**UNIT V : Protection**

## **Goals of Protection in Operating System**

## The primary goals of protection in an operating system (OS) are to ensure the security, integrity, and privacy of the system and its users. Here's a breakdown of these goals:

## 1. **Security**: Protection mechanisms aim to prevent unauthorized access to system resources, such as files, memory, and devices. This includes ensuring that only authorized users can perform certain actions, like reading, writing, or executing files, and that sensitive system functions are protected from tampering or misuse.

## 2. **Integrity**: Protection mechanisms help maintain the integrity of system resources by preventing unauthorized modification or corruption. This involves verifying that data and system configurations have not been altered in unauthorized ways, ensuring that processes cannot interfere with each other's data or code, and detecting and preventing malicious software (malware) from compromising system integrity.

## 3. **Privacy**: Protection mechanisms safeguard the privacy of user data by controlling access to personal information and preventing unauthorized disclosure. This includes enforcing access controls to ensure that only authorized users or processes can access sensitive data, encrypting data to prevent unauthorized interception or eavesdropping, and implementing privacy-preserving measures to minimize the collection and disclosure of personally identifiable information.

## 4. **Resource** **Management**: Protection mechanisms in an OS aim to efficiently manage system resources, such as CPU time, memory, and input/output (I/O) devices, to ensure fair and equitable access among users and processes. By enforcing resource allocation policies and preventing resource contention, an OS can optimize system performance and prevent resource starvation or abuse.

## 5. **Fault Tolerance**: Protection mechanisms help enhance the fault tolerance of the system by detecting and recovering from errors, failures, or hardware malfunctions. This includes implementing error handling and recovery mechanisms, redundancy strategies, and failover mechanisms to maintain system availability and reliability in the face of unexpected failures.

## 6. **Isolation**: Isolation mechanisms in an OS aim to separate processes or components to prevent failures or security breaches from spreading across the system. This involves using techniques such as process isolation, virtualization, and containerization to create boundaries between different software components, ensuring that faults or attacks in one area do not affect the rest of the system.

## 7. **Auditing and Accountability**: Protection mechanisms enable auditing and accountability features to track and monitor system activities, user actions, and security events. By logging relevant information, such as user logins, file accesses, and system configurations, an OS can facilitate forensic analysis, compliance auditing, and incident response to identify security breaches, investigate unauthorized activities, and enforce accountability among users and administrators.

## 8.**Adaptability and Flexibility**: Protection mechanisms should be adaptable and flexible to accommodate evolving security threats, user requirements, and system configurations. This includes supporting configurable security policies, role-based access controls, and security updates to address new vulnerabilities and emerging security challenges effectively.

## 9. **Usability and User Experience**: While prioritizing security and protection, an OS should also aim to maintain a positive user experience and usability. This involves implementing security features that are intuitive, non-intrusive, and transparent to users, minimizing the impact on productivity and usability while ensuring robust protection against security threats.

## **Domain of Protection**

In operating systems, protection mechanisms are typically organized into several domains, each focusing on specific aspects of system security and resource management. Here are the common domains of protection in an OS:

1. **Memory Protection**: Memory protection ensures that each process running on the system can only access memory regions allocated to it. This prevents processes from interfering with each other's data or code, enhancing system stability and security.

2. **File Protection**: File protection controls access to files and directories, allowing users and processes to read, write, or execute files based on their permissions. This domain ensures that sensitive files are only accessible to authorized users and prevents unauthorized modification or deletion of critical system files.

3. **Access Control**: Access control mechanisms enforce security policies by regulating access to system resources, such as files, devices, and network services. This includes authentication, authorization, and auditing mechanisms to verify the identity of users and processes, determine their permissions, and track their actions for accountability purposes.

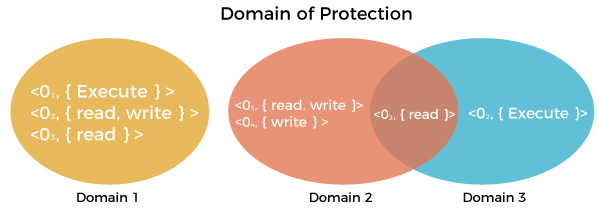
4. **Process Isolation**: Process isolation ensures that processes running on the system are isolated from each other, preventing them from accessing each other's memory or resources without explicit permission. This domain includes features like process sandboxing, virtualization, and containerization to create secure execution environments for applications and services.

5. **Network Security**: Network security mechanisms protect the integrity and confidentiality of data transmitted over computer networks. This includes encryption, authentication, and secure communication protocols to prevent eavesdropping, tampering, and unauthorized access to network traffic.

6. **Device Protection**: Device protection mechanisms control access to hardware devices, such as disk drives, printers, and network interfaces, to prevent unauthorized use or manipulation. This domain ensures that only authorized processes can interact with devices and that device drivers are secure and free from vulnerabilities.

7. **User Authentication**: User authentication verifies the identity of users logging into the system, typically through passwords, biometrics, or multi-factor authentication methods. This domain ensures that only legitimate users can access system resources and prevents unauthorized access by malicious actors.

A domain component is defined as **<object, {set of operations on object}>**.



**Cryptography**

Cryptography is a **technique of encoding information and communications to secure them from unauthorized access, using mathematical algorithms to convert messages into hidden formats, safeguarding data privacy and ensuring secure transactions**. The prefix "**crypt**" means hidden and the suffix "**graphy**" means writing, cryptography relies on mathematical principles and algorithmic rules to transform messages into complex formats, making them resistant to decoding.

**Techniques of Cryptography**

* **Encryption**: The process where an ordinary plain text is converted to cipher text such that only the intended receiver of the text can decode it.
* **Decryption**: The process of conversion of cipher text into plain text.

**Features of Cryptography**

1. **Confidentiality:** Information can only be accessed by the person for whom it is intended and no other person except him can access it.
2. **Integrity:** Information cannot be modified in storage or transition between sender and intended receiver without any addition to information being detected.
3. **Non-repudiation:** The creator/sender of information cannot deny his intention to send information at later stage.
4. **Authentication:** The identities of sender and receiver are confirmed. As well as destination/origin of information is confirmed.

**Types/Methods Of Cryptography**

1. **Symmetric Key Cryptography:** It is an encryption system where the sender and receiver of message use a **single common key** to encrypt and decrypt messages. Symmetric Key Systems are faster and simpler but the problem is that sender and receiver have to somehow exchange key in a secure manner.

**Example:**- Data Encryption Standard(DES) and Advanced Encryption Standard (AES) and Triple DES (3DES).

1. **Hash Functions:** There is **no usage of any key** in this algorithm. A hash value with fixed length is calculated as per the plain text which makes it impossible for contents of plain text to be recovered. Many operating systems use hash functions to encrypt passwords.

**Example:**- SHA-256 (Secure Hash Algorithm 256-bit), MD5 (Message Digest Algorithm 5).

1. **Asymmetric Key Cryptography:** Under this system **a pair of keys** is used to encrypt and decrypt information. A public key is used for encryption and a private key is used for decryption. Even if the public key is known by everyone the intended receiver can only decode it because he alone know his private key. The most popular asymmetric key cryptography algorithm is RSA algorithm.

**Example:**- RSA (Rivest-Shamir-Adleman), Elliptic Curve Cryptography (ECC).

**Applications Of Cryptography**

1. **Computer passwords:**Cryptography is widely utilized in computer security, particularly when creating and maintaining passwords. When a user logs in, their password is hashed and compared to the hash that was previously stored. Passwords are hashed and encrypted before being stored. In this technique, the passwords are encrypted so that even if a hacker gains access to the password database, they cannot read the passwords.
2. **Digital Currencies:** To safeguard transactions and prevent fraud, digital currencies like Bitcoin also use cryptography. Complex algorithms and cryptographic keys are used to safeguard transactions, making it nearly hard to tamper with or forge the transactions.
3. **Secure web browsing:** Online browsing security is provided by the use of cryptography, which shields users from eavesdropping and man-in-the-middle assaults. Public key cryptography is used by the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols to encrypt data sent between the web server and the client establishing a secure channel for communication.
4. **Electronic signatures:**Electronic signatures serve as the digital equivalent of a handwritten signature and are used to sign documents. Digital signatures are created using cryptography and can be validated using public key cryptography. In many nations, electronic signatures are enforceable by law, and their use is expanding quickly.
5. **Authentication:** Cryptography is used for authentication in many different situations, such as when accessing a bank account, logging into a computer, or using a secure network. Cryptographic methods are used to confirm the user’s identity and confirm that they have the required access rights to the resource.
6. **End-to-End Encryption:**End-to-end encryption is used to protect two-way communications like video conversations, instant messages, and email. Even if the message is encrypted, it assures that only the intended receivers can read the message. It is widely used in communication apps like WhatsApp and Signal, and it provides a high level of security and privacy for users.

**Advantages**

1. **Access Control:** Cryptography can be used for access control to ensure that only parties with the proper permissions have access to a resource. Only those with the correct decryption key can access the resource.
2. **Secure Communication:** For secure online communication, cryptography is crucial. It offers secure mechanisms for transmitting private information like passwords, bank account numbers, and other sensitive data over the internet.
3. **Protection against attacks:** Cryptography aids in the defence against various types of assaults, including replay and man-in-the-middle attacks. It offers strategies for spotting and stopping these assaults.
4. **Compliance with legal requirements:**Cryptography can assist firms in meeting a variety of legal requirements, including data protection and privacy legislation.

**Access Matrix**

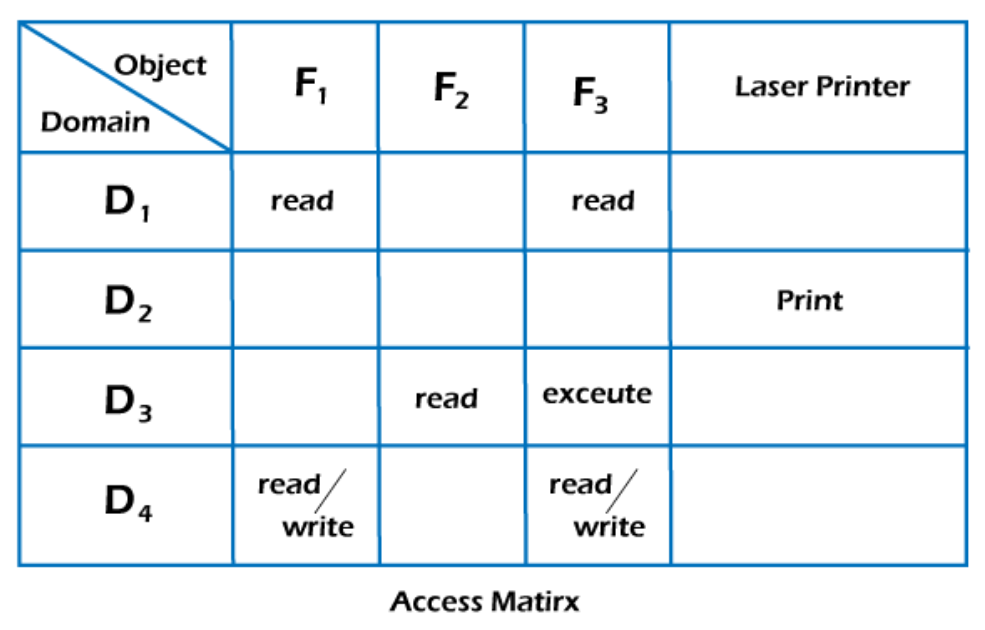
**Access Matrix** is a **security model of protection state in computer system.** An access matrix **helps in the protection of** **operating** **system and is represented by a two-dimensional matrix**.

**Access matrix** is used to **define the rights of each process executing in the domain with respect to each object**. The **rows of matrix represent domains and columns represent objects**. **Each cell of matrix represents set of access rights which are given to the processes**.

**Different types of rights:**

There are different types of rights the files can have. The most common ones are:

1. **Read**- This is a right given to a process in a domain, which allows it to read the file.
2. **Write**- Process in domain can write into the file.
3. **Execute**- Process in domain can execute the file.
4. **Print**- Process in domain only has access to printer.



**Observations of above matrix:**

* There are **four domains** and**four objects**– three files(F1, F2, F3) and one printer.
* A process executing in D1 can read files F1 and F3.
* A process executing in domain D4 has same rights as D1 but it can also write on files.
* Printer can be accessed by only one process executing in domain D2.
* A process executing in domain D3 has the right to read file F2 and execute file F3.

**Switch operation:**When we switch a process from one domain to another, we execute a switch operation on an object (the domain). We can control domain switching by including domains among the objects of the access matrix. Processes should be able to switch from one domain (Di) to another domain (Dj) if and only if a switch right is given to access(i, j). This is explained using an example below:

|  | **F1** | **F2** | **F3** | **Printer** | **D1** | **D2** | **D3** | **D4** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| D1 | read |  | read |  |  | switch |  |  |
| D2 |  |  |  | print |  |  | switch | switch |
| D3 |  | read | execute |  |  |  |  |  |
| D4 | read write |  | read write |  | switch |  |  |  |

According to the above matrix, a process executing in domain D2 can switch to domain D3 and D4. A process executing in domain D4 can switch to domain D1 and process executing in domain D1 can switch to domain D2.

There are various methods of **implementing the access matrix in the operating system**. These methods are as follows:

1. **Global Table**
2. **Access Lists for Objects**
3. **Capability Lists for Domains**
4. **Lock-Key Mechanism**

Global Table

It is the most basic access matrix implementation. A set of ordered triples **<domain, object, rights-set>** is maintained in a file. When an operation **M** has been performed on an object Oj within domain Di, the table is searched for a triple **<Di, Oj, Rk>.** The operation can proceed if this triple is located; otherwise, an exception (or error) condition has arrived. This implementation has various drawbacks. The table is generally large and cannot be stored in the main memory, so additional input and output are required.

Access Lists for Objects

Every access matrix column may be used as a single object's access list. It is possible to delete the blank entries. For each object, the resulting list contains ordered pairs **<domain, rights-set>** that define all domains for that object and a nonempty set of access rights.

We may start by checking the default set and then find the access list. If the item is found, we enable the action; if it isn't, we verify the default set. If M is in the default set, we grant access. Access is denied if this is not the case, and an extraordinary scenario arises.

Capability Lists for Domains

A domain's capability list is a collection of objects and the actions that can be done on them. A capacity is a name or address that is used to define an object. If you want to perform operation M on object **Oj,** the process runs operation M, specifying the capability for object **Oj.** The simple possession of the capability implies that access is allowed.

In most cases, capabilities are separated from other data in one of two ways. Every object has a tag to indicate its type as capability data. Alternatively, a program's address space can be divided into two portions. The programs may access one portion, including the program's normal instructions and data. The other portion is a capability list that is only accessed by the operating system.

Lock-Key Mechanism

It is a compromise between the access lists and the capability lists. Each object has a list of locks, which are special bit patterns. On the other hand, each domain has a set of keys that are special bit patterns. A domain-based process could only access an object if a domain has a key that satisfies one of the locks on the object. The process is not allowed to modify its keys.

### **Association between Process and Domain**

When processes have the necessary access rights, they can switch from one domain to another. It could be of two types, as shown below.

**1. Fixed or Static**

In a fixed association, all access rights could be given to processes at the start. However, the results in a large number of access rights for domain switching. As a result, a technique of changing the domain's contents is found dynamically.

**2. Changing or dynamic**

A process may switch dynamically and creating a new domain in the process.

## **Security measures of Operating System**

There are various security measures of the operating system that the users may take. Some of them are as follows:

1. The network used for file transfers must be secure at all times. During the transfer, no alien software should be able to harvest information from the network. It is referred to as network sniffing, and it could be avoided by implementing encrypted data transfer routes. Moreover, the OS should be capable of resisting forceful or even accidental violations.
2. Passwords are a good authentication method, but they are the most common and vulnerable. It is very easy to crack passwords.
3. Security measures at various levels are put in place to prevent malpractices, like no one being allowed on the premises or access to the systems.
4. The best authentication techniques include a username-password combination, eye retina scan, fingerprint, or even user cards to access the system.

## **System Authentication**

**One-time passwords, encrypted passwords,** and **cryptography** are used to create a strong password and a formidable authentication source.

**1. One-time Password**

It is a way that is unique at every login by the user. It is a combination of two passwords that allow the user access. The system creates a random number, and the user supplies a matching one. An algorithm generates a random number for the system and the user, and the output is matched using a common function.

**2. Encrypted Passwords**

It is also a very effective technique of authenticating access. Encrypted data is passed via the network, which transfers and checks passwords, allowing data to pass without interruption or interception.

**3. Cryptography**

It's another way to ensure that unauthorized users can't access data transferred over a network. It aids in the data secure transmission. It introduces the concept of a key to protecting the data. The key is crucial in this situation. When a user sends data, he encodes it using a computer that has the key, and the receiver must decode the data with the same key. As a result, even if the data is stolen in the middle of the process, there's a good possibility the unauthorized user won't be able to access it.

## **What is Operating System Security?**

The process of ensuring OS availability, confidentiality, integrity is known as operating system security. OS security refers to the processes or measures taken to protect the operating system from dangers, including viruses, worms, malware, and remote hacker intrusions. Operating system security comprises all preventive-control procedures that protect any system assets that could be stolen, modified, or deleted if OS security is breached.

Security refers to providing safety for computer system resources like software, CPU, memory, disks, etc. It can protect against all threats, including viruses and unauthorized access. It can be enforced by assuring the operating system's **integrity, confidentiality**, and **availability**. If an illegal user runs a computer application, the computer or data stored may be seriously damaged.

System security may be threatened through two violations, and these are as follows:

**1. Threat**

A program that has the potential to harm the system seriously.

**2. Attack**

A breach of security that allows unauthorized access to a resource.

There are two types of security breaches that can harm the system: malicious and accidental. Malicious threats are a type of destructive computer code or web script that is designed to cause system vulnerabilities that lead to back doors and security breaches. On the other hand, Accidental Threats are comparatively easier to protect against.

Security may be compromised through the breaches. Some of the breaches are as follows:

**1. Breach of integrity**

This violation has unauthorized data modification.

**2. Theft of service**

It involves the unauthorized use of resources.

**3. Breach of confidentiality**

It involves the unauthorized reading of data.

**4. Breach of availability**

It involves the unauthorized destruction of data.

**5. Denial of service**

It includes preventing legitimate use of the system. Some attacks may be accidental.

## **The goal of Security System**

There are several goals of system security. Some of them are as follows:

**1. Integrity**

Unauthorized users must not be allowed to access the system's objects, and users with insufficient rights should not modify the system's critical files and resources.

**2. Secrecy**

The system's objects must only be available to a small number of authorized users. The system files should not be accessible to everyone.

**3. Availability**

All system resources must be accessible to all authorized users, i.e., no single user/process should be able to consume all system resources. If such a situation arises, service denial may occur. In this case, malware may restrict system resources and preventing legitimate processes from accessing them.

## **Types of Threats**

There are mainly two types of threats that occur. These are as follows:

### **Program threats**

The operating system's processes and kernel carry out the specified task as directed. Program Threats occur when a user program causes these processes to do malicious operations. The common example of a program threat is that when a program is installed on a computer, it could store and transfer user credentials to a hacker. There are various program threats. Some of them are as follows:

**1.Virus**

A virus may replicate itself on the system. Viruses are extremely dangerous and can modify/delete user files as well as crash computers. A virus is a little piece of code that is implemented on the system program. As the user interacts with the program, the virus becomes embedded in other files and programs, potentially rendering the system inoperable.

**2. Trojan Horse**

This type of application captures user login credentials. It stores them to transfer them to a malicious user who can then log in to the computer and access system resources.

**3. Logic Bomb**

A logic bomb is a situation in which software only misbehaves when particular criteria are met; otherwise, it functions normally.

**4. Trap Door**

A trap door is when a program that is supposed to work as expected has a security weakness in its code that allows it to do illegal actions without the user's knowledge.

### **System Threats**

System threats are described as the misuse of system services and network connections to cause user problems. These threats may be used to trigger the program threats over an entire network, known as program attacks. System threats make an environment in which OS resources and user files may be misused. There are various system threats. Some of them are as follows:

**1. Port Scanning**

It is a method by which the cracker determines the system's vulnerabilities for an attack. It is a fully automated process that includes connecting to a specific port via TCP/IP. To protect the attacker's identity, port scanning attacks are launched through Zombie Systems, which previously independent systems now serve their owners while being utilized for such terrible purposes.

**2. Worm**

The worm is a process that can choke a system's performance by exhausting all system resources. A Worm process makes several clones, each consuming system resources and preventing all other processes from getting essential resources. Worm processes can even bring a network to a halt.

**3. Denial of Service**

Denial of service attacks usually prevents users from legitimately using the system. For example, if a denial-of-service attack is executed against the browser's content settings, a user may be unable to access the internet.

## **Threats to Operating System**

There are various threats to the operating system. Some of them are as follows:

### **Malware**

It contains viruses, worms, trojan horses, and other dangerous software. These are generally short code snippets that may corrupt files, delete the data, replicate to propagate further, and even crash a system. The malware frequently goes unnoticed by the victim user while criminals silently extract important data.

### **Network Intrusion**

Network intruders are classified as masqueraders, misfeasors, and unauthorized users. A masquerader is an unauthorized person who gains access to a system and uses an authorized person's account. A misfeasor is a legitimate user who gains unauthorized access to and misuses programs, data, or resources. A rogue user takes supervisory authority and tries to evade access constraints and audit collection.

### **Buffer Overflow**

It is also known as buffer overrun. It is the most common and dangerous security issue of the operating system. It is defined as a condition at an interface under which more input may be placed into a buffer and a data holding area than the allotted capacity, and it may overwrite other information. Attackers use such a situation to crash a system or insert specially created malware that allows them to take control of the system.

## **How to ensure Operating System Security?**

There are various ways to ensure operating system security. These are as follows:

### **Authentication**

The process of identifying every system user and associating the programs executing with those users is known as authentication. The operating system is responsible for implementing a security system that ensures the authenticity of a user who is executing a specific program. In general, operating systems identify and authenticate users in three ways.

**1. Username/Password**

Every user contains a unique username and password that should be input correctly before accessing a system.

**2. User Attribution**

These techniques usually include biometric verification, such as fingerprints, retina scans, etc. This authentication is based on user uniqueness and is compared to database samples already in the system. Users can only allow access if there is a match.

**3. User card and Key**

To login into the system, the user must punch a card into a card slot or enter a key produced by a key generator into an option provided by the operating system.

### **One Time passwords**

Along with standard authentication, one-time passwords give an extra layer of security. Every time a user attempts to log into the One-Time Password system, a unique password is needed. Once a one-time password has been used, it cannot be reused. One-time passwords may be implemented in several ways.

**1. Secret Key**

The user is given a hardware device that can generate a secret id that is linked to the user's id. The system prompts for such a secret id, which must be generated each time you log in.

**2. Random numbers**

Users are given cards that have alphabets and numbers printed on them. The system requests numbers that correspond to a few alphabets chosen at random.

**3. Network password**

Some commercial applications issue one-time passwords to registered mobile/email addresses, which must be input before logging in.

### **Firewalls**

Firewalls are essential for monitoring all incoming and outgoing traffic. It imposes local security, defining the traffic that may travel through it. Firewalls are an efficient way of protecting network systems or local systems from any network-based security threat.

### **Physical Security**

The most important method of maintaining operating system security is physical security. An attacker with physical access to a system may edit, remove, or steal important files since operating system code and configuration files are stored on the hard drive.

## **Operating System Security Policies and Procedures**

Various operating system security policies may be implemented based on the organization that you are working in. In general, an OS security policy is a document that specifies the procedures for ensuring that the operating system maintains a specific level of integrity, confidentiality, and availability.

OS Security protects systems and data from worms, malware, threats, ransomware, backdoor intrusions, viruses, etc. Security policies handle all preventative activities and procedures to ensure an operating system's protection, including steal, edited, and deleted data.

As OS security policies and procedures cover a large area, there are various techniques to addressing them. Some of them are as follows:

1. Installing and updating anti-virus software
2. Ensure the systems are patched or updated regularly
3. Implementing user management policies to protect user accounts and privileges.
4. Installing a firewall and ensuring that it is properly set to monitor all incoming and outgoing traffic.