These Slides are prepared from
Matt Bishop slides and book "Introduction to Computer Security"
Benefiting from the Slides posted by Ahmad Al-Mulhem

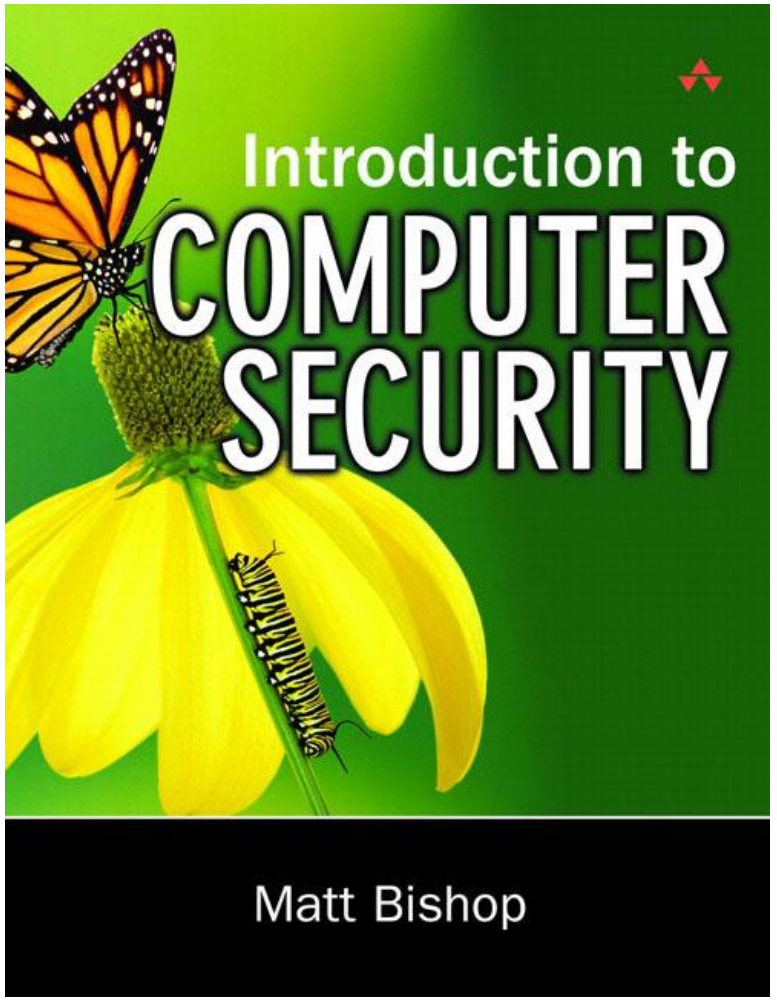
Polices

Security - Ch 4

Confidentiality - Ch 5

Integrity - Ch 6

Hybrid - Ch 7



Adnan Gutub
gutub@kfupm.edu.sa

Computer Engineering Department
King Fahd University of Petroleum & Minerals
Dhahran, Saudi Arabia

Chapter 4: Security Policies

Overview

The nature of policies

- What they cover
- Policy languages

The nature of mechanisms

- Types

Underlying both

- Trust

Security Policy

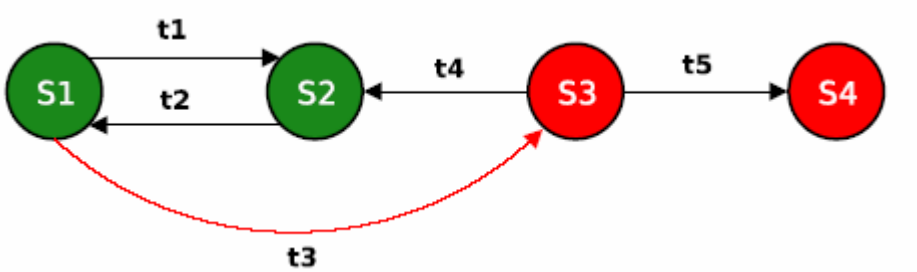
A security policy defines "secure" for a system

Policy partitions system states into:

- Authorized (secure)
  - These are states the system can enter
  - The system should stay in these states
- Unauthorized (nonsecure)
  - If the system enters any of these states, it's a security violation (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 (violation) of security occurs when a system enters an *unauthorized* state.

Confidentiality

X set of entities, *I* information (data)

I has *confidentiality* property with respect to *X* if no $x \in X$ can obtain information from *I*

Even though I can be disclosed (reveled) to other than X

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
  - Even though *I* can be disclosed to the instructor, chairman, and other faculty members

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*, the information, its conveyance (delivery) and storage 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
  - (e.g. a server for a book-store vs medical center)

Security Policies

A security policy considers all relevant aspects of :

- confidentiality
- integrity
- availability

– Who can access information? (confidentiality policy)
  - Dynamic changes

– What are the authorized ways to modify information? (integrity policy)

– What services must be provided and its QoS (quality of service)? (availability policy)

Statement of security policy can be formal (provable) or informal

Implicit (embedded) policies can be confusing (using mechanisms)

Example Question

Policy disallows cheating

- Includes copying homework, with or without permission

COE class has students do homework in computer lab

- Ali forgets to read-protect his homework file

Basem copies it

Who cheated or breached security?

- Ali, Basem, or both?

Answer Part 1

Basem cheated

- Policy forbids copying homework assignment
- Basem did it
- System entered unauthorized state
  - (Basem having a copy of Ali's assignment)

If not explicit in COE 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

Answer Part 2

Ali didn't protect his homework

- Too trusting
- The policy does not disallow this

Ali didn't breach security

If policy said students had to read-protect homework files, then Ali did breach security

- Ali didn't do this

Security Mechanism

Security Mechanism: is an entity or procedure that enforces some part of the security policy

Ali-Basem Homework Example:

- Policy: copying of hw between students are not allowed
- Mechanism: file access control –
  - Ali could protect his file
  - Basem cannot copy it

Security Mechanism Example

A product information is top secret and not allowed to leave the control of the company.

The company stores this information as backup in a Bank Safe Vault.

Mechanism:

- Company need to insure that only authorized employees have access to the backup
- Bank controls access to its safe vault – procedure to access it is part of the security mechanism
  - Note: not all mechanisms should be technical, they can be procedural or operational

Confusion: if security policy is defined through security mechanisms

Security Policy Models

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
- Privacy issues
- May care about integrity and less about availability

Commercial (industrial) security policy (e.g. banks)
- Policy primarily protecting integrity
- May care about availability and less about confidentiality

Confidentiality policy
- Policy protecting only confidentiality – nothing about whether objects should be believed

Integrity policy
- Policy protecting only integrity – how much objects can be trusted

The Role of 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:

1. Trusts patch came from vendor, not tampered with in transit
   - correct
2. Trusts vendor tested patch thoroughly
   - tested at vender
3. Trusts vendor's test environment corresponds to local environment
   - tested for user & conflicts
4. Trusts patch is installed correctly
   - installed as needed

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
   - No 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
   - No compiler bugs, linker/loader/library problems
4. Hardware executes *S′* as intended
   - No hardware bugs

Types of Access Control

- Discretionary Access Control (DAC) / Identity Based Access Control (IBAC)
  - individual user sets access control mechanism to allow or deny access to an object
    - Owner of object controls which subject or identity can access it
- Mandatory Access Control (MAC) / Rule-Based Access Control (RBCA)
  - system mechanism controls access to object, and individual cannot alter that access
    - Neither subject nor owner of object can decide on access permissions
- Originator Controlled Access Control (ORCON or ORG-CON)
  - originator (creator) of object or information is controlling who can access information or object (owner does not)

Example Policy

Computer security policy for academic institution

- Institution has multiple campuses, administered from central office
- Each campus has its own administration, and unique aspects and needs

Authorized Use Policy

Electronic Mail Policy

Authorized Use Policy

Intended for one campus (Davis) only

Goals of campus computing

- Underlying intent

Procedural enforcement mechanisms

- Warnings
- Denial of computer access
- Disciplinary action up to and including expulsion

Written informally, aimed at user community

Electronic Mail Policy

System-wide, not just one campus

Three parts

- Summary
  - genral users
- Full policy
  - precise for all specific users
- Interpretation at the campus
  - descirption on the implementation of the policies

Summary

Warns that electronic mail not private

- Can be read during normal system administration
- Can be forged, altered, and forwarded

Unusual because the policy alerts users to the threats

- Usually, policies say how to prevent problems, but do not define the threats

Summary

What users should and should not do

- Think before you send
- Be polite, respectful of others
- Don't interfere with others' use of email

Personal use okay, provided overhead minimal

Who it applies to

- Problem is the institution is quasi-governmental, so is bound by rules that private companies may not be
- Educational mission also affects application

Full Policy

Context

- Does not apply to Dept. of Energy labs run by the university
- Does not apply to printed copies of email
  - Other policies apply here

E-mail, infrastructure are university property

- Principles of academic freedom, freedom of speech apply
- Access without user's permission requires approval of vice chancellor of campus or vice president of institution
- If infeasible, must get permission retroactively

Uses of E-mail

Anonymity allowed

- Exception: if it violates laws or other policies

Can't interfere with others' use of e-mail

- No spam, letter bombs, e-mailed worms, *etc*.

Personal e-mail allowed within limits

- Cannot interfere with university business
- Such e-mail may be a "university record" subject to disclosure

Security of E-mail

University can read e-mail

- Won't go out of its way to do so
- Allowed for legitimate business purposes
- Allowed to keep e-mail robust, reliable

Archiving and retention allowed

- May be able to recover e-mail from end system (backed up, for example)

Implementation

Adds campus-specific requirements and procedures

- Example: “incidental personal use” not allowed if it benefits a non-university organization
- Allows implementation to take into account differences between campuses, such as self-governance by Academic council

Procedures for inspecting, monitoring, disclosing e-mail contents

Backups

Key Points

Policies describe *what* is allowed

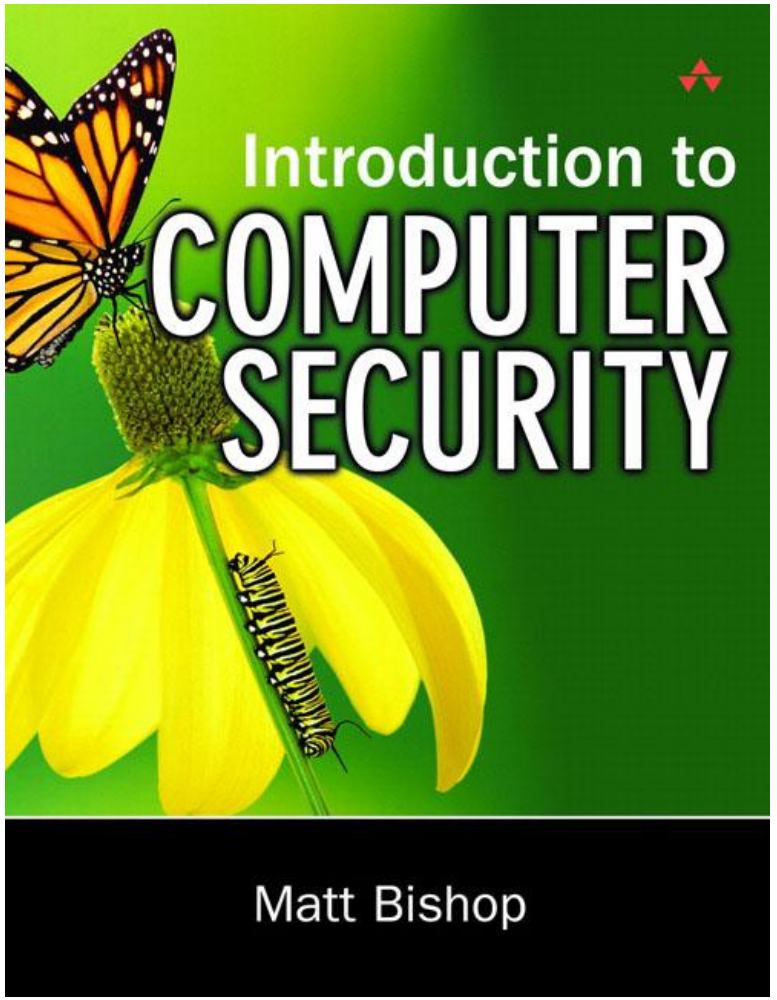
Mechanisms control *how* policies are enforced

Trust underlies everything

These Slides are prepared from
Matt Bishop slides and book "Introduction to Computer Security"
Benefiting from the Slides posted by Ahmad Al-Mulhem

Polices

Security - Ch 4
Confidentiality - Ch 5
Integrity - Ch 6
Hybrid - Ch 7



Adnan Gutub

Computer Engineering Department
King Fahd University of Petroleum & Minerals
Dhahran, Saudi Arabia

Chapter 5: Confidentiality Policies

Overview

- What is a confidentiality model

Bell-LaPadula Model

- General idea
- Informal description of rules

Confidentiality Policy

Also Called: Information flow policy

Goal: prevent the unauthorized disclosure of information

- Deals with information flow
- Integrity and availability are secondary goals
  - e.g. Military information on which & when a ship is out...

Multi-level security models are best-known examples

- Bell-LaPadula Model (military type) basis for many, or most, of these

Bell-LaPadula Model (B-LP)

- Introduced by Elliot Bell and Leonard LaPadula
  - in the 1970s ~ > 30 years old.

Security levels arranged in linear ordering

- TS = Top Secret: highest
- S = Secret
- C = Confidential
- UC = Unclassified: lowest

Security levels correspond to information sensitivity

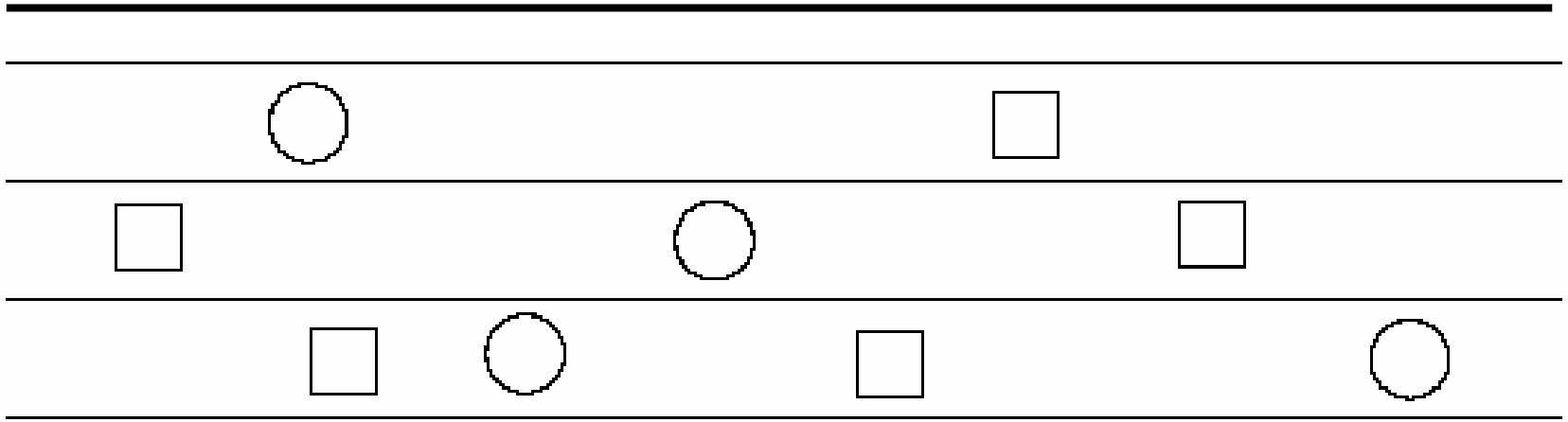
- Subjects have ***security clearance L(s)***
- Objects have ***security classification L(o)***

Example

security level	*subject*	*object*
Top Secret	Basem	Personnel Files
Secret	Ahmad	E-Mail Files
Confidential	Khalid	Activity Logs
Unclassified	Anas	Telephone Lists

- Basem can read all files
- Khalid cannot read Personnel or E-Mail Files
- Anas can only read Telephone Lists

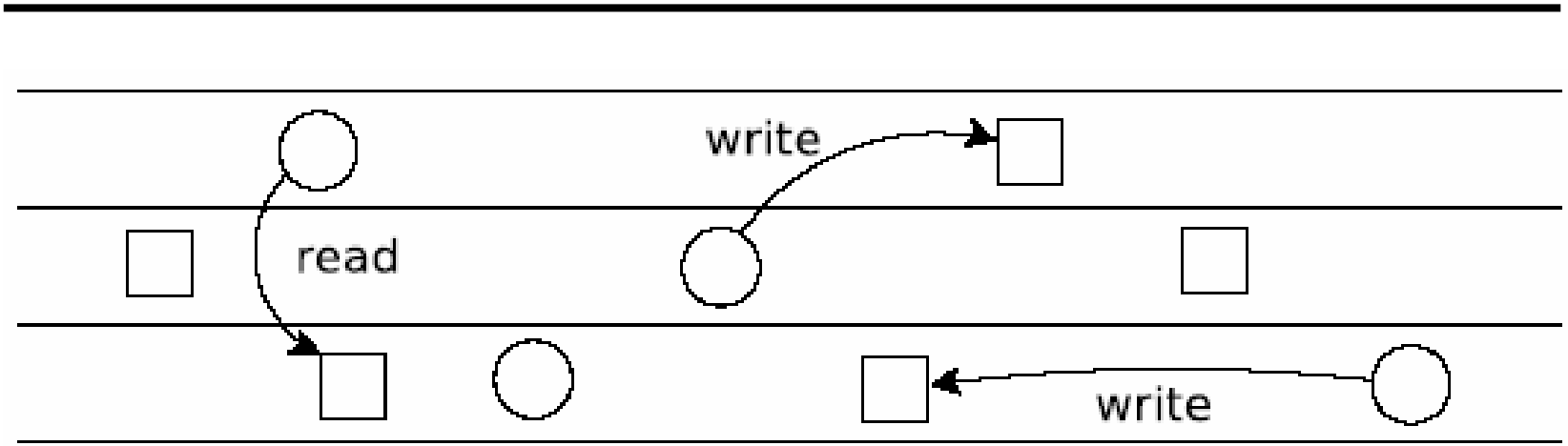
Level Diagrams (Amoroso 94)



Circles are *subjects*

Squares are *objects*

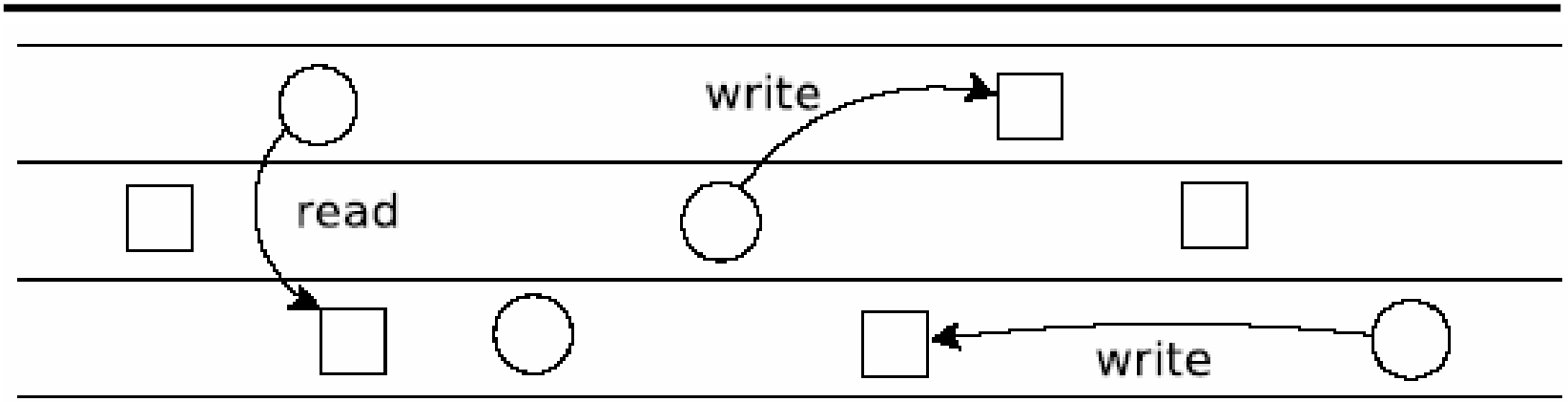
Read and Write



Arrows represent read and write operations

An arrow originates from a subject to an object

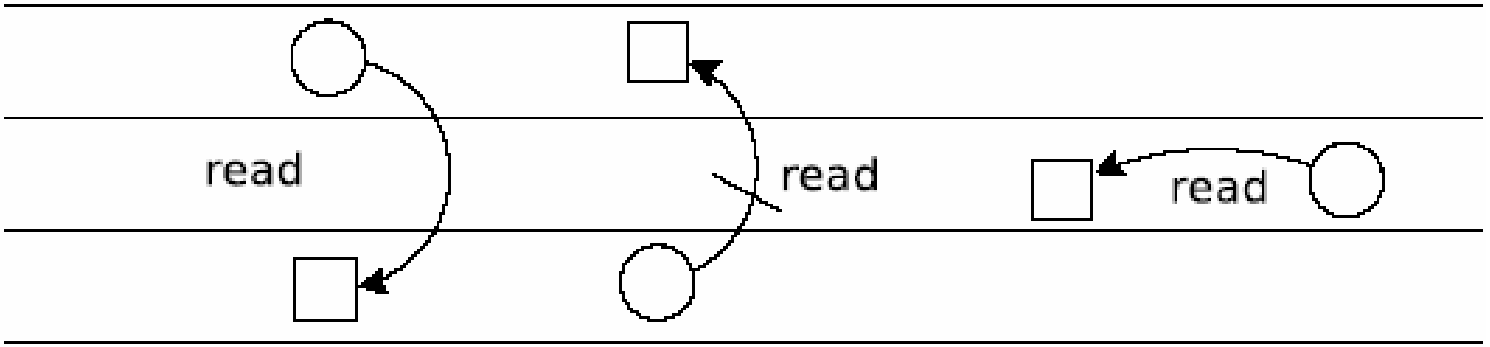
Read and Write (Information flow)



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 Information (B-LP)



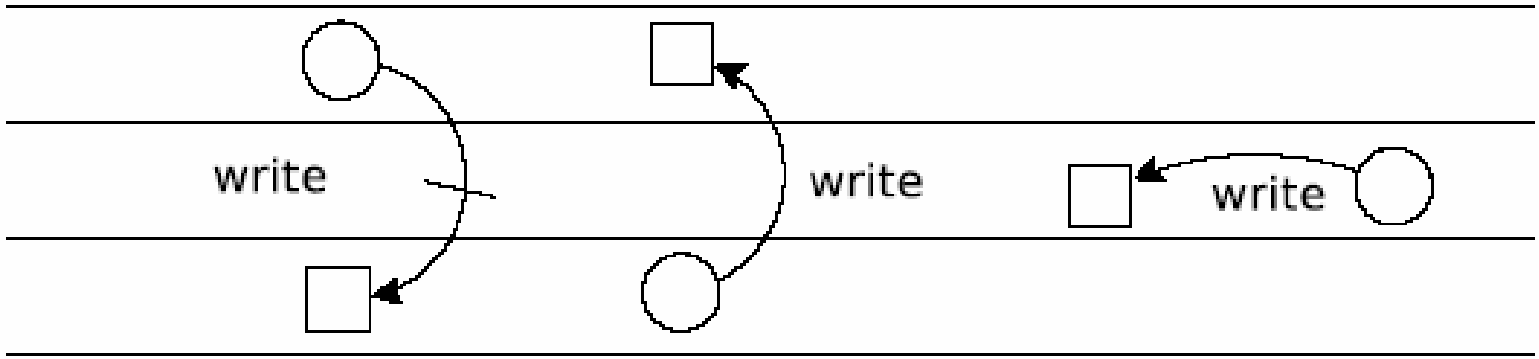
Information flows:

- "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*
  - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
- Sometimes called "no reads up" rule

Writing Information (B-LP)



Information flows up, not down

- "Writes up" allowed, "Writes down" disallowed

*-Property (star property)

- Subject *s* can write object *o* iff $L(s) \leq L(o)$ and *s* has permission to write *o*
  - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
- Sometimes called "no writes down" rule

Basic Security Theorem

Preliminary Version

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

Example

Admin
- Logs, MAC Labels

Users
- Data files

System Programs
- Virus Protection

Admin
- Can read all
- Not allowed to write for users or blow
- It can write to its level

Users
- Can read/write its level
- Can write in Admin level
- Can read system prog level but cannot write to it

System Programs
- Cannot read except its level
- Can write to all

Key Points

Confidentiality models restrict flow of information

Bell-LaPadula models multilevel security

- Influenced the beginning work in computer security
- Rules: NRU & NWD
  - NRU = No Read Up
  - NWD = No Write Down

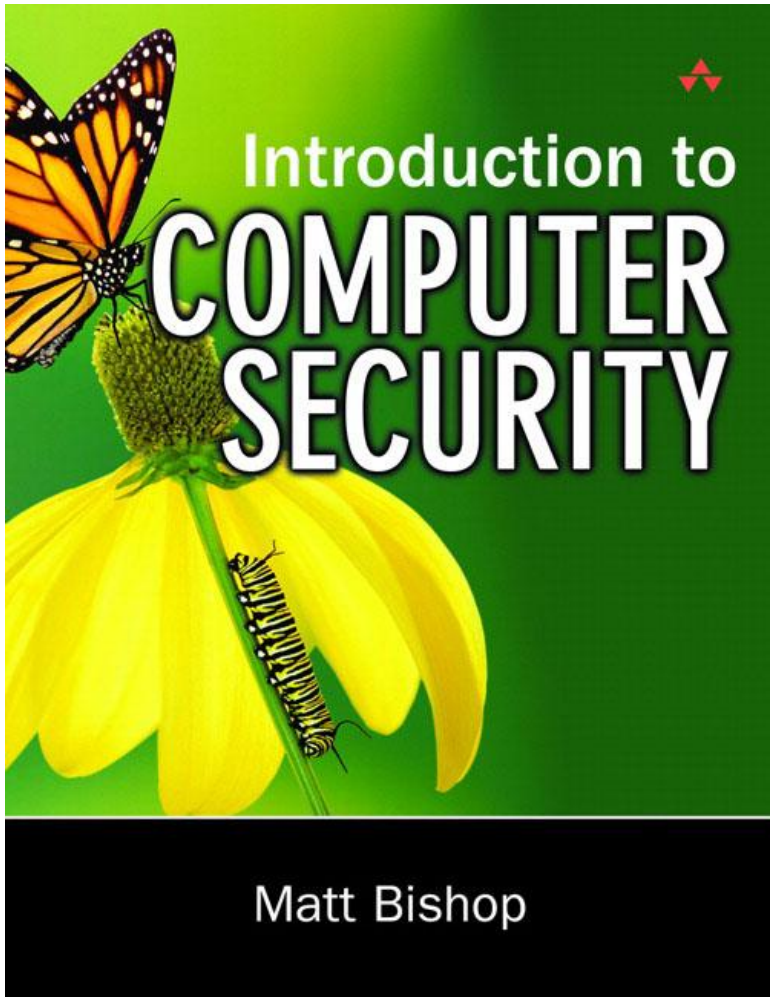
These Slides are prepared from
Matt Bishop slides and book "Introduction to Computer Security"
Benefiting from the Slides posted by Ahmad Al-Mulhem

Polices

Security - Ch 4
Confidentiality - Ch 5
Integrity - Ch 6
Hybrid - Ch 7

Adnan Gutub

Computer Engineering Department
King Fahd University of Petroleum & Minerals
Dhahran, Saudi Arabia



Chapter 6: Integrity Policies

Overview

Requirements & Principles of operation

Biba's models

Clark-Wilson model

Integrity Policies Overview

Very different from confidentiality policies

Concerned more with accuracy of data than their disclosure

- e.g. banks

Mostly used in commercial and industrial environments

5 Requirements (Goals) of Policies

1. Users will not write their own programs, but will use existing production programs and databases.
   - ***Passive Users: toward programming***
2. Programmers will develop and test programs on a non-production system; if they need access to actual data, they will be given production data via a special process, but will use it on their development system.
   - ***Not locally designed: while system is running***
3. **A special process must be followed to install program from the development system onto the production system.**
   - ***Installed correctly***
4. The special process in requirement 3 must be controlled and audited.
   - ***Assessed***
5. The managers and auditors must have access to both the system state and the system logs that are generated.
   - ***Locally controlled***

3 Principles of Operation

Separation of Duty

- Distribute critical function steps among different people
  - Errors can be caught
  - Data can be verified correctly

Separation of Function

- Real system cannot be used by the developers
  - Should be similar to the actual environment but not while running

Auditing

- Analyze to determine what actions and who performed them
- Allow widespread assessment
  - Need extensive logging
- Emphasize on recovery and accountability

Biba Integrity Model

- Introduced by Ken Biba in 1977~ >30 years

Use integrity levels (similar to security levels in BLP model)

The higher the level, the more confidence & trusted

- that a program will execute correctly
- that data is accurate and/or reliable

Note - relationship between integrity and trustworthiness

- Subject may be higher level than an object = Subject is considered more trustworthy than that object

Important point: *integrity levels* are *not* *security* levels

- Integrity labels are assigned and maintained separately due to difference in reasoning
  - Security labels – information flow
  - Integrity labels – information modification

Read and Write

B-LP upside-down!

- "no read down" rule (NRD)
- "no write up" rule (NWU)

Integrity levels

- not disclosure levels

Biba's Model Rules

Similar (Dual) to Bell-LaPadula model (B-LP)

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. $s_1 \in S$ can execute $s_2 \in S$ iff $i(s_2) \leq i(s_1)$

Strict integrity model – not having partitions nor discretionary controls

What condition between $i(s)$ & $i(o)$ can make the read & write operate simultaneously on s and o?

Biba's Model Example

Pozzo & Gray

Goal:

- limit execution domains of each program to prevent un-trusted data alteration

Approach:

- Explicit level of trust in software & data
- Explicit level of risk for each users
  - indicates lowest credibility level the user is allowed to execute

Mechanism:

- Programs distributed among classes
- Assign a trustworthy measure to every class
  - 0 = untrusted
- Users are not allowed to execute lower levels than their risk level
  - If needed = run untrusted command = high risk "acknowledgment"

Clark-Wilson Integrity Model

- Introduced by David Clark and David Wilson in 1987 ~ > 20 Years

Integrity model specifically targeting commercial applications

Built on several well-known accounting practices in traditional businesses

- No security levels - unlike B-LP & Biba
- Targets
  - Data –Integrity of data
  - Procedure - Permission of actions allowed on that data
- More realistic than B-LP & Biba

Clark-Wilson Integrity Model

Integrity is defined by a set of constraints: Data & Transaction

1-Data in consistent or valid state when it satisfies these constraints

- Example: deposits and withdrawals in a ***bank***
  - *D* today's deposits, *W* withdrawals, *YB* yesterday's balance, *TB* today's balance
  - Integrity constraint: $D + YB - W = TB$

2-Well-formed transaction only move system within *consistent* states

- *State Consistency* hold (verified) Before & After Transaction
- *Correct Transactions:* who examines, certifies transactions done correctly?
  - e.g. invoice paying in a purchasing department (5 steps)
    - » 1-request received; 2-determin account for payment; 3-validate invoice according to service needed; 4-account should assume debited; 5-check written & signed
- separation of duty: transactions implementer and certifier must be different people

Entities - Model Components

CDIs: constrained data items
- Data subject to integrity controls

UDIs: unconstrained data items
- Data not subject to integrity controls

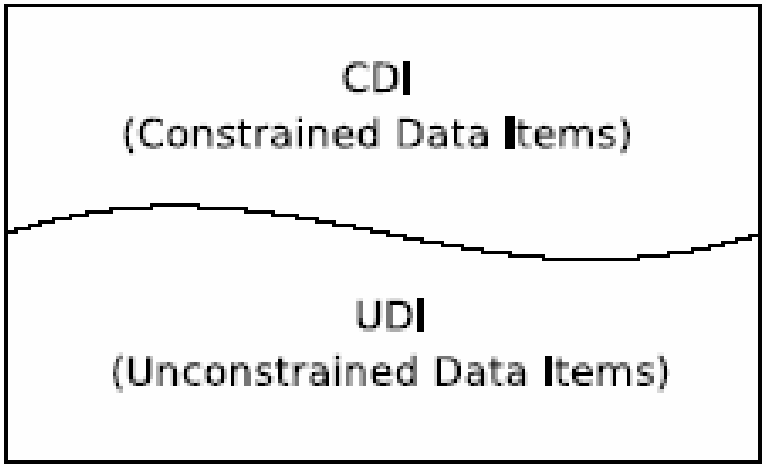
IVPs: integrity verification procedures
- Procedures that test the CDIs conform to the integrity constraints
  - Valid state

TPs: transaction procedures
- Procedures that take the system from one valid state to another
  - Implement ***well-formed transactions***

Subjects: Entities (e.g. bank officers, authorized users) that initiate TPs

Example (Bank)
- Balances in the accounts are (CDI), Checking the accounts are balanced (IVP), depositing, withdrawing money (TPs), (UDI)????

Data



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

Clark-Wilson Model Rules

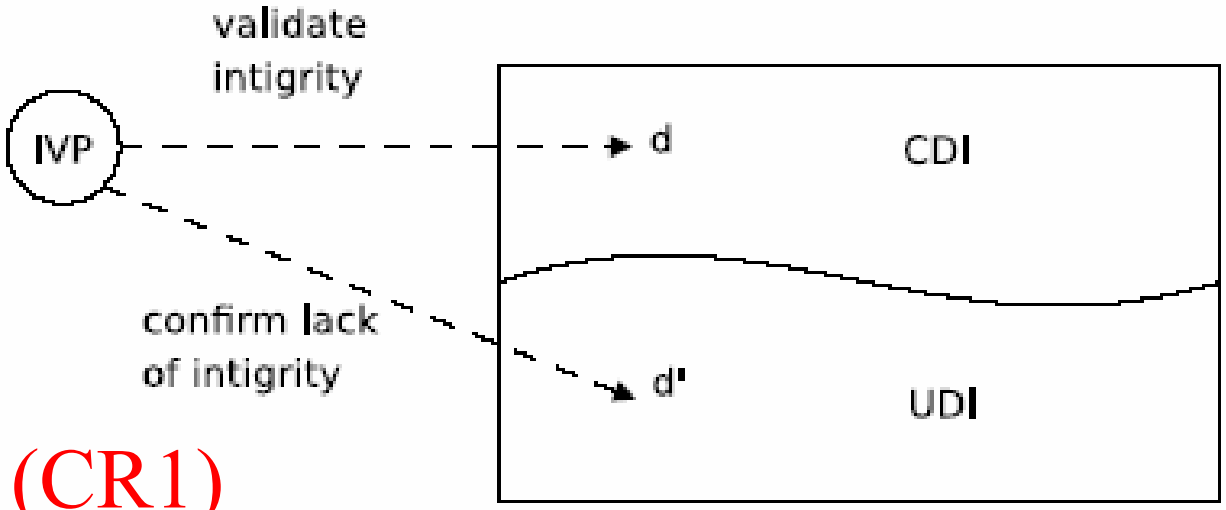
The model consists of 9 rules

- 5 Certification rules (CRs)
- 4 Enforcement rules (ERs)

The rules are expressed with respect to a given computing system

The rules are adopted *collectively*

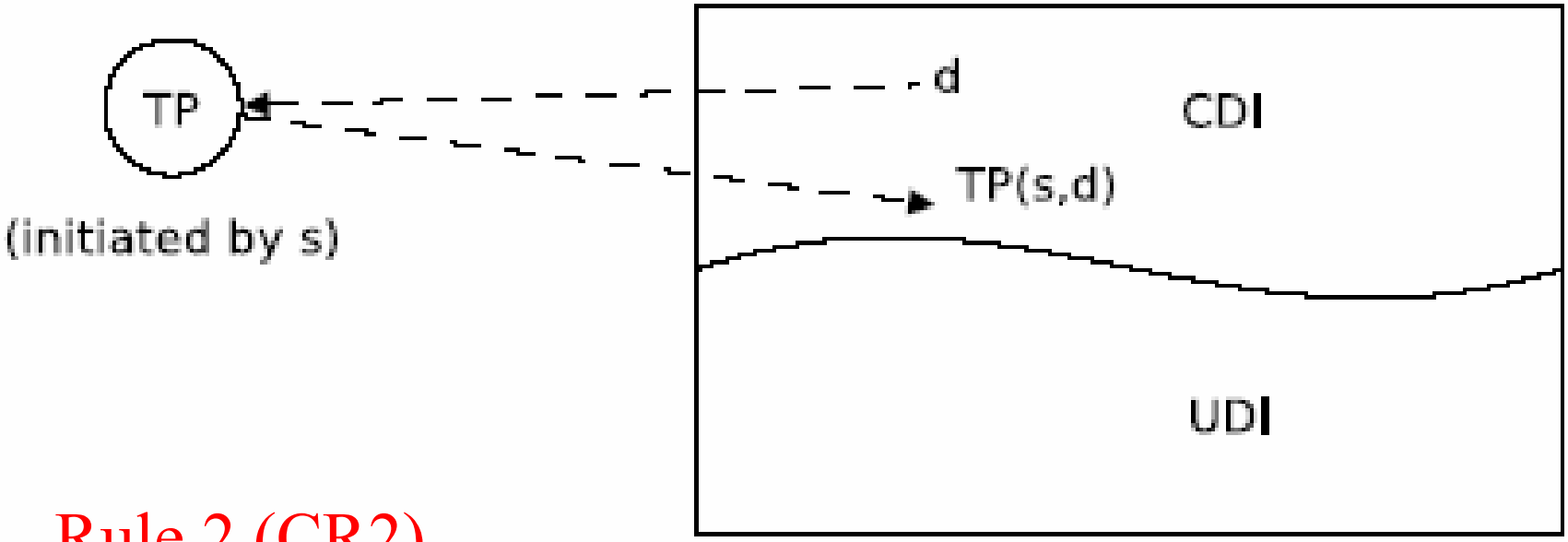
Rule 1 (CR1):
Integrity Validation Procedure (IVP)



Rule 1 (CR1)

- IVPs must be available on the system for validating the integrity of any CDI
  - IVP must insure that all CDIs are in valid state
    - e.g. checksums

Rule 2 (CR2):
Integrity Closure by TP



Rule 2 (CR2)

- Applications of a *TP* to any CDI must maintain the integrity of that CDI
  - TP may corrupt CDI
  - All CDIs are transformed between valid states

Rules 3,4,5

Rule 3 (ER1) : Certified relation

- A CDI can only be changed by a certified TP allowed to run on this CDI
  - E.g. janitor is not allowed to balance customers accounts
  - Need to enforce on persons or users performing TP => ER2

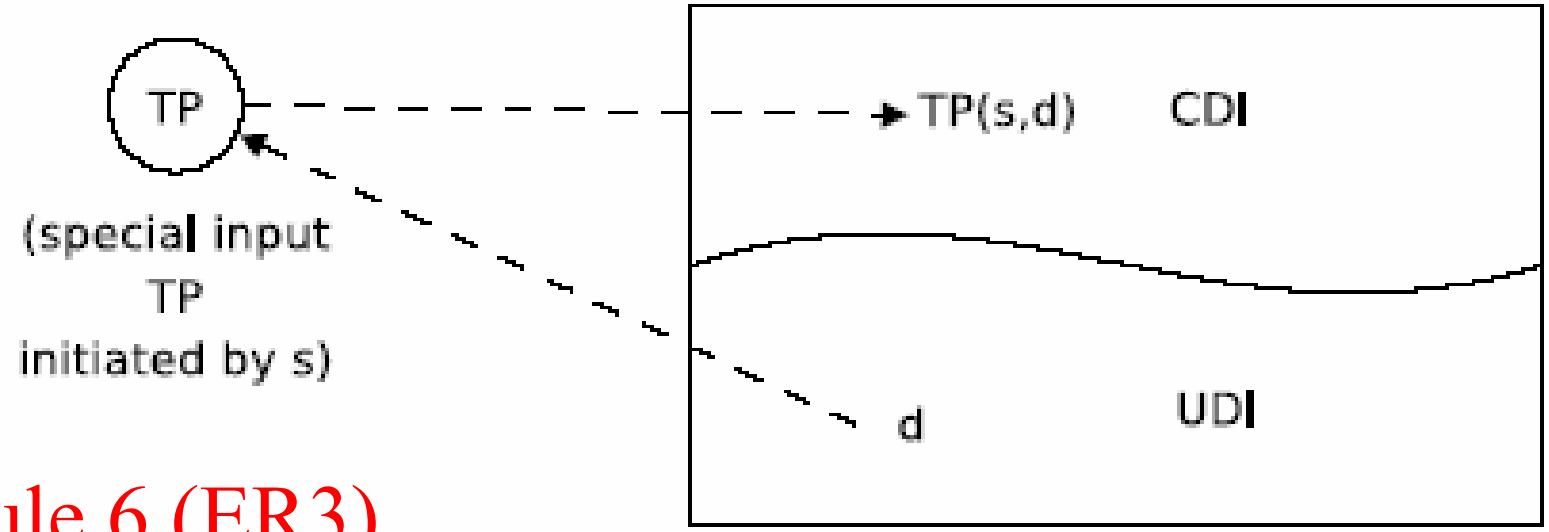
Rule 4 (ER2): Associate Users & CDIs

- Subjects can only initiate certain TPs on certain CDIs
  - Defined by set of CW-Triples: (subject or user, TP, CDI set)
  - These triples will set relations allowed –should be certified by CR3

Rule 5 (CR3) : Triples & Separation of Duty

- 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 (ER3)
Integrity Upgrade



Rule 6 (ER3)

- Certain special TPs on UDI can produce CDI as output
  - UDI may not need authentication after log in
  - If user want to manipulate CDI, TP is needed as ER2

Rules 7,8,9

Rule 7 (CR4) : Auditing/logging

- Each TP application must require & attach sufficient information to reconstruct the operation to an append-only CDI
  - Entering money (deposit) into ATM – if mentioned amount not correct
    - Not a problem
    - This is an example of UDI
  - But when the ATM is opened, correct detection and fixing any errors needs to be done before depositing the amount into one's account
    - Detection is example of transforming into CDI

Rule 8 (CR5) : TP Initiator Authentication

- The system must authenticate each user attempting to initiate a TP
  - If UDI is input to TP, either TP rejects UDI or valid transformation to CDI

Rule 9 (ER4) : Authorization Modification & Separation of duty

- The system must only permit special subjects (i.e. security officers) to make changes to any authorization-related lists
  - No TP executor is allowed to change authorization
  - Protect against intruders/attackers

Relating: Clark-Wilson 9 Rules with the 5 Requirements of Integrity Policies

Passive Users: toward programming & data production

- CR5 & ER4
- Users cannot write progs, they must use existing TPs and CDIs

Not locally designed: while system is running

- Simulate TP correctly

Installed correctly

- Using installing TP correctly – trusted personnel

Assessed

- CR4, ER3, CR5, ER4

Locally controlled

- Logs are CDIs
- Managers can control system through appropriate TPs

Comparison to Biba

Biba

- Attach many integrity levels to subjects & objects
- No notion of certification rules; trusted subjects ensure actions obey rules
- Untrusted data examined before being made trusted

Clark-Wilson

- 2 levels:
  - Objects: CDI or UDI
  - Subjects: TP & all others
- Explicit requirements that *actions* must meet
- Trusted entity must certify *method* to upgrade untrusted data (and not certify the data itself)

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

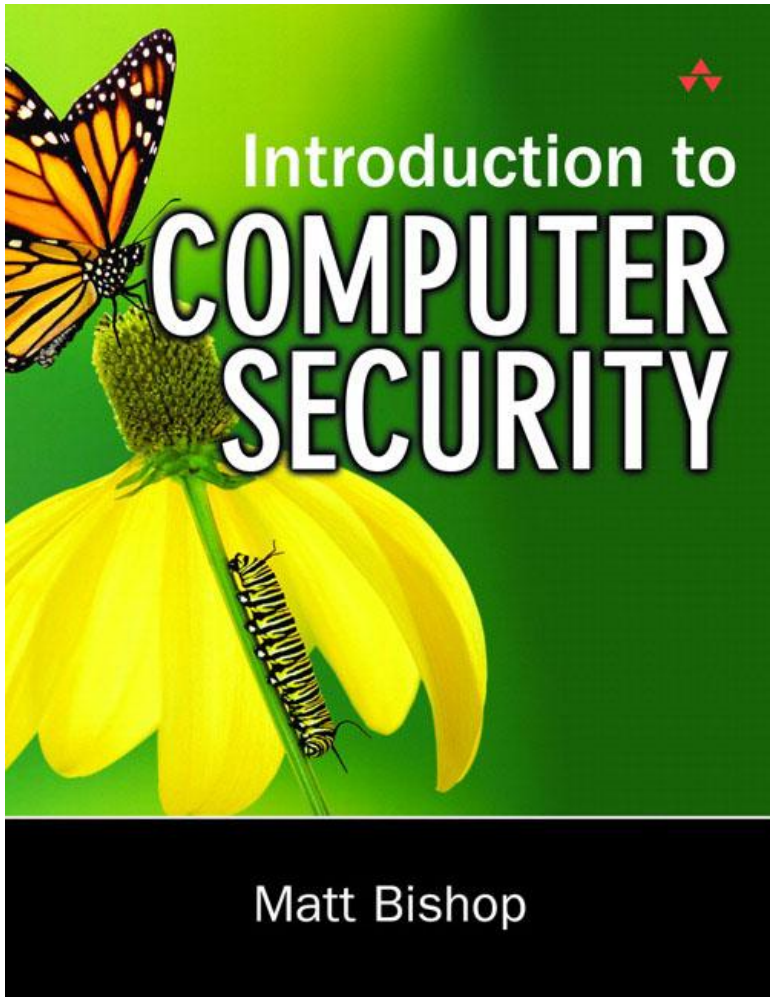
Biba based on multilevel integrity

Clark-Wilson focuses on separation of duty and transactions

These Slides are prepared from
Matt Bishop slides and book "Introduction to Computer Security"
Benefiting from the Slides posted by Ahmad Al-Mulhem

Polices

Security - Ch 4
Confidentiality - Ch 5
Integrity - Ch 6
Hybrid - Ch 7



Adnan Gutub

Computer Engineering Department
King Fahd University of Petroleum & Minerals
Dhahran, Saudi Arabia

Chapter 7: Hybrid Policies

Overview

Chinese Wall (CW) Model

Clinical Information Systems Security (CISS) Policy

Originator Controlled Access Control (ORCON)

Role-Based Access Control (RBAC)

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

Overview

Chinese Wall (CW) Model

- Focuses on conflict of interest

Self reading

CISS Policy

- Combines integrity and confidentiality

ORCON

- Combines mandatory, discretionary access controls

RBAC

- Base controls on job function

Chinese Wall (CW) Model

Problem:

- Anas advises Bank A about investments
- He is asked to advise Bank B 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 (COI) 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* (clean) data to be viewed by everyone – to become *unsanitized*

Definitions

Objects (O): items of information related to a company

Company dataset (CD): contains objects related to a single company

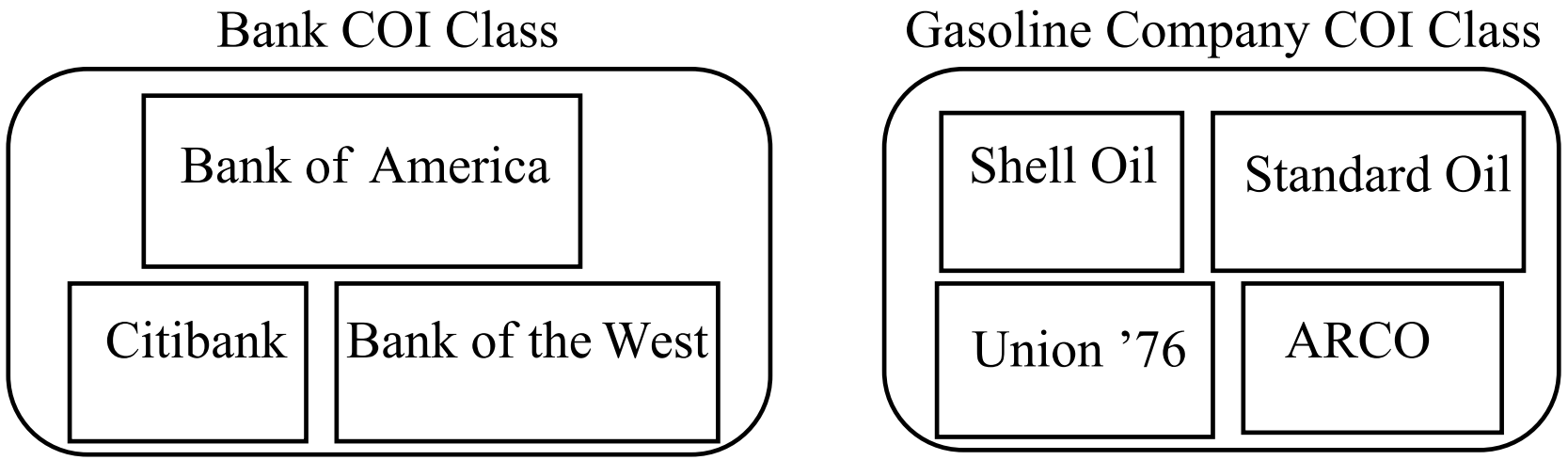
- Written *CD*(*O*): company dataset containing object O

Conflict of interest class (COI): contains datasets of companies in competition

- Written *COI*(*O*): COI class containing object O

- Assume: each object belongs to exactly one *COI* class

Example: Chinese Wall (CW) model database



It has two COI classes: for banks contains 3 CDs, for gasoline companies, contains four CDs.

Any employee may have access to no more than one CD in each COI

He could access Citibank's CD and ARCO's CD, but *not* Citibank's CD and Bank of America's CD.

Reading: Temporal Element

Chinese Wall model considers a *user's history*

If Anas 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:
  - There is an o′ such that s has accessed o′ and CD(o′) = CD(o)
    - (Meaning s has read something in os dataset)
  - For all o′ ∈ PR(s), COI (o′) ≠ COI (o)
    - (Meaning s has not read any objects in os conflict of interest class)

Writing

Ahmad & Sami both work in same trading house

Ahmad can read Bank 1's CD, Gas' CD

Sami can read Bank 2's CD, and same Gas' CD

If Ahmad could write to Gas' CD, Sami can read it

- Hence, indirectly, Sami can read information from Bank 1's CD, leading to same 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 un-sanitized 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

Compare to Bell-LaPadula

Fundamentally different

- CW has no security labels, B-LP does
- CW has notion of past accesses, B-LP does not

Bell-LaPadula can capture state at any time

- Each (COI, CD) pair gets security category
- Two clearances, *S* (sanitized) and *U* (unsanitized)
- Subjects assigned clearance for compartments without multiple categories corresponding to CDs in same COI class

Compare to Bell-LaPadula

Bell-LaPadula cannot track changes over time

- Sami becomes ill, Ali needs to take over
  - CW history lets Ali know if he can
  - No way for Bell-LaPadula to capture this

Access constraints change over time

- Initially, subjects in CW can read any object
- Bell-LaPadula constrains set of objects that a subject can access
  - Can't clear all subjects for all categories, because this violates CW-simple security condition

Compare to Clark-Wilson

Clark-Wilson Model covers integrity aspects, such as validation & verification as well as access control

CW only deal with access control, that is way it cannot fully emulate Clark-Wilson model

- Lets consider only access control of Clark-Wilson

If "subjects" and "processes" are interchangeable, a single person could use multiple processes to violate CW-simple security condition

- Would still comply with Clark-Wilson Model

If "subject" is a specific person and includes all processes the subject executes, then consistent with Clark-Wilson Model

Key Points

Hybrid policies deal with both confidentiality & integrity

Use different combinations of basic policies