

CS590U  
Access Control: Theory and Practice

Lecture 10 (Oct 2<sup>nd</sup>)  
Security Models

## What is a security model?

- An access control model / mechanism
- A formal system model
  - A formal mathematical model that functions as a concise and precise description of the behavior desired of the security-relevant portions of the system
- An interface model
  - specifying restrictions on a system's interface (usually its input/output relation) that are sufficient to ensure confidentiality

2

## Why Formal System Models?

- Demonstrating that systems are secure by showing
  - systems that enforce the model are secure
  - the design to which the implementation corresponds enforces the model
  - the implementation conforms to the design

3

## Bell & LaPadula Model (1)

- Each subject has a *clearance*
- Each object has a *classification*
- Each subject has a *current security level*
  - which may not exceed the subject's clearance
- Four model of access
  - read, append, execute, read-write

4

## Bell & LaPadula Model (2)

- A state is secure if it satisfies
  - Simple security condition (clearance)
    - no read access to higher-level objects
  - \*-property (current security level)
    - no append-access to lower-level objects
    - read-write access only to same-level objects
    - no read-access to higher-level objects

5

## Bell & LaPadula (3)

- A set of rules governing state changes
  - get access
  - release access
  - give access
  - rescind access
  - create object
  - delete object
  - change security level

6

## Bell & LaPadula Model (4)

- Trusted subjects
  - may violate \*-property
  - trusted not to compromise security

7

## Why Trusted Subjects?

- Strict enforcement of \*-property is impractical
  - a user may need to extract an UNCLASSIFIED paragraph from a CONFIDENTIAL document
  - a user may create a message at TOP SECRET, and after entering the message, decide that the message should be SECRET

8

## Difficulties When Applying Bell & LaPadula to MMS

- prohibition of write-downs
- absence of multilevel objects
  - e.g., paragraphs in a message may have different classifications from the message
- no structure for application-dependent security rules
  - e.g., only allows users with release authority to invoke the release operation

9

## Issues When Using Trusted Subjects

- Adding trusted subjects made the security policy enforced difficult to understand
- The design assumption that "user confirmation of security-relevant operations would prevent security violations" does not hold

10

## Other Experiences of Using Bell & LaPadula

- Multics-AIM
  - If a user operating at the TOP SECRET level wishes to send an UNCLASSIFIED message to another user operating at the SECRET level, Multics-AIM requires that the message be treated as though it were TOP SECRET.

11

## Lessons Learned

- The axioms of the Bell-LaPadula model are overly restrictive
- Trusted subjects are not restrictive enough
- Using axioms and trusted subjects together complicate the security policy being enforced

12

## Quote of Carl Landwehr's Survey

- "In sum, the principal problems with the (Bell & LaPadula) model are not in the things it allows but in those it disallows: many operations that are in fact secure will be disallowed by the model. Systems based on the model are then faced with the choice between obeying the model but imposing severe constraints on functionality and allowing the desired functions by relying heavily on trusted processes."

13

## Lattices

- A security label consists of
  - a sensitivity level
  - a set of compartments
- $\langle \text{lev}_1, C_1 \rangle \leq \langle \text{lev}_2, C_2 \rangle$ 
  - if  $\text{lev}_1 \leq \text{lev}_2$  &  $C_1 \subseteq C_2$

14

## Information-Flow Models

- Three kinds of channels through which information could flow
  - read/write
  - storage channels, shared temporary storage
    - e.g., writing tem
  - covert channels

15

## Denning's Information Flow Model

- $\text{FM} = \langle N, P, SC, \oplus, \rightarrow \rangle$ 
  - $N = \{a, b, \dots\}$  is a set of logical storage objects
  - $P = \{p, q, \dots\}$  is a set of processes
  - $SC = \{A, B, \dots\}$  is a set of security classes
    - each object is bound to a class, using either static binding or dynamic binding
    - each process may also be bound to a class

16

## Denning's Information Flow Model (continued)

- $\oplus$  is an associative and commutative binary operator in SC (lub)
  - The class of  $a+b$  is (a's class  $\oplus$  b's class)
  - SC is closed under  $\oplus$
- $\rightarrow$  is a binary relation on SC, known as the flow relation

17

## Denning's Information Flow Model (continued)

- A flow model FM is secure if and only if execution of a sequence of operations cannot give rise to a flow that violates the relation  $\rightarrow$ 
  - assuming that  $\rightarrow$  is transitive, security of individual operations implies security of arbitrary sequences of operations

18

## Denning's Information Flow Model (continued)

- $\langle SC, \rightarrow \rangle$  forms a universally bounded lattice and  $\oplus$  is the least upper bound operator
  - $\langle SC, \rightarrow \rangle$  is a partially ordered set
  - SC is finite
  - SC has a lower bound L such that  $L \rightarrow A$  for all  $A \in SC$

19

## Examples of Covert Channels

- CPU usage
- Disk usage

20

## Covert channels

- Ways to analyze covert channels
  - tracing information-flow paths of programs
  - checking programs for shared resources that can be used to transfer information
  - checking systems for clocks that can be used for timing channels

21

## Covert channels

- Basically impossible to eliminate, can only try to make the rate of information leakage slower
  - still an issue, e.g., leaking an encryption key

22

## The Noninterference Model for Deterministic Systems

- Basic idea: high-level actions (inputs) have no effect on what low-level users can see (outputs)
- One needs a formal definition of the inputs and outputs of a system
  - if one removes any high-level inputs, the low-level user sees the same thing

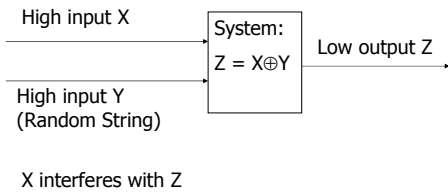
23

## The Noninterference Model

- For deterministic systems, if input sequence X is noninterfering with output sequence Y and X is independent of the input from other users, then  $I(X,Y)=0$ , here  $I(X,Y)$  is the *mutual information* between X and Y

24

## Problem of Noninterference



25

## Notion of Nondeducibility

- Basic idea:
  - whatever the low-level users see is compatible with any high-level input
- Equivalent to noninterference in deterministic systems with two users

26

## Problem with Nondeducibility

- Consider a system that receives messages and outputs their encryption and generates random strings when no message is inputted
- Nondeducibility is satisfied even when outputs are not encrypted
- Problem: probabilities are not taken into account.

27

## Limits of Formal Security Models (Dorothy Denning)

- Security models have theoretical limits.
- Security models based on strict mathematical properties can lead to systems that are totally unusable
- Building systems from rigorous mathematical security models is extremely time-consuming and costly

28

## Limits of Formal Security Models (Continued)

- Security models and formal methods do not establish security. Systems are hacked outside the models' assumptions
- Provable security, even if were achievable, is not a panacea

29