CS 392/ CS 681 - Computer Security

Nasir Memon – Polytechnic University
Module 7 – Security Policies
Course Logistics

- Security Week
- Questions about Midterm grading
- Read parts of chapters 4, 5, 6 and 7.
- Homework Due
A security policy is a set of rules stating which actions or 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.
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.
  - 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.

- We have learnt some cryptographic and non-cryptographic mechanisms.
Types of Security Policies

- Two types of security policies have been well studied in the literature:
  - 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 integrity.
    - Focus on how much the object can be trusted.

- Also called confidentiality policy and integrity policy.
CS Department Security Policy

http://cis.poly.edu/security-policy.html
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. They are 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.
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 L(s) (clearance) and L(o) (classification).
- Security classes are ordered by a relation $\leq$
- Combines mandatory and discretionary access control.
A basic confidentiality classification system. The four levels are arranged on the list from most sensitive at top and least sensitive at bottom. In the middle are individuals grouped by their security clearance and at the right are documents grouped by their security level.

So Nasir cannot read personnel files and David can read any file. But what if Bill reads contents of personnel files and writes them onto a class file?
BLP – Simple Version

- Two properties characterize the secure flow of information:
  
  - *Simple Security Property:* A subject $s$ may have read access to an object $o$ if and only if $L(o) \leq L(s)$ and $s$ has discretionary read access to $o$.
    
    (Security clearance of subject has to be at least as high as that of the object).
  
  - *-Property:* A subject $s$ who has read access to an object $o$ may have write access to an object $p$ only if $L(o) \leq L(p)$ and $s$ has discretionary write access to $o$.
    
    (Contents of a sensitive object can only be written to objects at least as high. That is, prevent write-down).
Basic Security Theorem: Let $\Sigma$ be a system with a secure initial state $\sigma_0$ and let $T$ be a set of transformations. If every element of $T$ preserves the simple security property and $\ast$-property, then every state $\sigma_i$, $i \geq 0$, is secure.
BLP - Communicating with Subjects at a Lower Level

- If Alice wants to talk to Bob who is at a lower level how does she write a message to him?
- BLP allows this by having notion of maximum security level and current security level.
- Maximum security level must dominate current security level.
- A subject may effectively decrease its security level in order to communicate with entities at lower level.
BLP – Extending to Categories

- 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\}).
We can now define a new relationship to capture the combination of security level and category set:

\[(L, C) \text{ dom } (L', C') \text{ if and only if } L' \leq L \text{ and } C' \subseteq C.\]

This relationship also induces a lattice on the set of compartments.

Example: George is cleared for \{SECRET, \{NUC, EUR\}\}, DocA is classified as \{CONFIDENTIAL, \{NUC\}\}, DocB as \{SECRET< \{EUR, US\}\} and DocC as \{SECRET, \{EUR\}\}. George dom DocA and DocC but not DocB.
Relations and Orderings.

- For a set S, a relation R is any subset of S x S. For (a, b) in R we write aRb.
- A relation defined over S is said to be:
  - Reflexive – if aRa for all a in S.
  - Transitive – If aRb and bRc, then aRc. For a, b, c in S.
  - Anti-symmetric – If aRb and bRa, then a = b for all a, b in S.
- For a, b in S, if there exists u in S such that aRu and bRu, then u is an upper bound of a and b.
- Let U be the set of upper bounds of a and b. Let u in U such that there is no t in U where tRu. The u is the least upper bound of a and b.
- Similarly define lower bound and greatest lower bound.
Lattices

- A partial ordering occurs when a relation orders some, but not all, elements of a set.
- A lattice is a set of elements $S$ and a relation $R$ defined on the elements in $S$ such that
  - $R$ is reflexive, antisymmetric and transitive.
  - For every $s, t$ in $S$ there exists a lub.
  - For every $s, t$ in $S$ there exists a gub.
Examples

- The set \( \{0, 1, 2\} \) forms a lattice under the relation “less than equal to” i.e. \( \leq \)
- The set of integers form a lattice under the relation \( \leq \)
- Is \( B \leq G \)? Is \( B \leq E \)?
Example BLP Lattice

The set of categories form a lattice under the subset $\subseteq$ operation
BLP * Property

- Two properties characterize the secure flow of information:
  - *Simple Security Property:* A subject $s$ may have read access to an object $o$ if and only if $C(s) \text{ dom } C(o)$ and $s$ has discretionary read access to $o$.
  - *Property:* A subject $s$ who has read access to an object $o$ may have write access to an object $p$ only if $C(p) \text{ dom } C(o)$ and $s$ has discretionary write access to $o$.

- Basic Security Theorem: Let $\Sigma$ be a system with a secure initial state $\sigma_0$ and let $T$ be a set of transformations. If every element of $T$ preserves the simple security property and *-property, then every state $\sigma_i$, $i \geq 0$, is secure.
Tranquility

- Principle of tranquility states that subjects and objects may not change their security level once instantiated.
- Principle of strong tranquility states that security levels do not change during the lifetime of the system.
- Principle of weak tranquility states that security levels do not change in a way that violates the rules of a given security policy.
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 – Information Transfer Path

- An information transfer path is a sequence of objects $o_1, \ldots, o_{n+1}$ and a corresponding sequence of subjects $s_1, \ldots, s_n$ such that $s_i$ reads $o_i$ and $s_i$ writes $o_{i+1}$ for all $i$ between 1 and $n$ (both end points inclusive).

- Different access policies along path possible –
  - Low-Water Mark Policy
  - Ring Policy
  - Strict Integrity Policy
Low-Watermark Policy

- $s$ in $S$ can write to $o$ in $O$ if and only if $i(o) \leq i(s)$.
- If $s$ in $S$ reads $o$ in $O$, then $i'(s)$ is taken to be the minimum of $i(s)$ and $i(o)$ where $i'(s)$ is the subject's integrity level after the read.
- $s_1$ can execute $s_2$ if and only if $i(s_2) \leq i(s_1)$.

The policy prevents direct modification that would lower integrity labels as well as indirect modifications.

Can be too restrictive.
Ring Policy

- Any subject may read any object, regardless of integrity levels.
- $s$ can write to $o$ if and only if $i(o) \leq i(s)$.
- $s_1$ can execute $s_2$ if and only if $i(s_2) \leq i(s_1)$. 
Strict Integrity Policy

- s can read o if and only if $i(s) \leq i(o)$.
- s can write to o if and only if $i(o) \leq i(s)$.
- $s_1$ can execute $s_2$ if and only if $i(s_2) \leq i(s_1)$.
Biba security theorem

- Enforcing any of the three policies described above results in the following security theorem:

- **Theorem**: If there is an information transfer path from object $o_1$ to object $o_{n+1}$, then enforcement of the low-water mark policy (or ring policy or strict integrity policy) requires that $i(o_{n+1}) \leq i(o_1)$ for all $n > 1$. 
Lipner’s Model

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 due to 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 accountability.
Lipner’s Integrity Matrix Model

- Combines confidentiality (BLP) and Integrity (Biba).
- Provides two security levels
  - Audit Manager (AM)
  - System Low
- Defines five categories (compartments)
  - Development (D)
  - Production Code (PC)
  - Production Data (PD)
  - System Development (SD)
  - Software Tools (T)
## Table: Lipner Subject Security Levels

<table>
<thead>
<tr>
<th>Users</th>
<th>Clearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordinary Users</td>
<td>{SL, {PC, PD}}</td>
</tr>
<tr>
<td>Application Developers</td>
<td>{SL, {D, T}}</td>
</tr>
<tr>
<td>System Programmers</td>
<td>{SL, {SD, T}}</td>
</tr>
<tr>
<td>System Controllers</td>
<td>{SL, {D, PC, PD, SD, T}} and downgrade privilege.</td>
</tr>
<tr>
<td>System Management and Staff</td>
<td>{AM, {D, PC, PD, SD, T}}</td>
</tr>
</tbody>
</table>
### Lipner Object Security Levels

<table>
<thead>
<tr>
<th>Objects</th>
<th>Class(es)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development Code Test Data</td>
<td>{SL, {D, T}}</td>
</tr>
<tr>
<td>Production Code</td>
<td>{SL, {PC}}</td>
</tr>
<tr>
<td>Production Data</td>
<td>{SL, {PC, PD}}</td>
</tr>
<tr>
<td>Software Tools</td>
<td>{SL, {T}}</td>
</tr>
<tr>
<td>System Programs</td>
<td>{SL, {Null}}</td>
</tr>
<tr>
<td>System Programs in Modification</td>
<td>{SL, {SD, T}}</td>
</tr>
<tr>
<td>System and Application Logs</td>
<td>{AM, {appropriate categories}}</td>
</tr>
</tbody>
</table>
Completeness of Lipner Model

- We can show that Lipner matrix model as described above can meet all the Lipner requirements stated earlier - see text for why (page 174 – 175).
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:

$$D + YB - W = TB$$
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.
The Clark-Wilson Model defines data subject to its integrity controls as *constrained data items* or CDIs.

Data not subject to the integrity controls are called *unconstrained data items*, or UDIs.

*Integrity verification procedures*, or IVPs, test that the CDIs conform to the integrity constraints at the time the IVPs are run. In this case, the system is said to be in a *valid state*.

*Transformation procedures*, or TPs, change the state of the data in the system from one valid state to another; TPs implement well-formed transactions.
Certification Rule 1 (CR1): When any IVP is run, it must ensure that all CDIs are in a valid state.

Certification Rule 2 (CR2): For some associated set of CDIs, a TP must transform those CDIs in a valid state into a (possibly different) valid state.

- CR2 defines a relation *certified C* that associates a set of CDIs with a particular TP;

Enforcement Rule 1 (ER1): The system must maintain the *certified* relations, and must ensure that only TPs certified to run on a CDI manipulate that CDI.
**CW Model**

- **Enforcement Rule 2 (ER2):** The system must associate a user with each TP and set of CDIs. The TP may access those CDIs on behalf of the associated user. If the user is not associated with a particular TP and CDI, then the TP cannot access that CDI on behalf of that user.
  
  - This defines a set of triple \((user, TP, \{ CDI \text{ set} \})\) to capture the association of users, TPs, and CDIs. Call this relation *allowed* A. Of course, these relations must be certified:
CW Model

- **Enforcement Rule 3 (ER3):** The system must authenticate each user attempting to execute a TP.

- **Certification Rule 4 (CR4):** All TPs must append enough information to reconstruct the operation to an append-only CDI.

- **Certification Rule 5 (CR5):** Any TP that takes as input a UDI may perform only valid transformations, for all possible values of the UDI. The transformation either rejects the UDI or transforms it into a CDI.

- **Enforcement Rule 4 (ER4):** Only the certifier of a TP may change the list of entities associated with that TP. No certifier of a TP, or of an entity associated with that TP, may ever have execute permission with respect to that entity.
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.
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.
Chinese Wall Model

- Let \( PR(S) \) be the set of objects that \( S \) has read:
- **CW-simple security rule**: \( S \) can read \( O \) if and only if either:
  - There exists an object \( O' \) such that \( CD(S) = CD(O') \) and \( CD(O') = CD(O) \); or
  - For all objects \( O' \), \( O' \in PR(S) \Rightarrow COI(O') \neq COI(O) \).
- Hence the minimum number of subjects needed to access every object in a COI is the same as the number of CDs in that COI.
**Chinese Wall Model**

- In practice, companies have information they can release publicly, such as annual stockholders’ reports or filings before government commissions.

- Hence, *CW-simple security rule*: $S$ can read $O$ if and only if either:
  - There exists an object $O'$ such that $CD(S) = CD(O')$ and $CD(O') = CD(O)$; or
  - For all objects $O'$, $O' \in PR(S) \Rightarrow COI(O') = COI(O)$.
  - $O$ is a sanitized object
Chinese Wall Model

- Suppose Anthony and Susan work in the same trading house. Anthony can read objects in Bank of America’s CD, and Susan can read objects in Citibank’s CD. Both can read objects in ARCO’s CD. If Anthony can also write objects in ARCO’s CD, then he can read information from objects in Bank of America’s CD, write it to objects in ARCO’s CD, and then Susan can read that information;

- \( CW-* \) Property Rule: A subject \( S \) may write to an object \( O \) if and only if all of the following conditions hold:
  - The CW-simple security rule permits \( S \) to read \( O \); and
  - For all unsanitized objects \( O' \), \( S \) can read \( O' \) \( \Rightarrow \) \( CD(O') = CD(O) \).
Critical Information Systems Security Policy

- Policy for health information systems (Anderson).

- A *patient* is the subject of medical records, or an agent for that person who can give consent for the person to be treated.

- *Personal Health Information* is information about a patient's health or treatment enabling that patient to be identified.

- A *clinician* is a health-care professional who has access personal access to personal health information while performing their jobs.
Access Principles

- Each medical record has an access control list naming the individuals or groups who may read and append information to the record. The system must restrict access to those identified in the list.

- One of the clinicians on the access control list (responsible clinician) must have the right to add other clinicians to the access control list.

- The responsible clinician must notify the patient of the names on the access control list whenever the patients medical record is opened. Except for situations given in Statutes or in cases of emergency the responsible clinician must obtain the patients consent.
Security Policies in Practice

- A security policy is essentially a document stating security goals, and which actions are required, which are permitted and which are allowed.
  - Policies may apply to actions by a system, by management procedures, by employees, by system users.
  - A complete security policy is a collection policies on specific security issues.
Security Policy: General Principals

- Security policies are detailed, written documents
  - There are usually multiple documents describing policy on specific areas; e.g., “Internet usage by employees”, “Security patch installation policy”, “Password selection and handling policy” etc.
- Top level policies are often determined by management with significant input from IT: they represent the agency or corporate goals and principals
- It is important that the policies be distributed to those who have to follow the policy and/or implement the policy enforcement method.
- It is critical that employees be made aware of policies that affect their actions, violations of which may result in reprimand, suspension, or firing. The fact that individual employees have been made aware should be documented, e.g., by having the employee sign a statement that they attended a training session.
- Every policy must have an enforcement mechanism
Basic Policy Requirements for Employee Policies

Basic Policy Requirements

- Policies must:
  - be implementable and enforceable
  - be concise and easy to understand
  - balance protection with productivity

- Policies should:
  - state reasons why policy is needed
  - describe what is covered by the policies
  - define contacts and responsibilities
  - discuss how violations will be handled
Who Should Be Concerned About Security Policy

- Managers
- System designers
- Users: what are the policy’s impacts on their actions, and what are the ramifications of not following policy
- System administrators, support personnel who manage enforcement technologies and processes
- Company lawyers: they may have to use the written policies in support of actions taken against employees in violation
Inclusive versus Exclusive Policies

- Inclusive polices explicitly state what is allowed, and all other actions are prohibited
  - “Employees may only use the Internet from corporate systems for business related email and web browsing”
  - “Employees may only use the Internet from corporate systems for business related email and web browsing. Occasional personal email and browsing are permitted as long as it does not impact employee performance, corporate system performance and does not include any pornography, illegal activities, or other materials detrimental to the corporation or its perception by the public”

- Exclusive policies explicitly state what is prohibited
  - “Employees may not use email or web browsers from corporate systems for personal use.”
  - “Employees may not use email or web browsers from corporate systems for pornography, illegal activities or other materials detrimental to the corporation or its perception by the public”
Inclusive versus Exclusive Policies (continued)

- Inclusive policies provide automatic prohibition for new applications, technologies, (some) attacks, etc. without changing policy
  - Downloading copyright material for personal use
  - Instant Messaging
- Inclusive policies may need to be updated and updates distributed whenever a new application, technology, etc. comes along
- It is a matter of (high level) corporate policy whether to use inclusive or exclusive policies
Examples of Policy Areas

- Employee email usage
- Employee web browsing usage
- Privacy of user information
- Password selection and protection
- Handling of proprietary information
- Cryptographic policy (what needs to be encrypted, what algorithms/implementations/key lengths to use)
- Remote Access
- Protection of employee issued laptops (physical and network connections)
Examples of Policy Areas--
System Management

- Configuration Management
- Ongoing Security Monitoring
- Security Patch Management
- Incident Response
- Business Continuity
- Security Audit
Security Policies are at Multiple Levels

- High level policies are “human readable”
- High level policies are often at an organizational level and apply to all systems
- High level policies may be refined into multiple low level policies that are apply to system actions, management processes, and actions by employees/users
  - For example, a top level policy on protection of sensitive information may include lower level policies on access control lists (system actions), determining the sensitivity level of information (management processes), and who an employee may discuss the information with (employee actions)
  - Lower level policies may be specific to individual systems
Security Policies are at Multiple Levels (continued)

- Multiple levels of a policy may be in a single document, but the development of the complete policy is “top down”
- This refinement process level policies may be integrated into the system design process
  - For example, you cannot define a firewall policy until you know your system will use a firewall as enforcement mechanism for a higher level policy
- “High level” and “lower level” policy is not a standard terminology--this is a useful just a way to think about policies
- Some authors only consider the high level policies as “policies”
Example of Hierarchical Policies

- **High level:** “company proprietary information shall be protected from release to unauthorized personnel”

- **Mid level procedural policy**
  - All proprietary information shall have a committee responsible for its control
  - A member must authorize any distribution of material
  - Enforcement: training, audit

- **Mid level technology policy:**
  - Proprietary information may only be stored on protected systems, accessible only to those with authorized access. There shall be no externally initiated, automated means to retrieve information from the protected systems
    - Low level; e.g., a firewall rule blocking incoming traffic on ports 20 (ftp data), 21 (ftp control), and 69 (tftp)
    - The firewall is the enforcement mechanism
Security Policies and Systems Engineering

- Top level policies are usually at an organizational, not system level
  - Such policies typically exist before a system development process begins
  - They reflect general organizational policies and goals, such as the handling of classes of sensitive data used broadly in the organization, not just in a single system
  - Top level polices lead to top level system requirements in the initial requirements phase
  - All organizational policies should be reviewed for relevancy at the start of the systems engineering process
At the start of the system design process proceeds, the top level policies may impact the requirements (and hence architecture and design).

As the system design process proceeds, the architecture of the system will lead to system specific policy interpretations of the top level policy, labeled the “system policy” on the next slide.

The system policy in turn is an input into the system and security design.

This may be iterated (depending on the systems engineering model used), with policy refinements occurring as the design is refined.

- At what point does this become requirements allocation and not policy refinement? There is no set rule...
- Conformance with policy is part of the assessment at each iteration.
Source of Sample Policy Documents and Information

- The SANS (SysAdmin, Audit, Network, Security) Institute has sample security policies available on-line in many areas. These can be downloaded and used as is, or modified to the needs of a specific company
  - [http://www.sans.org/resources/policies/](http://www.sans.org/resources/policies/)
- IETF Site Security Handbook (polices for systems admins)
- NIST web site: lots of material on security: technology, best practices, policies, regulations, etc. A search for “security policy” on that site got 6090 hits
  - [csrc.nist.gov](http://csrc.nist.gov)
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 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 $r$?
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.
Fundamental Results of Security

- It is undecidable whether a given state of a given protection system is safe for a given generic right.
  - For example, the protection system of the Unix OS, requires more than one command to be represented by the model used. Hence it is undecidable whether it is secure!