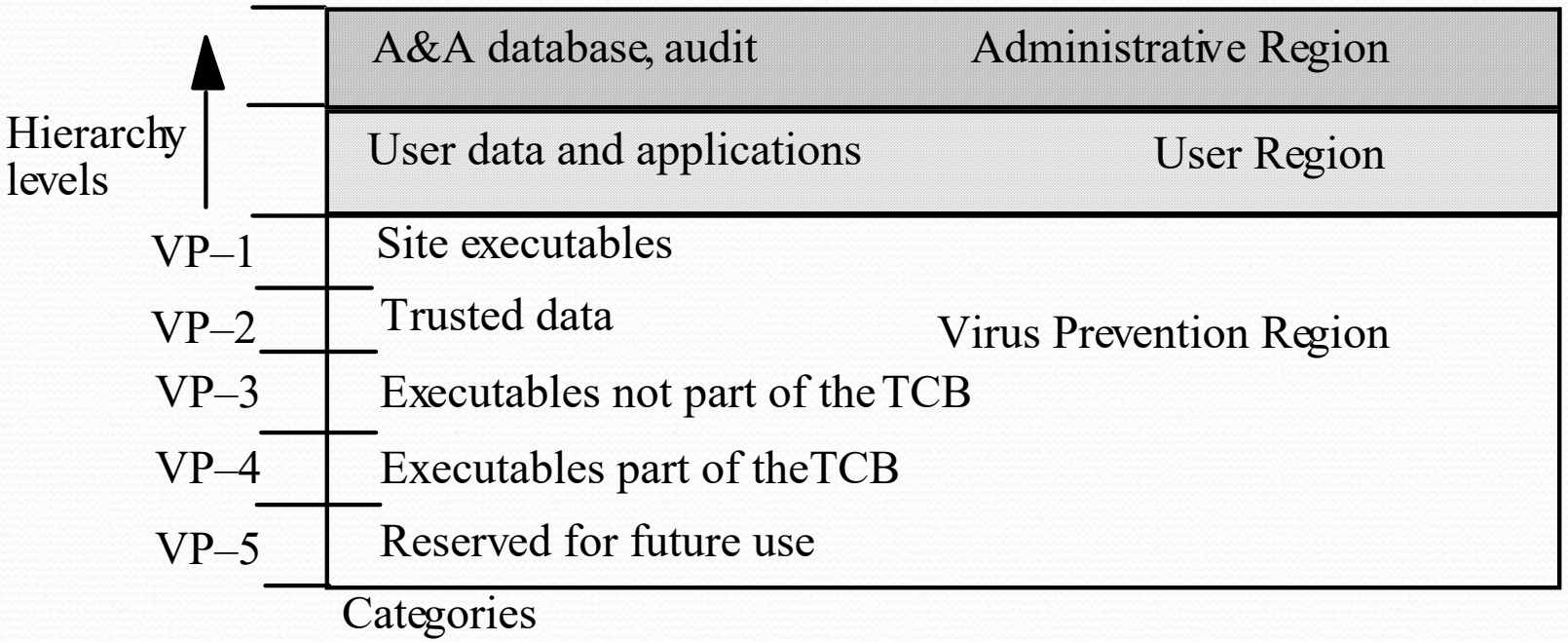
Solution

- Define *maximum* and *current* levels for subjects
  - maxlevel(s) *dom* curlevel(s)
- Allow subjects to adjust their current levels (thus power) as needed
- Example
  - Treat Major as an object (Colonel is writing to him/her)
  - Colonel has *maxlevel* (Secret, { NUC, EUR })
  - Colonel sets *curlevel* to (Secret, { EUR })
  - Now *L*(Major) *dom curlevel*(Colonel)
    - Colonel can write to Major without violating "no writes down"

The Data General/Unix (DG/UX) System

- Provides mandatory access controls
  - Was a Unix operating system
  - MAC label identifies security level
- Initially
  - Subjects assigned MAC label of parent
    - Initial label (at login time) is the label assigned to the user, kept in Authorization and Authentication database
  - Object assigned label at creation, and the label may be
    - Explicit: stored as part of attributes
    - Implicit: determined from parent directory

MAC Regions in the DG/UX Lattice



IMPL_HI is "maximum" (least upper bound) of all levels
IMPL_LO is "minimum" (greatest lower bound) of all levels

Object Labels

- Requirement: every file system object must have a MAC label
  1. Roots of file systems have explicit MAC labels
     - If mounted file system has no label, it gets label of mount point
  2. Object with implicit MAC label inherits label of parent
  3. Creating hard link requires explicit label
     - If target object label implicit, it is made explicit
     - Moving a file makes label explicit

Object Labels (Cont.)

4. Change to directory label makes child labels explicit *before* the change

- Symbolic links are files, and treated as such, so …

5. When resolving symbolic link, label of object is label of target of the link
   - System needs access to the symbolic link itself

Using MAC Labels

- Simple security condition implemented
- *-property not fully implemented
  - Process MAC must equal object MAC
  - Writing allowed only at same security level
- Overly restrictive in practice

MAC Tuples

- Up to 3 MAC ranges (one per region)
- MAC range is a set of labels with upper, lower bound
  - Upper bound must dominate lower bound of range

MAC Range Examples

1. [(Secret, {NUC}), (Top Secret, {NUC})]
2. [(Secret, ∅), (Top Secret, {NUC, EUR, ASI})]
3. [(Confidential, {ASI}), (Secret, {NUC, ASI})]

- (Top Secret, {NUC}) in ranges 1, 2
- (Secret, {NUC, ASI}) in ranges 2, 3
- [(Secret, {ASI}), (Top Secret, {EUR})] not valid range
  - as (Top Secret, {EUR}) ¬*dom* (Secret, {ASI})

Objects and Tuples

- Objects must have MAC labels
  - May also have a MAC tuple
  - If both, tuple overrides label
- Example
  - Paper has MAC range:
    [(Secret, {EUR}), (Top Secret, {NUC, EUR})]

Read Control Based on MAC Tuples

- Process can read object when:
  - Object MAC range (*lr*, *hr*); process MAC label *pl*
  - *pl dom hr*
    - Process MAC label grants read access to upper bound of range
- Example
  - Paper has MAC range: [(Secret, {EUR}), (Top Secret, {NUC, EUR})]
  - Can Peter, with label (Secret, {EUR}), read paper?
    - No, because (Secret, {EUR}) ¬*dom* (Top Secret, {NUC, EUR})
  - Can Paul, with label (Top Secret, {NUC, EUR, ASI}), read paper?
    - Yes, (Top Secret, {NUC, EUR, ASI}) *dom* (Top Secret, {NUC, EUR})

Write Control Based on MAC Tuples

- Process can write object when:
  - Object MAC range (*lr*, *hr*); process MAC label *pl*
  - *pl* ∈ (*lr*, *hr*)
    - Process MAC label grants write access to any label in range
- Example
  - Paper has MAC range: [(Secret, {EUR}), (Top Secret, {NUC, EUR})]
  - Can Peter, with label (Secret, {EUR}), write paper?
    - Yes, because (Top Secret, {NUC, EUR}) *dom* (Secret, {EUR}) and (Secret, {EUR}) *dom* (Secret, {EUR})

Key Points

- Confidentiality models restrict flow of information
- Bell-LaPadula models multilevel security
  - Cornerstone of much work in computer security