L3: Security Policies

Hui Chen, Ph.D.
Dept. of Engineering & Computer Science
Virginia State University
Petersburg, VA 23806

Acknowledgement

- Many slides are from or are revised from the slides of the author of the textbook
  - Matt Bishop, Introduction to Computer Security, Addison-Wesley Professional, October, 2004, ISBN-13: 978-0-321-24774-5. Introduction to Computer Security @ VSU's Safari Book Online subscription
  - http://nob.cs.ucdavis.edu/book/book-intro/slides/

Outline

- Review and Overview
- Confidentiality Polices
- Integrity Policies
- Availability Policies
- Case Study

Security Policy and Mechanism

- Security policy
  - A statement of what is allowed and what is not allowed
  - Example
    - A student may not copy another student's homework
  - Can be informal or highly mathematical
- Security mechanism
  - A method, tool, or procedure for enforcing security policy
  - Technical and non-technical
    - A homework electronic submission system (e.g., Blackboard) enforces who may read a homework submission

Security Policy

- Security policy
  - Partitions system states
    - Authorized (or secure) states
      - States the system can enter
    - Unauthorized (non-secure) states
      - Security violation if the system enters any of these states
  - Sets the context in which we can define a *secure* system.

Secure System

- A secure system is a system that starts in an authorized state and cannot enter an unauthorized state

Transfer Funds

- Processes P and Q

```
……
int fromAccount, toAccount, amountToTransfer;
……
transferFunds(fromAccount, toAccount, amountToTransfer);
……
```

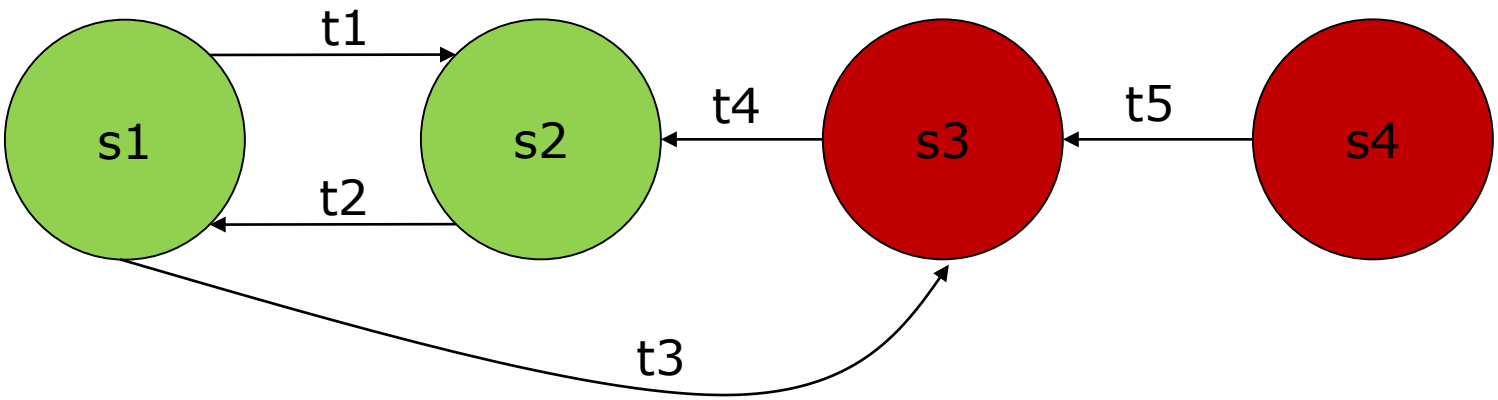
	fromAccount	toAccount	amountToTransfer	P	Q
P	rw	rw	rw	rwxo	
Q	r	r	r		rwxo

Secure System and Policy

- Secure System
  - A system is secure under one policy may not be secure under a different policy

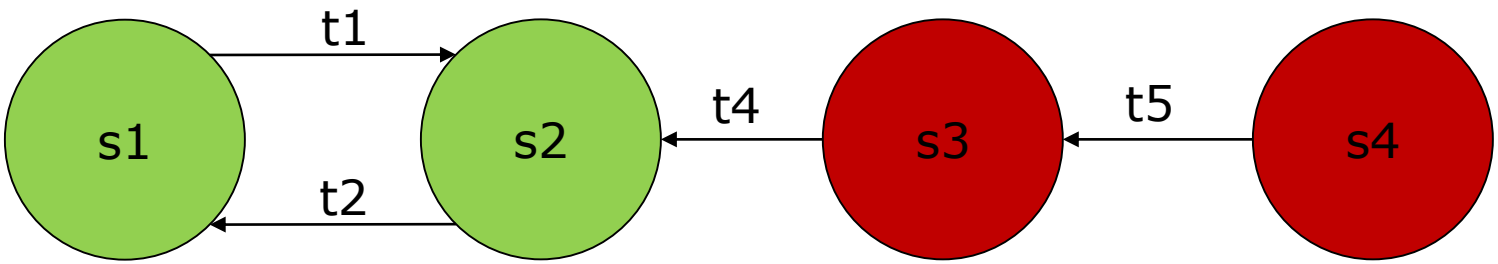
Example

- An example
  - A = {s1, s2}, UA = {s3, s4}
- Is the system secure?



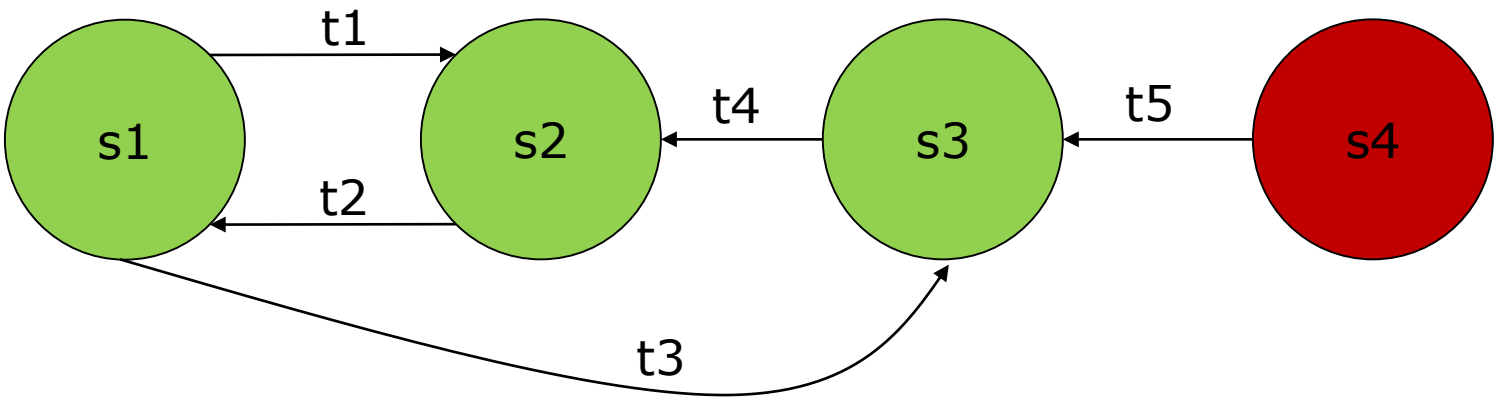
Example

- An example
  - A = {s1, s2, s3}, UA = {s4}
- Is the system secure?



Example

- An example
  - A = {s1, s2, s3}, UA = {s4}
- Is the system secure?



Breach of Security

- A breach of security occurs when a system enters an unauthorized state

Security Properties

- Confidentiality
- Integrity
- Availability

Confidentiality

- X set of entities, I information
- I has *confidentiality* property with respect to X if no $x \in X$ can *obtain* information from I
- I can be disclosed to others
- Example:
  - X set of students
  - I final exam answer key
  - I is confidential with respect to X if students cannot obtain final exam answer key

Confidentiality

- Example:
  - X set of students
  - I final exam answer key
  - I is confidential with respect to X if students cannot obtain final exam answer key

Confidentiality

- Implies
  - Information must not be disclosed to some set of entities
  - May be disclosed to others
- Membership of X is often implicit
  - States entities that have access to information I
  - X is implicitly those entities that are not authorized to have such an access

Confidentiality

- Example:
  - Only course instructors can obtain the answer keys to the courses' final exam
- What is X and what is I?

Integrity

- X set of entities, I information
- I has *integrity* property with respect to X if all $x \in X$ *trust* information in I
- Types of integrity:
  - trust I, its conveyance and protection (data integrity)
  - I information about origin of something or an identity (origin integrity, authentication)
  - I resource: means resource functions as it should (assurance)

Availability

- X set of entities, I resource
- I has *availability* property with respect to X if all $x \in X$ can *access* I
- Types of availability:
  - traditional: x gets access or not
  - quality of service: promised a level of access (for example, a specific level of bandwidth) and not meet it, even though some access is achieved

Security Policies

- Confidentiality policy
- Integrity policy
- Quality of service (Availability) policy

Policy Model

- Abstract description of a policy or class of policies
- Focus on points of interest in policies
  - Security levels in multilevel security models
  - Separation of duty in Clark-Wilson model
  - Conflict of interest in Chinese Wall model

Types of Security Policies

- Military (governmental) security policy
  - Policy primarily protecting confidentiality
- Commercial security policy
  - Policy primarily protecting integrity
- Confidentiality policy
  - Policy protecting only confidentiality
- Integrity policy
  - Policy protecting only integrity
  - Transaction-oriented integrity security policies

Integrity and Transactions

- Begin in consistent state
  - "Consistent" defined by specification
- Perform series of actions (*transaction*)
  - Actions cannot be interrupted
  - If actions complete, system in consistent state
  - If actions do not complete, system reverts to beginning (consistent) state

Trust

- Confidentiality policies place no trust in objects
- Integrity policies defines the level of trust
- Example 1

Trust

- Confidentiality policies place no trust in objects
- Integrity policies defines the level of trust
- Example 1
  - Administrator installs patch
    1. Trusts patch came from vendor, not tampered with in transit
    2. Trusts vendor tested patch thoroughly
    3. Trusts vendor's test environment corresponds to local environment
    4. Trusts patch is installed correctly

Trust

- Confidentiality policies place no trust in objects
- Integrity policies defines the level of trust
- Example 2

Trust

- Confidentiality policies place no trust in objects
- Integrity policies defines the level of trust
- Example 2
  - Trust in Formal Verification
    - Gives formal mathematical proof that given input *i*, program *P* produces output *o* as specified
    - Suppose a security-related program *S* formally verified to work with operating system *O*
    - What are the assumptions?

Trust

- Confidentiality policies place no trust in objects
- Integrity policies defines the level of trust
- Example 2
  - Trust in Formal Verification
    - Proof has no errors
      - Bugs in automated theorem provers
    - Preconditions hold in environment in which S is to be used
    - S transformed into executable S' whose actions follow source code
      - Compiler bugs, linker/loader/library problems
    - Hardware executes S' as intended
      - Hardware bugs (e.g., Pentium CPU's f00f bug)

Types of Access Control

- Discretionary Access Control (DAC, IBAC)
  - individual user sets access control mechanism to allow or deny access to an object
- Mandatory Access Control (MAC)
  - system mechanism controls access to object, and individual cannot alter that access
  - sometimes called *rule-based access control*
- Originator Controlled Access Control (ORCON)
  - originator (creator) of information controls who can access information

Case Studies

- Policy disallows cheating
  - Includes copying homework, with or without permission
- CS class has students do homework on computer
- Anne forgets to read-protect her homework file on the computer
- Bill copies it
- Who cheated?
  - Anne, Bill, or both?

Who Violated Security Policy?

- Bill cheated
  - Policy forbids copying homework assignment
  - Bill did it
  - System entered an unauthorized state
    - Unauthorized state: Bill having a copy of Anne's assignment
- If not explicit in computer security policy, certainly implicit
  - Not credible that a unit of the university allows something that the university as a whole forbids, unless the unit explicitly says so

Who Violated Security Policy?

- Anne did not protect her homework
  - Not required by security policy
- She did not breach security

Who Violated Security Policy?

- Let us change the policy
  - The university disallows cheating, which is defined to include copying another student's work with or without permission. The university mandates that every student must read-protect her or his work files on university computers.
- The policy said students had to read-protect homework files,
  - Anne did not do this
  - Anne also breached security (violated security policy)

Mechanisms

- Entity or procedure that enforces some part of the security policy
  - Access controls (like bits to prevent someone from reading a homework file)
  - Disallowing people from bringing CDs and floppy disks into a computer facility to control what is placed on systems

Reading Assignment

- Section 4.5

Summary

- Policies describe *what* is allowed
- Mechanisms control *how* policies are enforced
- Trust underlies everything

Exercise L3-1

- Exercises 1 of Exercises 4.8 in page 59 of the textbook

Exercise L3-2

- Exercises 5(d) of Exercises 4.8 in page 60 of the textbook

Homework 3

- Exercises 5(a), 5(b), and 5(c) of Exercises 4.8 in page 59 of the textbook