Foundations of Computer Security

Lecture 23: Lipner's Model

Dr. Bill Young
Department of Computer Sciences
University of Texas at Austin

Commercial Integrity Constraints

Recall that Steve Lipner (Microsoft) described some integrity concerns you might find in a commercial data processing environment:

1. Users will not write their own programs, but use existing production software.
2. Programmers develop and test applications on a nonproduction system, possibly using contrived data.
3. Moving applications from development to production requires a special process.
4. This process must be controlled and audited.
5. Managers and auditors must have access to system state and system logs.

Can we use our existing modeling mechanisms to build a secure system that addresses such constraints?

Lipner's Integrity Matrix Model

Lipner devised his Integrity Matrix Model to handle those concerns via a combination of BLP and Biba Integrity.

There are two confidentiality levels:

Audit Manager (AM): system audit and management.

System Low (SL): all other processes.

In addition there are three confidentiality categories:

Production (SP): production code and data.

Development (SD): programs under development.

System Development (SSD): system programs in development.

Lipner's Model (Cont.)

In addition to the confidentiality constraints, we also impose integrity constraints. There are three integrity classification (highest to lowest):

System Program (ISP): system software

Operational (IO): production programs and development software

System Low (ISL): user level behavior

and two integrity categories:

Development (ID)

Production (IP)

Subject Levels

Security levels (both confidentiality and integrity) are assigned to subjects based on their roles in the organization and their need to know.

User Role	**Confidentiality**	**Integrity**
Ordinary users	$(SL, \{SP\})$	$(ISL, \{IP\})$
Application developers	$(SL, \{SD\})$	$\{ISL, \{ID\})$
System programmers	$(SL, \{SSD\})$	$\{ISL, \{ID\})$
System managers/auditors	$(AM, \{SP, SD, SSD\})$	$\{ISL, \{IP, ID\})$
System controllers	$(SL, \{SP, SD\})$ and downgrade	$\{ISP, \{IP, ID\})$

Here *downgrade* means the ability to move software (objects) from development to production.

Object Levels

Security levels (both confidentiality and interity) are assigned to objects based on who should access them.

Object type	**Confidentiality**	**Integrity**
Development code/test data	$(SL, \{SD\})$	$\{ISL, \{ID\})$
Production code	$(SL, \{SP\})$	$\{IO, \{IP\})$
Production data	$(SL, \{SP\})$	$\{ISL, \{IP\})$
Software tools	$(SL, \emptyset)$	$\{IO, \{ID\})$
System programs	$(SL, \emptyset)$	$\{ISP, \{IP, ID\})$
System programs in modification	$(SL, \{SSD\})$	$\{ISL, \{ID\})$
System and application logs	$(AM, \{categories\})$	$\{ISL, \emptyset)$

Lipner's Model

Some questions:

1. Can an ordinary user utilize a system program? Modify it?
2. Can a system programmer use production software? Modify it?
3. Why is that special downgrade permission required? Could it be done with BLP and Biba alone?

The answers:

1. That depends on what "utilize" means. If "utilize" means "read" then he can read, but not modify.
2. Neither.
3. Moving objects from the development to production world means changing their labels. There's no obvious way to do that in BLP or Biba.

Lessons

- Lipner developed a hybrid policy using both BLP and Biba's Strict Integrity to address commercial integrity concerns.
- Some modifications relating to tranquility were required to allow moving applications from the development to production domains.
- The result is acceptable but not entirely intuitive. Perhaps an entirely new modeling paradigm would be preferable.

Next lecture: Clark-Wilson Model