

Trusted Operating Systems

*CSCI283 Fall 2003 Lecture 8
GWU*

*Draws extensively from Memon's notes, Brooklyn Poly
And text, Chapter 5*

**YOU ARE EXPECTED TO READ CHAPTER 5 FROM
THE TEXT IN ADDITION TO THIS**

Announcements

Project proposal deadline extended to 3rd Nov., Monday

Graduate Events next week (undergrads very welcome too)

<http://www.cs.gwu.edu/~studentcorner/graduateEvents/>

- Monday, 3rd Nov., 12:30, Peter Bock, Faculty Talk: What You See is not What You get
- Wed., 5th Nov., 4-6 GW Women in CS
- Fri., 7th Nov., 2-3:30, Graduate Seminar

A summer internship possible in intrusion modelling for a defense project, contact me

Next semester crypto class: see website linked off my website.

Security Services in an OS

- A general purpose OS provides the following security mechanisms:
 - Memory protection
 - File protection
 - General object protection
 - Access authentication
- How do we go about designing a “trusted” OS (one that we believe provides the above)?
- We prefer the term “trust” as opposed to “secure”.

Need

- Policy: description of requirements
- Model: policy representation
- Design: implementation of policy
- Trust: based on features and assurance

Trust

- Policies, mechanisms and procedures make assumptions and one trusts these assumptions hold.
- SA receives security patch and installs it. Has she increased the security of the system?
- Aspirin from drugstore is considered trustworthy. The basis of this trust is:
 - Testing and certification by FDA.
 - Manufacturing standard of company and regulatory mechanisms that ensure it.
 - Safety seal on the bottle.
- Similarly, for a secure system to achieve trust, specific steps need to be taken.

Qualities of Security and Trustedness

Secure	Trusted
<ul style="list-style-type: none">• <i>Either-or</i>: Something is secure or not secure.• Property of <i>presenter</i>• <i>Asserted</i>: based on product characteristics• <i>Absolute</i>: not quantified as to how, when, where or by whom.• <i>A goal</i>	<ul style="list-style-type: none">• <i>Graded</i>: There are degrees of trustedness.• Property of <i>receiver</i>• <i>Judged</i>: based on evidence and analysis• <i>Relative</i>: viewed in context of use • <i>A characteristic</i>.

Security Policy

- A security policy is a set of rules stating which actions are permitted and which are not.
- Can be informal or highly mathematical.
- If we consider a computer system to be a finite state automaton with state transitions then
 - A *security policy* is a statement that partitions the states of a system into a set of authorized or secure states and a set of unauthorized or non-secure states.
 - A *secure system* is a system that starts in an authorized state and cannot enter an unauthorized state.
 - A *breach of security* occurs when a system enters an unauthorized state.
- We expect a trusted system to enforce the required security policies.

Confidentiality, Integrity and Availability

- *Confidentiality*: Let X be a set of entities and I be some information. Then I has the property of confidentiality with respect to X if no member of X can obtain information about I .
- *Integrity*: Let X be a set of entities and I some information or a resource. Then I has the property of integrity with respect to X if all members of X trust I .
- *Availability*: Let X be a set of entities and I a resource. Then I has the property of availability with respect to X if all members of X can access I .

Example

- Let X be the set of students in CS283.
- Let I be:
 - the set of student grades in the test
 - the set of class notes on the website
 - the US's military policy in Iraq
- Let X be a single student in CS283. Let I be this student's grade in the last exam.
- For all the above, does I enjoy confidentiality, integrity and accessibility wrt X ?

What technical mechanisms are used?

To ensure

- Confidentiality
- Integrity
- Accessibility

- What needed in addition to the mechanisms?

Elements of a Security Policy

- A security policy considers all relevant aspects of confidentiality, integrity and availability.
 - Confidentiality policy: Identifies information leakage and controls information flow.
 - Integrity Policy: Identifies authorized ways in which information may be altered. Enforces separation of duties of individuals to ensure robustness against single corrupt individuals.
 - Availability policy: Describes what services must be provided: example – a browser may download pages but no Java applets.

Mechanism and Policy

- **Example:** University policy disallows cheating – copying another student's homework assignment. Student A has her homework file world-readable. Student B copies it. Who has violated policy?
- Mechanism should not be confused with policy.
- A **security mechanism** is an entity or procedure that enforces some part of a security policy.

Types of Security Policies

- A *military security policy* (also called government security policy) is a security policy developed primarily to provide confidentiality.
 - Not worrying about trusting the object as much as disclosing the object
- A *commercial security policy* is a security policy developed primarily to provide a combination of confidentiality and integrity.
 - Focus on how much the object can be trusted.
- Also *confidentiality policy* and *integrity policy*.

Military Security Policy

- *Heirarchy of sensitivities*, e.g.: top secret, secret, confidential, restricted, unclassified
- *Compartments*, e.g: Iraq, WMDs, Crypto, non-proliferation, RSA, India, Israel
- Pieces of Information held by US military, e.g.: Saddam's location, India's and Israels' nuclear capability, security of RSA, published information on RSA, the names of Israel's cabinet ministers
- Classify above pieces of information into their *classes*: <rank; compartments>

Domination

- Subject s cleared for class $\langle \text{rank}_s; \text{compartments}_s \rangle$ and Object o is in class $\langle \text{rank}_o; \text{compartments}_o \rangle$
- $s \leq o$ iff
 - $\text{rank}_s \leq \text{rank}_o$ and
 - $\text{compartments}_s \subseteq \text{compartments}_o$
- o dominates s
- Military Security Policy: Subject has access to Object iff Subject dominates Object

Examples

- Consider the subjects:
 - President of the USA
 - Prime Minister of Israel
 - Saddam Hussein
 - Bin Laden

- Assess their
 - Domination
 - Access

wrt the previous examples

Example from text

User cleared for: <secret: {dog, cat, pig}> has access to?

<top secret; {dog}>

<secret; {dog}>

<secret; {dog, cow}>

<secret; {moose}>

<confidential; {dog, pig, cat}>

<confidential; {moose}>

Security Models

- To formulate a security policy you have to describe entities it governs and what rules constitute it – a *security model* does just that!
- A *security model* is a model that represents a particular policy or set of policies. Used to:
 - Describe or document a policy
 - Test a policy for completeness and consistency
 - Help conceptualize and design an implementation
 - Check whether an implementation meets requirements.

Partial Ordering

- A partial ordering is a relation \leq that is reflexive, transitive and anti-symmetric
 - Reflexive – $a \leq a$
 - Transitive – If $a \leq b$ and $b \leq c$, then $a \leq c$.
 - Anti-symmetric – If $a \leq b$ and $b \leq a$, then $a = b$.
- Example:
 - Child of?
 - Sibling of?
 - Identical genetic content?
 - Subset of \subseteq ?
 - Divisor of ?
 - Less than equal to \leq ?

Lattices

- A lattice is a collection of tuples (x, y) from a set X where
 - both x and y belong to set A and $x \leq y$.
 - Every tuple in the lattice has a greatest upper bound.
 - Every tuple has a least lower bound.
- Note that not all tuples that can be formed from elements in X belong to the lattice. That is some elements are not comparable (partial ordering).

Examples

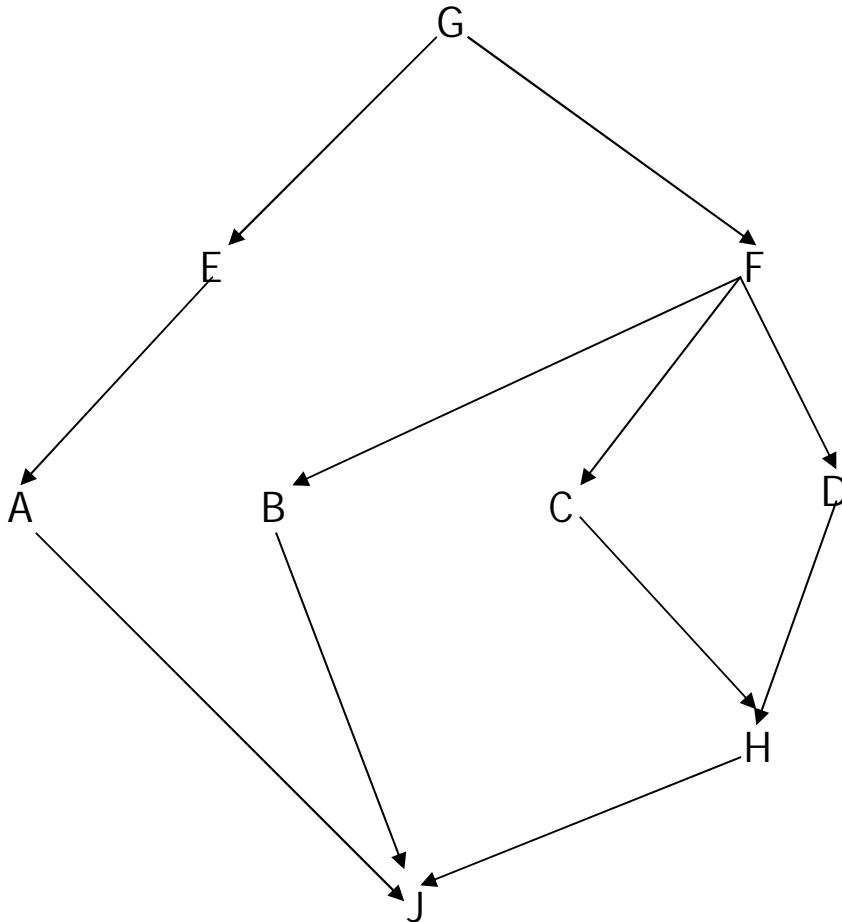
Using the following partial orders, define lattices

Subset of \subseteq ?

Divisor of ?

Less than equal to \leq ?

Lattice - Example



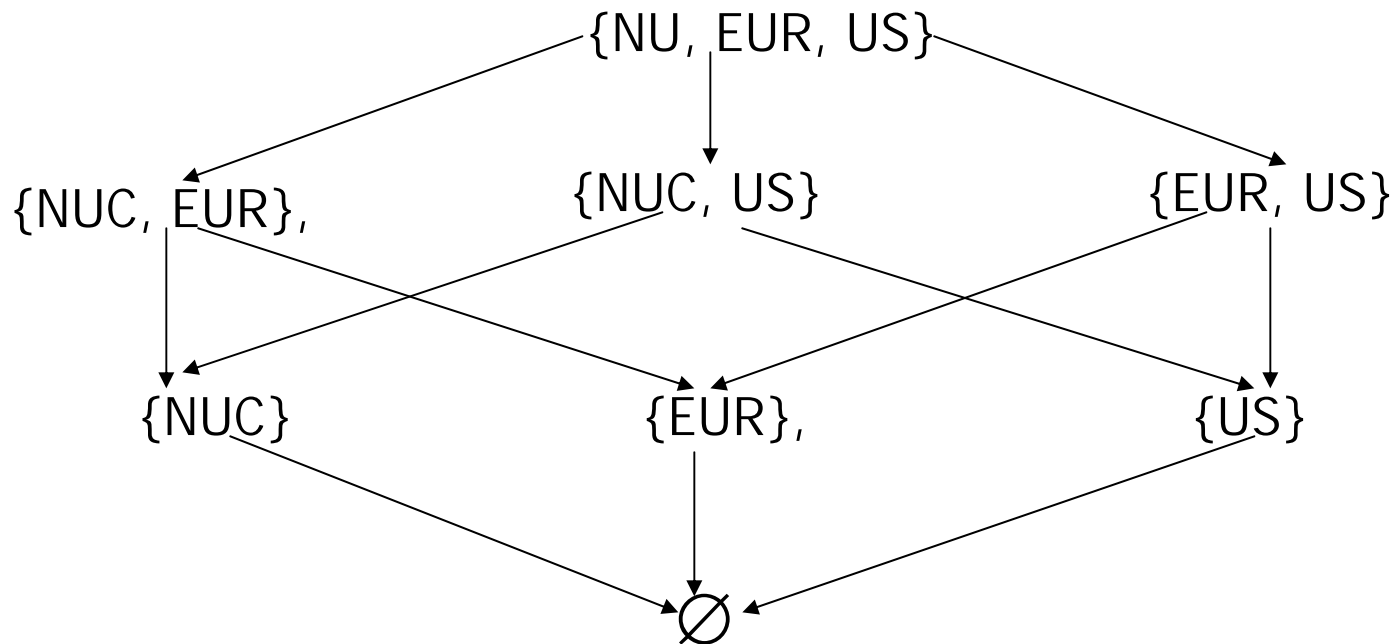
Is $B \leq G$?

Is $B \leq E$?

BLP

- Divide each security level into a set of categories.
- Each security level and category forms a *compartment*. We say subjects have clearance for a set of compartments and objects being at the level of a compartment.
 - Need to know principle.
- Example: Let NUC, EUR and US be categories.
 - Sets of categories are Null, {NUC}, {EUR}, {US}, {NUC, US}, {NUC, EUR}, {EUR, US} and {NU, EUR, US}.
 - George is cleared for (TOP SECRET, {NUC, US})
 - A document may be classified as (CONFIDENTIAL, {EUR}).

Example Lattice



- The set of categories form a lattice under the subset operation

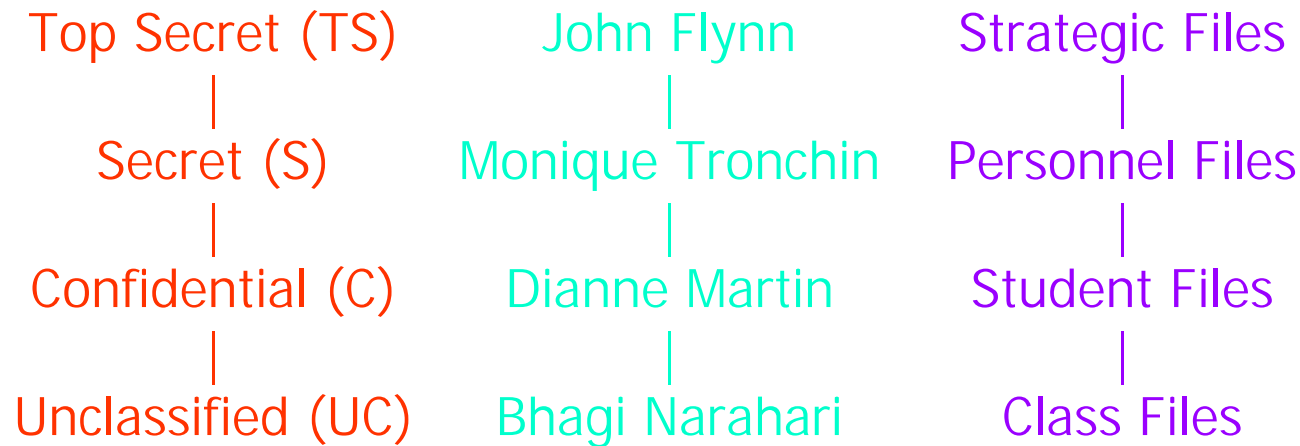
The Bell-La Padula (BLP) Model

- BLP model is a formal description of allowable paths of information flow in a secure system.
- Formalization of military security policy – confidentiality.
- Set of subjects S and objects O . Each subject s in S and o in O has a fixed security class
- Security classes are ordered by a relation \leq
- Combines mandatory and discretionary access control.

Discretionary and Mandatory Access Control

- A security policy may use two types of access controls:
 - An individual user can set an access control mechanism to allow or deny access to an object, that mechanism is a *discretionary access control (DAC)* or *identity-based access control (IBAC)*.
 - When a system mechanism controls access to an object and an individual user cannot alter that access, the control is called *mandatory* or *rule-based access control (MAC)*.

Example



A basic confidentiality classification system.

Security Levels, Subjects, Objects

What is the Information Leakage?

BLP – Simple Version

The secure flow of information is characterized by:

- *Simple Security Property*: A subject s may have read access to an object o if and only if $o \leq s$
 - **-Property*: A subject s who has read access to an object o may have write access to an object p only if $o \leq p$
- (Contents of a sensitive object can only be written to objects at least as high. That is, prevent write-down).

BLP – Simple Version (Contd.)

Basic Security Theorem: Let Σ be a system with a secure initial state σ_0 and let T be a set of transformations. If every element of T preserves the simple security property and $*$ -property, then every state σ_i $i \geq 0$ is secure.

Commercial Environments

Commercial requirements differ from military requirements in their emphasis on preserving data integrity. For Example:

1. Users will not write their own programs, but will use existing production programs and databases.
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.
3. A special process must be followed to install a program from the development system onto the production system.
4. The special process in 3, above, must be controlled and audited.
5. The management and auditors must have access to both the system state and to the system logs that are generated.

Principles of Operation

- *Separation of duty*. If two or more steps are required to perform a critical function, at least two different people should perform the steps.
- *Separation of function*. Developers do not develop new programs on production systems because of the potential threat to production data.
- *Auditing*. Auditing is the process of analyzing systems to determine what actions took place and who performed them. Commercial systems emphasize recovery and account-ability.

Biba Integrity Model

- Biba integrity model is counterpart (dual) of BLP model.
- It identifies paths that could lead to inappropriate modification of data as opposed to inappropriate disclosure in the BLP model.
- A system consists of a set S of subjects, a set O of objects, and a set I of integrity levels. The levels are ordered.
- Subjects and Objects are ordered by the integrity classification scheme; denoted by $I(s)$ and $I(o)$.

Biba Integrity Model

- The properties of the Biba Integrity Model are:
 - *Simple Integrity Property*: Subject s can modify (have write access to) object o if and only if $I(s) \geq I(o)$.
 - *Integrity *-property*: If subject S has read access to object o with integrity level $I(o)$, S can have write access to p if and only if $I(o) \geq I(p)$.

Why does this make sense?

Clark-Wilson Integrity Model

- In commercial environment we worry about the integrity of the data in the system and the actions performed upon that data.
- The data is said to be *in a consistent state* (or *consistent*) if it satisfies given properties.
 - For example, let D be the amount of money deposited so far today, W the amount of money withdrawn so far today, YB be the amount of money in all accounts at the end of yesterday, and TB be the amount of money in all accounts so far today. Then the consistency property is:

$$D + YB - W = TB$$

CW Model

- A *well-formed transaction* is a series of operations that leave the data in a consistent state if the data is in a consistent state when the transaction begins.
- The principle of separation of duty requires the certifier and the implementers be different people.
 - In order for the transaction to corrupt the data (either by illicitly changing data or by leaving the data in an inconsistent state), either two different people must make similar mistakes or collude to certify the well-formed transaction as correct.

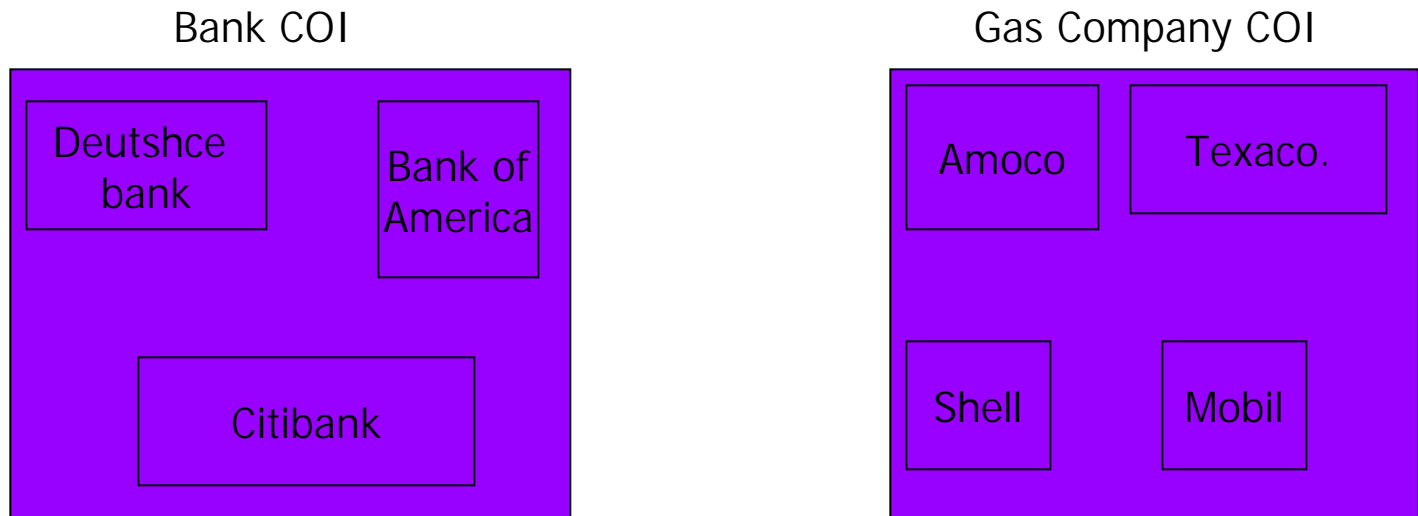
Chinese Wall Model

- The Chinese Wall Model is a model of a security policy that speaks equally to confidentiality and integrity. It describes policies that involve a conflict of interest in Business. For example:
 - In the environment of a stock exchange or investment house the goal of the model is to prevent a conflict of interest in which a trader represents two clients, and the best interests of the clients conflict, so the trader could help one gain at the expense of the other.

Chinese Wall Model

- The *objects* of the database are items of information related to a company.
- A *company dataset* (CD) contains objects related to a single company.
- A *conflict of interest class* (COI) contains the datasets of companies in competition.
- $COI(O)$ represents the conflict of interest class that contains object O .
- $CD(O)$ represents the company dataset that contains object O . The model assumes that each object belongs to exactly one conflict of interest class.

Chinese Wall Model



Anthony has access to the objects in the CD of Bank of America.
Because the CD of Citibank is in the same COI as that of Bank of America, Anthony cannot gain access to the objects in Citibank's CD.
Thus, this structure of the database provides the required ability.

A General Question

- Given a computer system, how can we determine if it is secure? More simply, is there a generic algorithm that allows us to determine whether a computer system is secure?
- What policy shall define “secure?” For a general result, the definition should be as broad as possible – access control matrix with some basic operations and commands.

Formalizing the question

- When a generic right r is added to an element of the access control matrix not already containing r , that right is said to be *leaked*.
- Let a computer system begin in protection state s_0 . If a system can never enter leak the right r , the system (including the initial state s_0) is called *safe with respect to the right* r . If the system can enter an unauthorized state, it is called *unsafe with respect to the right* r .
- Our question (called the *safety question*) :
- Does there exist an algorithm to determine whether a given protection system with initial state s_0 is safe with respect to a generic right r ?

Fundamental Results of Security

- There exists an algorithm that will determine whether a given mono-operational protection system with initial state s_0 is safe with respect to a generic right r .
 - By enumerating all possible states we determine whether the system is safe. It is computationally infeasible, (*NP*-complete) but still it can be done in principle.
- Unfortunately, this result does not generalize to all protection systems.