# Security Policies


Dr. Ahmad Almulhem

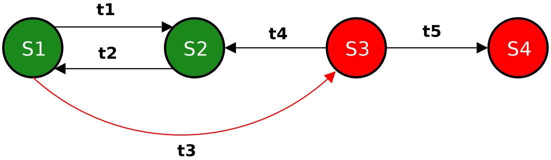
Computer Engineering Department, KFUPM


Spring 2008

# Part I

## Overview

# Security Policies




- A security policy defines “secure” for a system
- A security policy partitions system states into:
  1. Authorized (secure) states
     The system should stay in these states
  2. Unauthorized (nonsecure) states
     If the system enters any of these states, its a security breach
- Secure system
  - Starts in authorized state
  - Never enters unauthorized state

# Definitions

## Definition (security policy)

A *security policy* is a statement that partitions the states of the system into a set of authorized, or secure, states and a set of unauthorized, or nonsecure, states.

## Definition (secure system)

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

## Definition (breach of security)

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

# Confidentiality

- $X$ set of entities, $I$ some information
- $I$ has confidentiality property with respect to $X$ if no $x \in X$ can obtain information about $I$
- $I$ 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

# Integrity

- $X$ set of entities, $I$ some information
- $I$ has integrity property with respect to $X$ if all $x \in X$ trust information in $I$
- Types of integrity:
  1. trust $I$, its conveyance and protection (data integrity)
  2. $I$ information about origin of something or an identity (origin integrity, authentication)
  3. $I$ 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:
  1. Traditional: $x$ gets access or not
  2. 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 (e.g. a server for a book-store vs medical center)

# Security Policies

- A security policy considers all relevant aspects of confidentiality, integrity, and availability
  - Who can access information? (confidentiality policy)
  - What are the authorized ways to modify information? (integrity policy)
  - What services must be provided and its QoS? (availability policy)
- Statement of security policy can be formal (provable) or informal
- Implicit policies can be confusing (using mechanisms)

# Policy Models

## Definition (Policy Model)

- Policy Model is an abstract description of a policy or class of policies
- Focus on points of interest in policies
  - Security levels in multilevel security models (Bell-LaPadula Model)
  - 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
  - Privacy issues
- Commercial security policy
  - Policy primarily protecting integrity(e.g. banks)
- Confidentiality policy
  - Policy protecting only confidentiality
- Integrity policy
  - Policy protecting only integrity

# Integrity and Transactions

- Integrity policies using the notion of transactions (e.g. databases)
- 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

- Trust and assumptions underlies security policies and mechanisms
- Trust some assumptions will hold

**Example (Administrator installs patch)**

Question: Does the security improved?

Answer: Depends on the following assumptions:

- Trusts patch came from vendor, not tampered with in transit
- Trusts vendor tested patch thoroughly
- Trusts vendors test environment corresponds to local environment
- Trusts patch is installed correctly

# Example: 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?
  1. Proof has no errors (bugs in automated theorem provers)
  2. Preconditions hold in environment in which S is to be used
  3. S transformed into executable S' whose actions follow source code (Compiler bugs, linker/loader/library problems)
  4. Hardware executes S' as intended (Hardware bugs)

# Types of Access Control

- Discretionary Access Control/Identity-Based Access Control (DAC, IBAC)
  - individual user sets access control mechanism to allow or deny access to an object
- Mandatory Access Control/Rule-Based Access Control (MAC, RBAC)
  - system mechanism controls access to object, and individual cannot alter that access
- Originator Controlled Access Control (ORCON)
  - originator (creator) of information controls who can access information (owner does not)

# Key Points

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

# Part II

# Confidentiality Policies

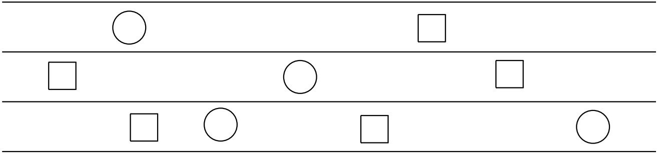
# Confidentiality Policies

- Goal: prevent the unauthorized disclosure of information
  - Deals with information flow
  - Integrity and availability are secondary goals
- Multi-level security models are best-known examples
  - Bell-LaPadula Model basis for many, or most, of these

# Bell-LaPadula Model (BLP)

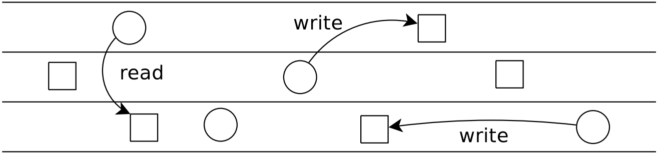
- Introduced by Elliot Bell and Leonard LaPadula in early 1970s.
- Security levels arranged in linear ordering
  - Top Secret: highest
  - Secret
  - Confidential
  - Unclassified: lowest
- Security levels correspond to information sensitivity
- Subjects have security clearance L(s)
- Objects have security classification L(o)

# Level Diagrams (Amoroso 94)



- Circles are subjects
- Squares are objects

# Read and Write

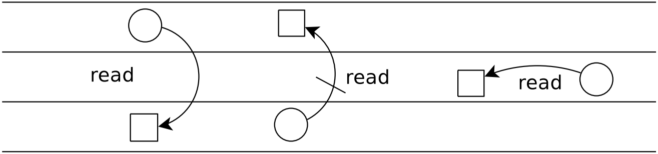



- Arrows represent read and write operations
- An arrow originates from a subject to an object

# Read and Write (Information flow)

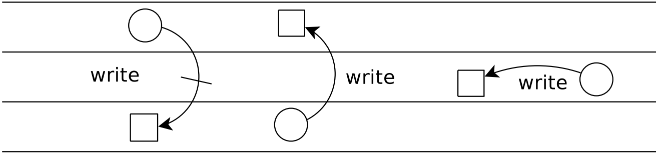

- Read and write operations cause information to flow between subjects and objects
- In write operations, information flows from subject to object
- In read operations, information flows from object to subject

# Reading (BLP)




- Information flows up, not down
  - "Reads up" disallowed, "reads down" allowed
- Simple Security Property
  - Subject $s$ can read object $o$ iff $L(o) \leq L(s)$ and $s$ has permission to read $o$
- Combines mandatory control (relationship of security levels) and discretionary control (the required permission)
- Sometimes called "no reads up" rule (NRU)

# Writing (BLP)




- Information flows up, not down
  - "Writes up" allowed, "writes down" disallowed
- *-Property
  - Subject $s$ can write object $o$ iff $L(s) \leq L(o)$ and $s$ has permission to write $o$
- Combines mandatory control (relationship of security levels) and discretionary control (the required permission)
- Sometimes called "no writes down" rule (NWD)

# Example

| **security level** | **subject** | **object** |
|---|---|---|
| top secret | Bell | personal files |
| secret | James | email files |
| confidential | Frank | activity logs |
| unclassified | Tom | telephone lists |

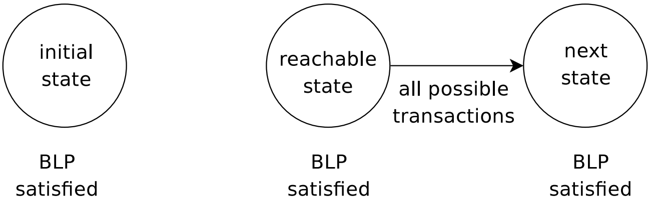
- Bell can read all files
- Frank cannot read personnel or email files
- Tom can only read telephone lists

# Basic Security Theorem

## Theorem

*If a system is initially in a secure state, and every transition of the system satisfies the simple security property, and the *-property, then every state of the system is secure*

# BLP Model Induction




1. Show NRU and NWD are satisfied in all initial states
2. Show all actions can not change a state satisfying NRU/NWD into one which does not

# Key Points

- Confidentiality models restrict flow of information
- Bell-LaPadula model multilevel security
  - Cornerstone of much work in computer security
  - NRU and NWD rules

Part III

# Integrity Policies

# Integrity Policies

- Very different from confidentiality policies
- Concerned more with accuracy of data than their disclosure
  - e.g. banks
- Mostly used in commercial and industrial environments

# Biba Integrity Model

- Introduced by Ken Biba in the mid 1970s
- Use integrity levels (similar to security levels in BLP model)
- The higher the level, the more confidence
  - that a program will execute correctly
  - that data is accurate and/or reliable
- Note relationship between integrity and trustworthiness
- Important point: integrity levels are not security levels

# Read and Write


- BLP upside-down!
  - "no read down" rule (NRD)
  - "no write up" rule (NWU)
- Integrity levels (not disclosure levels)

# Biba Rules

- Set of subjects $S$, objects $O$, integrity levels $I$:
  1. $s \in S$ can read $o \in O$ iff $i(s) \leq i(o)$
  2. $s \in S$ can write to $o \in O$ iff $i(o) \leq i(s)$
  3. $s1 \in S$ can execute $s2 \in S$ iff $i(s2) \leq i(s1)$

# Clark-Wilson Integrity Model

- Introduced by David Clark and David Wilson in 1987
  - D. Clark, D. Wilson, "A Comparison of Commercial and Military Computer Security Policies," IEEE Symposium on Security and Privacy, 1987
- Integrity model specifically targeting commercial applications
- Built on several well-known accounting practices in traditional businesses
- No security levels (unlike BLP and Biba)

# Clark-Wilson Integrity Model

- "Integrity" is defined by a set of constraints
  - Data in a consistent or valid state when it satisfies these constraints
- Example: deposits and withdrawals in a bank
  - $D$ todays deposits, $W$ withdrawals, $YB$ yesterdays balance, $TB$ todays balance
  - Integrity constraint: $D + YB - W = TB$
- "Well-formed transactions" move system from one consistent state to another
- Issue: who examines, certifies transactions done correctly?
  - e.g. invoice paying in a purchasing department
  - separation of duty: transactions implementer and certifier must be different people

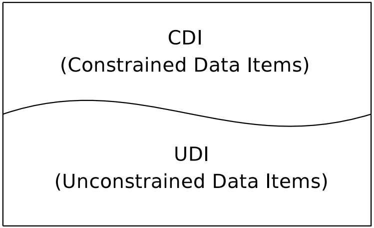
# Model Components

1. CDIs: constrained data items
   - Data subject to integrity controls
2. UDIs: unconstrained data items
   - Data not subject to integrity controls
3. IVPs: integrity verification procedures
   - Procedures that test the CDIs conform to the integrity constraints
4. TPs: transformation procedures
   - Procedures that take the system from one valid state to another
5. Subjects
   - Entities that initiate TPs

**Example (Bank)**

Balances in the accounts are (CDI), Checking the accounts are balanced (IVP), depositing, withdrawing money (TPs)

# Data

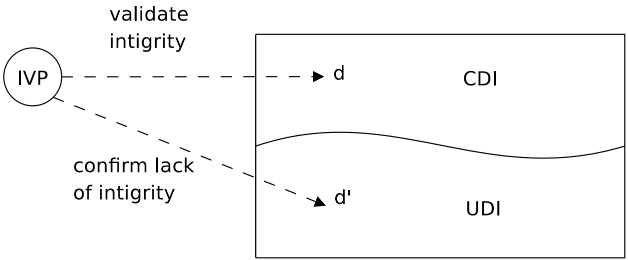



- $D$ is all the data in a computing system (e.g. files in OS)
- Two types of data: CDI and UDI
  - $D = CDI \cup UDI$
  - $CDI \cap UDI = \phi$

# Rules

- The model consists of 9 rules
- The rules are expressed with respect to a given computing system
- The rules are adopted collectively

# Rule 1: Integrity Validation Procedure (IVP)

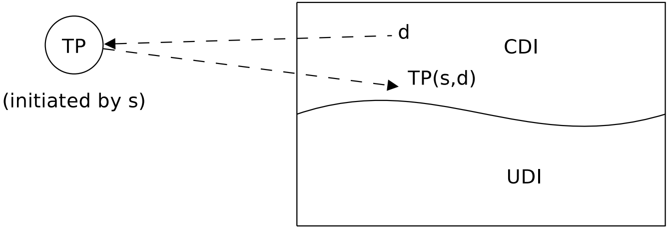



## Rule 1

IVPs must be available on the system for validating the integrity of any CDI

- e.g. checksums

# Rule 2: Intigrity Closure




## Rule 2

Applications of a TP to any CDI must maintain the integrity of that CDI

# Rules 3,4,5

## Rule 3

A CDI can only be changed by a TP

## Rule 4

Subjects can only initiate certain TPs on certain CDIs

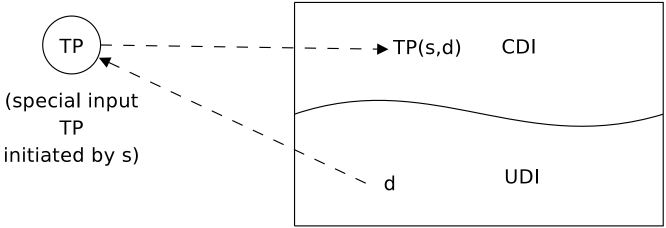
- CW-triple: (subject,tp,d)

## Rule 5

CW-Triples must enforce separation of duty principle

- A subject must not be able to change CDIs without appropriate involvement of other subjects

# Rule 6: Integrity Upgrade




## Rule 6

Certain special TPs on UDI can produce CDI as output

# Rules 7,8,9

## Rule 7

Each TP application cause information sufficient to reconstruct the operation to an append-only CDI

- Auditing/logging

## Rule 8

The system must authenticate each user attempting to initiate a TP

## Rule 9

The system must only permit special subjects (i.e. security officers) to make changes to any authorization-related lists

- Protect against intruders/attackers

# Key Points

- Integrity policies deal with trust
  - As trust is hard to quantify, these policies are hard to evaluate completely
  - Look for assumptions and trusted users to find possible weak points in their implementation
- Biba based on multilevel integrity
- Clark-Wilson focuses on separation of duty and transactions

# Part IV

## Hybrid Policies

# Hybrid Policies

- Most organizations needs a compositions of confidentiality and integrity policies
- Hybrid policies address specific environments
  - Chinese Wall Model: Conflict of Interest

# Chinese Wall Model

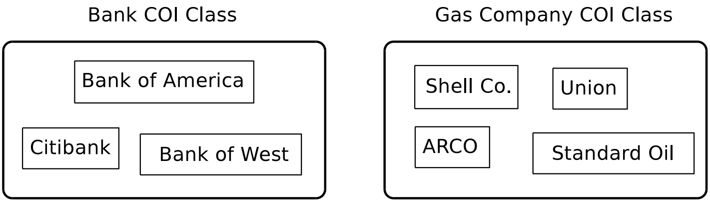
- Problem:
  - Tony advises American Bank about investments
  - He is asked to advise Toyland Bank about investments
- Conflict of interest to accept, because his advice for either bank would affect his advice to the other bank

# Organization

- Organize entities into "conflict of interest" classes
- Control subject accesses to each class
- Control writing to all classes to ensure information is not passed along in violation of rules
- Allow sanitized data to be viewed by everyone

# Definitions

- Objects: items of information related to a company
- Company dataset (CD): contains objects related to a single company
  - Written CD(O)
- Conflict of interest class (COI): contains datasets of companies in competition
  - Written COI(O)
  - Assume: each object belongs to exactly one COI class



# Reading

- Chinese Wall model considers a user's history
- If Anthony reads any CD in a COI, he can never read another CD in that COI

  - Possible that information learned earlier may allow him to make decisions later

## CW-Simple Security Condition

- Let $PR(S)$ be set of objects that s has already read
- $s$ can read $o$ iff either condition holds:
  1. There is an $o'$ such that s has accessed $o'$ and $CD(o') = CD(o)$ (Meaning s has read something in os dataset)
  2. For all $o' \in PR(s)$, $COI(o') \neq COI(o)$ (Meaning s has not read any objects in os conflict of interest class)

# Writing

- Anthony, Susan work in same trading house
- Anthony can read Bank 1s CD, Gas CD
- Susan can read Bank 2s CD, Gas CD
- If Anthony could write to Gas CD, Susan can read it
  - Hence, indirectly, she can read information from Bank 1s CD, a clear conflict of interest

## CW-*-Property

- $s$ can write to $o$ iff both of the following hold:
  1. The CW-simple security condition permits $s$ to read $o$; and
  2. For all objects $o'$, if $s$ can read $o'$, then $CD(o') = CD(o)$

Says that $s$ can write to an object if all the objects it can read are in the same dataset

# Key Points

- Hybrid policies deal with both confidentiality and integrity
- Use different combinations of basic policies