

CS590U

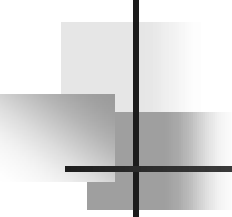
Access Control: Theory and Practice

Lecture 8 (February 3)

Noninterference

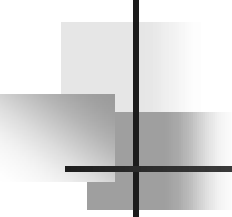
Security Policies and Security Models

J.A.Goguen and J.Meseguer
Oakland'1982



Distinction Between Models and Policies

- A model describes the system
 - e.g., a high level specification or an abstract machine description of what the system does
 - this paper uses a state transition systems with focus on operations and outputs
- A security policy
 - defines the security requirements for a given system
- Verification shows that a policy is satisfied by a system



Four Stages in Defining Security

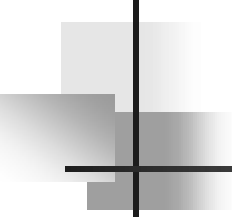
1. Determine the security needs of a given community
 2. Express those needs as a formal requirement
 3. Model the system which that community is (or will be) using
 4. Verify that systems in the model satisfies the requirement
- Maybe switch steps 2 & 3, as the formal security requirement will be based on the model; maybe an iterative process.

An Abstract System Model

- S : set of states
- U : set of subjects
- SC : set of state commands
- Out : set of all possible outputs
- $do: S \times U \times SC \rightarrow S$
 - $do(s, u, c) = s'$ means that at state s , when u performs command c , the resulting state is s'
- $out: S \times U \rightarrow Out$
 - $out(s, u)$ gives the output that u sees at state s
- $s_0 \in S$ initial state

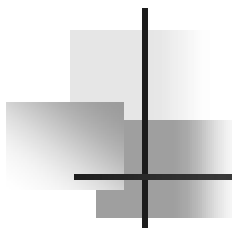
The Additional Capability Component

- Capt: set of capability tables
- CC: set of capability commands
- out: $S \times \text{Capt} \times U \rightarrow \text{Out}$
- do: $S \times \text{Capt} \times U \times \text{SC} \rightarrow S$
- cdo: $\text{Capt} \times U \times \text{CC} \rightarrow \text{Capt}$
 - decides how the capability table is updated
- s_0, t_0 : initial state and capability table



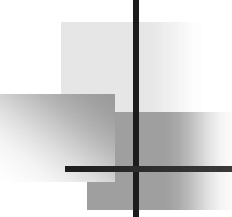
Summary of the Modeling Aspect

- The system is modeled as a state-transitional system
- Changes state by subjects executing commands
- Each state has an output for each subject
- Implicit assumptions:
 - Initial state of the system does not contain any sensitive information
 - Information comes into the system by commands
 - Only way to get information is through outputs



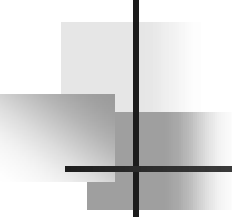
Security Policies

- A security policy is a set of noninterference assertions
- Definition of noninterference: Given two group of users G and G' , we say G does not interfere with G' if for any sequence of commands w , what users in G' can observe after executing w is the same as what users in G can observe after executing $P_G(w)$, which is w with command initiated by users in G removed.
- Similar in spirit to the notion of zero-knowledge in cryptography
 - if what one can see with high inputs is the same as what one sees without high inputs, no high information is leaked



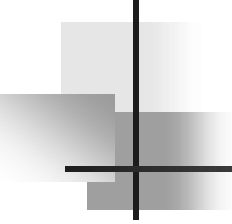
Usage Examples

- Information flow within a programs
 - certain variables are noninterfering with other variables
- Safety in automated trust negotiation
 - how to say that a negotiator's behavior does not leak information about its sensitive attributes to entities not authorized to know that information



Evaluation of The Non-Interference Policy

- The policy definition is elegant and natural
 - focuses on policy objective, rather than mechanism, such as BLP
- The model is useful for some applications, but may be difficult to apply to real world systems
 - how to model a system that BLP intends to model, with files storing sensitive information?



End of Lecture 8

- Next lecture
 - Covert channel