Computer Security
CS 426
Lecture 21

The Bell LaPadula Model
Announcements

- October 15: Guest lecture by Prof. Steve Elliott on biometrics

- October 22: Mid-term exam
Bell-LaPadula Model: A MAC Model for Achieving Multi-level Security

• Introduce in 1973

• Air Force was concerned with security in time-sharing systems
  – Many OS bugs
  – Accidental misuse

• Main Objective:
  – Enable one to formally show that a computer system can securely process classified information
What is a Security Model?

- A model describes the system
  - e.g., a high level specification or an abstract machine description of what the system does
- A security policy
  - defines the security requirements for a given system
- Verification shows that a policy is satisfied by a system
- System Model + Security Policy = Security Model
Security Goal of BLP

- There are security classifications or security levels
  - Users/principals/subjects have security clearances
  - Objects have security classifications
- Example
  - Top Secret
  - Secret
  - Confidential
  - Unclassified
- In this case Top Secret > Secret > Confidential > Unclassified
- Security goal (confidentiality): ensures that information do not flow to those not cleared for that level
Approach of BLP

- Use state-transition systems to describe computer systems

- Define a system as secure iff. every reachable state satisfies 3 properties
  - simple-security property, *-property, discretionary-security property

- Prove a Basic Security Theorem (BST)
  - so that one can prove a system is secure by proving things about the system description
The BLP Security Model

- A computer system is modeled as a state-transition system
  - There is a set of subjects; some are designated as trusted.
  - Each state has objects, an access matrix, and the current access information.
  - There are state transition rules describing how a system can go from one state to another
  - Each subject $s$ has a maximal sec level $L_m(s)$, and a current sec level $L_c(s)$
  - Each object has a classification level
Elements of the BLP Model

- Security levels, e.g.: \{TS, S, C, U\}
- \(L_m\): Max Sec. Level
- \(L_c\): Current Sec. Level
- \(L\): Class. Level
- Subjects
  - Trusted Subjects
  - Objects
    - Current Accesses
    - Access Matrix
The BLP Security Model

• A state is secure if it satisfies
  – Simple Security Condition (no read up):
    • S can read O iff $L_m(S) \geq L(O)$
  – The Star Property (no write down): for any S that is not trusted
    • S can read O iff $L_c(S) \geq L(O)$
    • S can write O iff $L_c(S) \leq L(O)$
  – Discretionary-security property
    • every access is allowed by the access matrix

• A system is secure if and only if every reachable state is secure.
STAR-PROPERTY

- Applies to subjects (principals) not to users
- Users are trusted (must be trusted) not to disclose secret information outside of the computer system
- Subjects are not trusted because they may have Trojan Horses embedded in the code they execute
- Star-property prevents overt leakage of information and does not address the covert channel problem
Is BLP Notion of Security Good?

- The objective of BLP security is to ensure
  - a subject cleared at a low level should never read information classified high

- The ss-property and the *-property are sufficient to stop such information flow at any given state.

- What about information flow across states?
BLP Security Is Not Sufficient!

- Consider a system with \( s_1, s_2, o_1, o_2 \)
  - \( f_S(s_1) = f_C(s_1) = f_O(o_1) = \text{high} \)
  - \( f_S(s_2) = f_C(s_2) = f_O(o_2) = \text{low} \)
- And the following execution
  - \( s_1 \) gets access to \( o_1 \), read something, release access, then change current level to low, get write access to \( o_2 \), write to \( o_2 \)
- Every state is secure, yet illegal information exists
- Solution: tranquility principle: subject cannot change current levels
Main Contributions of BLP

• The overall methodology to show that a system is secure
  – adopted in many later works
• The state-transition model
  – which includes an access matrix, subject security levels, object levels, etc.
• The introduction of *-property
  – ss-property is not enough to stop illegal information flow
Other Issues with BLP

• Deal only with confidentiality,
  – does not deal with integrity at all

• Does not deal with information flow through covert channels
Overt (Explicit) Channels vs. Covert Channels

- Security objective of MLS in general, BLP in particular
  - high-classified information cannot flow to low-cleared users
- Overt channels of information flow
  - read/write an object
- Covert channels of information flow
  - communication channel based on the use of system resources not normally intended for communication between the subjects (processes) in the system
Examples of Covert Channels

- Using file lock as a shared boolean variable
- By varying its ratio of computing to input/output or its paging rate, the service can transmit information to a concurrently running process
- Covert channels are often noisy
- However, information theory and coding theory can be used to encode and decode information through noisy channels
More on Covert Channels

- Covert channels cannot be blocked by *-property
- It is generally very difficult, if not impossible, to block all cover channels
- One can try to limit the bandwidth of covert channels
- Military requires cryptographic components be implemented in hardware
  - to avoid trojan horse leaking keys through covert channels
More on MLS: Security Levels

- Used as attributes of both subjects & objects
  - clearance & classification
- Typical military security levels:
  - top secret $\geq$ secret $\geq$ confidential $\geq$ unclassified
- Typical commercial security levels
  - restricted $\geq$ proprietary $\geq$ sensitive $\geq$ public
Security Categories

• Also known as compartments
• Typical military security categories
  – army, navy, air force
  – nato, nasa, noforn
• Typical commercial security categories
  – Sales, R&D, HR
  – Dept A, Dept B, Dept C
Security Labels

- Labels = Levels × P (Categories)
- Define an ordering relationship among Labels
  - \((e_1, C_1) \leq (e_2, C_2)\) iff. \(e_1 \leq e_2\) and \(C_1 \subseteq C_2\)
- This ordering relation is a partial order
  - reflexive, transitive, anti-symmetric
  - e.g., \(\subseteq\)
- All security labels form a lattice
An Example Security Lattice

- levels={top secret, secret}
- categories={army, navy}
The need-to-know principle

- Even if someone has all the necessary official approvals (such as a security clearance) to access certain information, they should not be given access to such information unless they have a *need to know*: that is, unless access to the specific information necessary for the conduct of one's official duties.
- Can be implemented using categories and/or DAC
Readings for This Lecture

• Wikipedia
  • Bell-LaPadula model

• David E. Bell: Looking Back at the Bell-La Padula Model
Coming Attractions …

• Trusted Operating Systems and Assurance