CS 556 – Computer Security
Fall 2012

Dr. Indrajit Ray
Email: indrajit@cs.colostate.edu

Department of Computer Science
Colorado State University
Fort Collins, CO 80523, USA
LIPPNER’S INTEGRITY MATRIX MODEL
Model Goals

- Relevant in the commercial sector
- Tries to control the production program
  - Integrity of the object is of prime importance
**Requirements in Production Program**

- Users will not write their own programs, but will use existing production programs and databases.
- Programmers will develop and test programs on a nonproduction system.
  - If they need access to actual data, they will be given production data via a special process, but will use it on their development system.
A special process must be followed to install a program from the development system onto the production system.

The special process must be controlled and audited.

The managers and auditors must have access to both the system state and the system logs that are generated.
Model Contributions

- Separation of duty.
- Separation of function.
- Auditing.
Separation of Duty

- If two or more steps are required to perform a critical function, at least two separate persons should perform the steps.
  - Moving a program from the development stage to the production system is an example.
  - A separate “installer” is more likely to catch a problem than the original developer.
  - If developer wants to subvert production data with a corrupt program, a separate certifier will be able to catch it.
Separation of Function

- The same person should not perform two or more different functions in the system.
  - Developers do not develop new programs on production systems because of the potential threat to production data.
  - Developers do not process production data on the development system.
Auditing

- Auditing is the process of analyzing systems to determine what actions took place and who performed them.
- This is needed for recovery and accountability.
INTEGRITY MATRIX MODEL
Model Overview

- Combined BLP with Biba to address the concerns of the commercial sector.
- Defined two security levels
  - Audit Manager (AM): system audit and management functions are at this level.
  - System Low (SL): any process can read information at this level.
- Defined five compartments
Lipner’s Compartments

- Development (D): production programs under development and testing but not yet in production state.
- Production Code (PC): production process and programs
- Production Data (PD): data covered by the integrity policy
- System Development (SD): system programs under development but not yet in production use
- Software Tools (T): programs provided on the production system not related to the sensitive or protected data.
User to Security Level Assignment

- Ordinary users will use production code to modify production data; their clearance is (SL, \{PC, PD\}).
- Application developers need tools for developing their programs and to a category for the programs that are being developed; their clearance is (SL, \{D, T\}).
- System programmers use tools to develop system programs; their clearance is (SL, \{SD, T\}).
System Managers and auditors need high system clearance as they must be able to access all logs; their clearance is (AM, \{D, PC, PD, SD, T\}).

System controllers must have ability to downgrade code once it is certified for production so other entities cannot write to it; thus the clearance is (SL, \{D, PC, PD, SD, T\}) and downgrade privilege.
Object to Security Level Assignment

- Objects are assigned to security level based on who should access them.
- Objects that may be altered have two compartments
  - That of the data itself.
  - That of the program that may alter it.
Object to Security Level Assignment

- Development code/test data – (SL, \{D,T\})
- Production code – (SL, \{PC\})
- Production data – (SL, \{PC, PD\})
- Software tools – (SL, \{T\})
- System programs – (SL, \{\phi\})
- System programs in modification – (SL, \{SD, T\})
- System and application logs – (AM, \{appropriate category\})
Security Level for Logs

- All logs are append only. By the BLP *-property, their classes must dominate those of the subjects that write to them. Hence each log will have its own category.
  
  - Simplest way to prevent log compromise is to put all logs at the highest security level.
**Lipner’s Lattice**

**Lippner’s Integrity Matrix Model**

**Integrity Matrix Model**

**The Combined Model**

- **S**: System Manager
  - **O**: Audit Trail

- **S**: System Control

- **S**: Application Programmers
  - **O**: Development Code & Data

- **S**: System Programmers
  - **O**: System Code in Development

- **S**: Repair
  - **S**: Production Users
  - **O**: Production Data

- **O**: Repair Code

- **O**: Production Code

- **O**: Tools

- **O**: System Program

---

Dr. Indrajit Ray, Computer Science Department

CS 556 - Computer Security - © 2012 Colorado State University – 18 / 25
Lipner’s Lattice

- 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 Star property and allowed to
  - Write up with respect to integrity or
  - Write down with respect to confidentiality
Lipner’s Use of BLP

- With the two security levels and five compartments, the model satisfies the five commercial security requirements.
- However, it allows little flexibility in special purpose software
  - A program for repairing an inconsistent or erroneous production database cannot be an application-level software.
- To remedy this Lipner integrates his BLP model with the Biba model (Combined Model)
THE COMBINED MODEL
Integrity Levels and Compartments

- Three integrity levels
  - System Program (ISP): the classification for system programs
  - Operational (IO): the classification for production programs and development software
  - System Low (ISL): the classification at which users log in

- Two integrity compartments:
  - Development (ID): development entities
  - Production (IP): production entities
New Security Compartments

- The previous security category T (tools) allowed application developers and system programmers to use the same programs without being able to alter these programs.
  - The Integrity compartments now distinguish between production and development so the compartment T can be eliminated.

- Production code and production data can be collapsed into the same compartment.
- This results in 3 security compartments in the combined model.
New Security Compartments

- Production (SP): production code and data
- Development (SD): same as previous security compartment Development (D)
- System Development (SSD): same as previous compartment System Development (SD)
Lippner’s Model’s Implication on LBAC

- In practice we will always need to violate the direction of information flow.
- This is done by downgrading objects by passing them through a sanitizing process which is typically a trusted subject.
  - For example – declassifying confidential objects.