BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Cyber Security

Security Architecture: Policies, Models and Mechanisms

Dr. Ramakrishna Dantu

Associate Professor, BITS Pilani

Disclaimer and Acknowledgement

- The content for these slides has been obtained from books and various other source on the Internet
- I here by acknowledge all the contributors for their material and inputs.
- I have provided source information wherever necessary
- I have added and modified the content to suit the requirements of the course

Agenda

- Introduction to security policies, models and mechanisms
- The Nature of Security Policies
- Types of Security Policies
- The Role of Trust
- Types of Access Control
- Policy Languages
- The CIA Classification:
  - Confidentiality Policies:
  - Integrity Policies:
  - Availability Policies:

The Nature of Security Policies

The Nature of Security Policies

Terms

- Security Policy
- Secure System
- Breach of Security
  - Confidentiality, Integrity, and Availability
- Security Mechanism
- Policy Model

The Nature of Security Policies

Overview

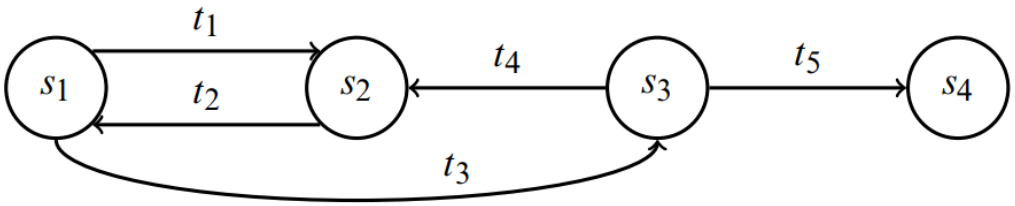
- Consider a computer system to be a finite-state automaton with a set of transition functions that change state, then:
- Definition:
  - A *security policy* is a statement that partitions the states of the system into a set of *authorized (*or *secure)*, states and a set of *unauthorized (*or *non-secure)*, states
- A security policy sets the context to define a *secure system*
- What is secure under one policy may not be secure under a different policy
- Definition:
  - A *secure system* is a system that starts in an authorized state and cannot enter an unauthorized state

The Nature of Security Policies

innovate achieve lead

Overview

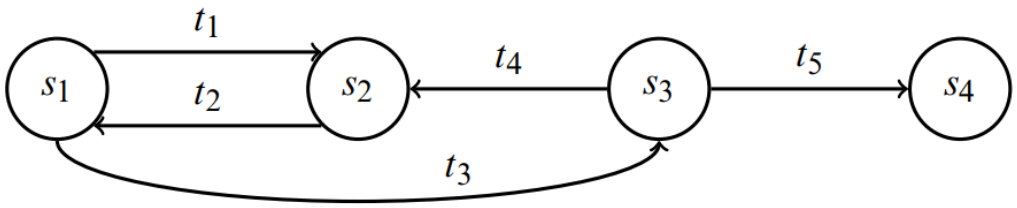
- Consider the finite-state machine
- It consists of four states and five transitions



- According the security policy:
  - A = {s_1, s_2} is a set of authorized states and
  - UA = {s_3, s_4} is a set of unauthorized states
- Is this a secure system?

The Nature of Security Policies

Overview



- This system is not secure because regardless of which authorized state it starts in, it can enter an unauthorized state
- However, if the edge from s_1 to s_3 were not present, the system would be secure, because it could not enter an unauthorized state from an authorized state
- **Definition:**
  - A *breach of security* occurs when a system enters an unauthorized state.

The Nature of Security Policies

Confidentiality

- Definition:
  - Let X be a set of entities and let 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
- Confidentiality implies that information:
  - must not be disclosed to some set of entities
  - it may be disclosed to others
- The membership of set X is often implicit (understood)
  - For example, when we speak of a document that is confidential,
    - all entities not authorized to have such access make up the set X

The Nature of Security Policies

Integrity

- Definition:
  - Let X be a set of entities and let I be some information or resource
  - Then I has the property of *integrity* with respect to X if all members of X trust I
- In addition, members of X also trust that the transmission and storage of I do not change the information or its trustworthiness
  - This aspect is sometimes called *data integrity*
- If I is information about the origin of something, or about an identity, the members of X trust that the information is correct and unchanged
  - This aspect is sometimes called *origin integrity* or, *authentication*
- If I is a resource (E.g., database or application), then integrity means that the resource functions correctly (meeting its specifications)
  - This aspect is called *assurance*

The Nature of Security Policies

Availability

- Definition
  - Let X be a set of entities and let I be a resource
  - Then I has the property of *availability* with respect to X if all members of X can access I
- The exact definition of "access" varies depending on:
  - the needs of the members of X,
  - the nature of the resource, and
  - the use to which the resource is put
- Example:
  - If a book-selling server takes up to 20 minutes to service a book purchase request, that may meet the client's requirements for "availability."
  - If a server of medical information takes up to 10 minutes to provide allergy information of a patient to an anesthetist, that will not meet an emergency room's requirements for "availability."

The Nature of Security Policies

Confidentiality Policy

- With respect to confidentiality,
  - a security policy identifies all the states in which information leaks to those who are not authorized to receive it
  - This includes the leakage of rights and the illicit transmission of information without leakage of rights, called *information flow*
- The policy must also handle changes of authorization, so it includes a temporal element
- For example:
  - A contractor working for a company may be authorized to access proprietary information during the lifetime of a nondisclosure agreement, but when that nondisclosure agreement expires, the contractor can no longer access that information
- This aspect of the security policy is often called a *confidentiality policy*

The Nature of Security Policies

Integrity Policy

- With respect to integrity,
  - a security policy identifies authorized ways in which information may be altered and entities authorized to alter it
- Authorization may derive from a variety of relationships, and external influences may constrain it
- For example:
  - In many transactions, a principle called *separation of duties* forbids an entity from completing the transaction on its own
- Those parts of the security policy that describe the conditions and manner in which data can be altered are called the *integrity policy*

The Nature of Security Policies

Availability Policy

- With respect to availability,
  - a security policy describes the availability details of various services
- It may present parameters within which the services will be accessible. For example:
  - A browser may download web pages but not Java applets
  - A web browser may not support adobe flash
- It may describe a level of service. For example
  - A server will provide authentication data within 1 minute of the request being made
- Those parts of the security policy that
  - discuss the conditions and manner in which systems and services must be available is called the *Availability policy*

The Nature of Security Policies

Desired Properties of the System

- Typically, the security policy assumes that the reader understands the context in which the policy is issued:
  - in particular, the laws, organizational policies, and other environmental factors
- The security policy then describes conduct, actions, and authorizations defining "authorized users" and "authorized use."
- EXAMPLE
  - A university disallows cheating, which is defined to include copying another student's homework assignment (with or without permission)
  - A computer science class requires the students to do their homework on the department's computer
  - Student A notices that student B has not read-protected the file containing her homework and copies it
    - Has either student (or have both students) breached security?

The Nature of Security Policies

Desired Properties of the System

- Student B
  - The student has not breached security, despite her failure to protect her homework
  - The security policy requires no action to prevent files from being read
  - She may have been too trusting, but the policy does not ban this
  - Thus, student B has not breached security
- Student A
  - The student has breached security
  - The security policy disallows the copying of homework, and the student has done exactly that
- Whether the security policy specifically states that:
  - "files containing homework shall not be copied" or simply says that
  - "users are bound by the rules of the university"

  is irrelevant
- If the security policy is silent on such matters, the most reasonable interpretation is that the policy disallows actions that the university disallows, because
  - the computer science department is part of the university

The Nature of Security Policies

Security Mechanism

- Definition:
  - A *security mechanism* is an entity or procedure that enforces some part of the security policy
- Example
  - In the preceding example, the policy states that no student may copy another student's homework
  - One mechanism is the file access controls
    - If the student B had set permissions to prevent the student A from reading the file containing her homework, then A could not have copied that file

The Nature of Security Policies

Procedural or Operational Security Mechanisms - Example

- A site's security policy states:
  - "Information relating to a particular product is proprietary and is not to leave the control of the company"
- The company stores its backup tapes in a vault in the town's bank
- The company must ensure that only authorized employees have access to the backup tapes even when the tapes are stored off-site
- The bank's controls on access to the vault, and the procedures used to transport the tapes to and from the bank, are considered *security mechanisms*
- These mechanisms are not technical controls built into the computer
- Procedural, or operational, controls also can be *security mechanisms*

The Nature of Security Policies

Security Mechanism - Example

- The UNIX operating system, initially developed for a small research group, had mechanisms sufficient to prevent users from accidentally damaging one another's files
  - For e.g., the user A could not delete the user B's files (unless B had set the files and the containing directories to allow this)
- The implied security policy for this "friendly" environment was
  - "do not delete or corrupt another's files, and any file not protected may be read."
- When the UNIX operating system moved into academic, commercial, and government environments, the previous security policy became inadequate
  - For e.g., some files had to be protected from individual users (rather than from groups of users)
- Similarly, the security mechanisms were inadequate for those environments

Types of Security Policies

Types of Security Policies

Policy Model

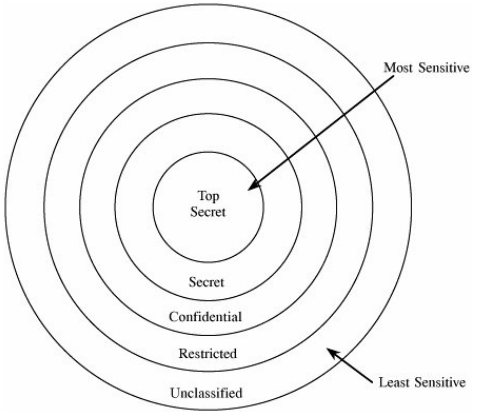
- Every ite has its own requirements for the levels of confidentiality, integrity, and availability
- The site security policy states these needs for that particular site
- Types of Security Policies
  - Military (or governmental) Security Policy
    - Policy primarily protecting confidentiality
  - Commercial Security Policy
    - Policy primarily protecting integrity
    - Transaction-oriented integrity security policy
  - Confidentiality Policy
    - Policy protecting only confidentiality
  - Integrity Policy
    - Policy protecting only integrity

Types of Security Policies

innovate achieve lead

Military Security Policy

- A military security policy (or a governmental security policy) is concerned with protecting classified information
- It is a security policy developed primarily to provide confidentiality
- Each piece of information is ranked at a particular sensitivity level,
  - such as unclassified, restricted, confidential, secret, or top secret.
- The name comes from the military's need to keep information secret
  - E.g., the date that a troop ship will sail
- Although integrity and availability are important, organizations using this class of policies can overcome the loss of either
  - For example, they can use orders not sent through a computer network
- But the compromise of confidentiality would be catastrophic, because an opponent would be able to plan countermeasures



Hierarchy of Sensitivities.

Image Source: https://flylib.com

Types of Security Policies

Commercial Security Policy

- A *commercial security policy* is a security policy developed primarily to provide integrity
- The name comes from the need of commercial firms to prevent tampering with their data, because they could not survive such compromises
- For example:
  - If the confidentiality of a bank's computer is compromised, a customer's account balance may be revealed
  - This would certainly embarrass the bank and possibly cause the customer to take her business elsewhere
  - But the loss to the bank's "bottom line" would be minor
- However, if the integrity of the computer holding the accounts were compromised, the balances in the customers' accounts could be altered
  - This can lead to financially ruinous effects on the bank

Types of Security Policies

Commercial Security Policy

- Some integrity policies use the notion of a transaction
  - E.g., a database transaction must not leave the database in an inconsistent state
- Like database specifications, they require that actions occur in such a way as to leave the database in a consistent state
- These policies, called *transaction-oriented integrity security policies*, are critical to organizations that require consistency of databases.

Types of Security Policies

Commercial Security Policy – Example

- When a customer moves money from one account to another, the bank uses a well-formed transaction
- This transaction has two distinct parts:
  - money is first debited from the original account and then credited to the second account
- Unless both parts of the transaction are completed successfully,
  - the customer will lose the money
- With a well-formed transaction, if the transaction is interrupted, the state of the database is still consistent
  - Either as it was before the transaction began or as it would have been when the transaction ended
- Hence, part of the bank's security policy is that all transactions must be well-formed

Types of Security Policies

Confidentiality Policy Vs. Integrity Policy

- The difference in these two policies is based on the role of trust in these policies
- Confidentiality policy
  - Places no trust in objects
  - The policy dictates whether the object can be disclosed
  - The policy says nothing about whether the object should be believed or trusted
- Integrity policy
  - Indicate how much the object can be trusted
  - The policy dictates what a subject can do with that object
  - But the crucial question is how the level of trust is assigned

Types of Security Policies

Confidentiality Policy Vs. Integrity Policy – Example

- Consider a site obtains a new version of a software. Should that software have
  - high integrity (that is, the site trusts the new version of that program) or
  - low integrity (that is, the site does not yet trust the new program) or
  - somewhere in between (because the vendor supplied the program, but it has not been tested at the local site as thoroughly as the old version)?
- This makes integrity policies considerably more vague than confidentiality policies
- Assigning a level of confidentiality is based on what the organization wants others to know
- Assigning a level of integrity is based on what the organization subjectively believes to be true about the trustworthiness of the information

Types of Security Policies

Confidentiality Policy Vs. Integrity Policy

- Definition
  - A confidentiality policy is a security policy dealing only with confidentiality
  - An integrity policy is a security policy dealing only with integrity
- Both confidentiality policy and military policy deal with confidentiality
- However, a confidentiality policy does not deal with integrity at all, whereas a military policy may
- A similar distinction holds for integrity policies and commercial policies

innovate achieve lead

The Role of Trust

The Role of Trust

Overview

- The role of trust is crucial to understanding the nature of computer security
- All theories and mechanisms for analyzing and enhancing computer security rely on certain assumptions
- If we understand these assumptions on which security policies, mechanisms, and procedures are based, then
  - we will have a very good understanding of the effectiveness of these policies, mechanisms, and procedures
- Let us examine the consequences of this maxim
  - A system administrator receives a security patch for his computer's operating system. He installs it. Has he improved the security of his system?

Maxim = a short statement of a general truth, principle, or rule for behaviour

The Role of Trust

Assumptions – Informal

- The system administrator has succeeded, given the correctness of certain assumptions:
  - that the patch came from the trusted or known vendor
  - that the patch didn't come from an attacker who is trying to trick him into installing a bogus patch that would actually open security holes
  - that the patch was not tampered with in transit
  - that the vendor tested the patch thoroughly
  - that the vendor's test environment corresponds to his environment
  - that there are no possible conflicts between different patches and patches from different vendors of software that the system is using
  - that the patch is installed correctly

The Role of Trust

Assumptions – Some examples

- The vendor tested the patch thoroughly
  - Vendors are often under considerable pressure to issue patches quickly and sometimes test them only against a particular attack
  - The vulnerability may be deeper and other attacks may succeed
  - When someone released an exploit of one vendor's operating system code, the vendor released a correcting patch in 24 hours
  - Unfortunately, the patch opened a second hole, one that was far easier to exploit
  - The next patch (released 48 hours later) fixed both problems correctly

The Role of Trust

Assumptions – Some examples

- The vendor's test environment corresponds to his environment
  - A vendor's patch once enabled the host's personal firewall, causing it to block incoming connections by default
  - This prevented many programs from functioning
  - The host had to be reconfigured to allow the programs to continue to function

The Role of Trust

Assumptions – Some examples

- The patch is installed correctly
  - Some patches are simple to install, because they are simply executable files
  - Others are complex, requiring the system administrator to
    - reconfigure network-oriented properties, add a user, modify the contents of a registry, give rights to some set of users, and then reboot the system
  - An error in any of these steps could prevent the patch from correcting the problems
    - Something similar to an inconsistency between the environments in which the patch was developed and in which the patch is applied
  - Furthermore, the patch may claim to require specific privileges, when in reality the privileges are unnecessary and in fact dangerous

The Role of Trust

Trust in Formal Verification

- Formal verification gives formal mathematical proof that
  - given input *i*, program *P* produces output *o* as specified in the requirements
- Suppose a security-related program S has been formally verified for the operating system O
- What assumptions are made when it was installed?
  - The formal verification of S is correct—that is, the proof has no errors
  - The preconditions hold in the environment in which the program is to be executed
  - The version of O in the environment in which the program is to be executed is the same as the version of O used to verify S

The Role of Trust

Trust in Formal Methods – Assumptions

- The program will be transformed into an executable
- The actions of the executable correspond to those indicated by the source code
  - In other words, the compiler, linker, loader, and any libraries are correct
- Example
  - An experiment with one version of the UNIX operating system demonstrated how devastating a rigged compiler could be
  - Some attack tools replace libraries with others that perform additional functions, thereby increasing security risks

The Role of Trust

Trust in Formal Methods – Assumptions

- The hardware will execute the program as intended
- Example
  - A program that relies on floating-point calculations would yield incorrect results on some computer CPU chips
    - regardless of any formal verification of the program, owing to a flaw in these chips
  - The Pentium F00F bug
    - The name is shorthand for F0 0F C7 C8, the hexadecimal encoding of one offending instruction
    - A design flaw in the majority of Intel Pentium, Pentium MMX, and Pentium OverDrive processors (all in the P5 microarchitecture)
    - Discovered in 1997, it can result in the processor ceasing to function until the computer is physically rebooted
    - The bug has been circumvented through operating system updates


innovate
achieve
lead

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
ज्ञानं परमं बलम्

Thank You!