OPERATING SYSTEM ASSIGNMENT

Submitted by:

(102103267) PULKIT ARORA

BE Second Year (COE 10)

Submitted to: DR. VINAY ARORA

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology

November 2022

Question 1. Define various aspects of Security of/in an OS and list various security Attacks / Program Threats

Answer: -

Aspects of Security in OS

A. Encryption - It is used to send messages securely across a network, as well as to protect database data, files, and even entire disks from having their contents read by unauthorized entities. An encryption algorithm enables the sender of a message to ensure that only a computer possessing a certain key can read the message, or ensure that the writer of data is the only reader of that data.

B. User Authentication - The protection system depends on the ability to identify the programs and processes currently executing, which in turn depends on the ability to identify each user of the system. User Authentication can be done using any of these methods –

    1. Passwords - The most common approach to authenticating a user identity is the use of passwords. When the user identifies herself by user ID or account name, they are asked for a password. If the user-supplied password matches the password stored in the system, the system assumes that the account is being accessed by the owner of that account.
    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.

C. 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 can be implemented in given 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. 3.Network Password - Some commercial applications issue one-time passwords to registered mobile/email addresses, which must be input before logging in.

D. 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.

Various Security Attacks/Program threats

Threats can be classified into the following two categories:

1. Program Threats: A program was written by a cracker to hijack the security or to change the behaviour of a normal process. In other words, if a user program is altered and further made to perform some malicious unwanted tasks, then it is known as Program Threats. Types of Program threats are: -

   A. Virus: An infamous threat, known most widely. It is a self-replicating and malicious thread that attaches itself to a system file and then rapidly replicates itself, modifying and destroying essential files leading to a system breakdown.
   B. Trojan Horse: A code segment that misuses its environment is called a Trojan Horse. They seem to be attractive and harmless cover programs but are really harmful hidden programs that can be used as the virus carrier
   C. Trap Door: The designer of a program or system might leave a hole in the software that only he is capable of using, the Trap Door works on similar principles. Trap Doors are quite difficult to detect as to analyse them, one needs to go through the source code of all the components of the system.
   D. Logic Bomb: A program that initiates a security attack only under a specific situation. A logic bomb is actually the most malicious program that is triggered or functions when specific conditions have been met for it to work.

2. System Threats: These threats involve the abuse of system services. They strive to create a situation in which operating-system resources and user files are misused. They are also used as a medium to launch program threats.

   A. Worm: An infection program that spreads through networks. Unlike a virus, they target mainly LANs. A computer affected by a worm attacks the target system and writes a small program “hook” on it. This hook is further used to copy the worm to the target computer

   B. Port Scanning: It is a means by which the cracker identifies the vulnerabilities of the system to attack. It is an automated process that involves creating a TCP/IP connection to a specific port. To protect the identity of the attacker, port scanning attacks are launched from Zombie Systems, that is systems that were previously independent systems that are also serving their owners while being used for such notorious purposes.

   C. Denial of Service: Such attacks aren’t aimed for the purpose of collecting information or destroying system files. Rather, they are used for disrupting the legitimate use of a system or facility. These attacks are generally network-based.

Question 2. Explain various categories of viruses (related to Operating systems).

Answer –

File: - A standard file virus infects a system by appending itself to a file.
It changes the start of the program so that execution jumps to its code.
After it executes, it returns control to the program so that its execution is
not noticed. File viruses are sometimes known as parasitic viruses, as they
leave no full files behind and leave the host program still functional.

Boot: - A boot virus infects the boot sector of the system, executing every time the system is booted and before the operating system is loaded. It watches for other bootable media and infects them. These viruses are also known as memory viruses, because they do not appear in the file system. Figure 15.5 shows how a boot virus works.

Macro: - Most viruses are written in a low-level language, such as assembly or C. Macro viruses are written in a high-level language, such as Visual Basic. These viruses are triggered when a program capable of executing the macro is run. For example, a macro virus could be contained in a spreadsheet file.

Source code: - A source code virus is a computer virus that attacks source code to corrupt it in some way. It may render a program or operating system unusable, hijack a machine for a given purpose, or generate errors on the system. Viruses of this nature are relatively rare, but can be found in the wild and are sometimes difficult to combat because source code is often not human readable and thus can be hard to repair.

Polymorphic: - Polymorphic viruses are complex file infectors that can create modified versions of itself to avoid detection yet retain the same basic routines after every infection. To vary their physical file makeup during each infection, polymorphic viruses encrypt their codes and use different encryption keys every time. This way, traditional security solutions may not easily catch them because they do not use a static, unchanging code. Polymorphic viruses are usually distributed via spam, infected sites, or through the use of other malware.

Encrypted. An encrypted virus includes decryption code along with the encrypted virus, again to avoid detection. The virus first decrypts and then executes.

Stealth. This tricky virus attempts to avoid detection by modifying parts of the system that could be used to detect it. For example, it could modify the read system calls so that if the file it has modified is read, the original form of the code is returned rather than the infected code.

Tunnelling. A tunnelling virus is a virus that attempts to intercept anti-virus software before it can detect malicious code. A tunnelling virus launches itself under anti-virus programs and then works by going to the operating system’s interruption handlers and intercepting them, thus avoiding detection. Interception programs, which remain in the background of an operating system and catch viruses, become disabled during the course of a tunnelling virus. Some anti-virus programs do find the malicious code attached to tunnel viruses, but they often end up being reinstalled under the tunnelling virus. To combat this, some anti-virus programs use their own tunnelling techniques, which uncover hidden viruses located within computer memories. Multipartite. Virus of this type is able to infect multiple parts of a system, including boot sectors, memory, and files. This makes it difficult to detect and contain.

Armoured. An armoured virus is coded to make it hard for antivirus researchers to unravel and understand. It can also be compressed to avoid detection and disinfection. In addition, virus droppers and other full files that are part of a virus infestation are frequently hidden via file attributes or unviewable file names.

Question 3. Describe various Security Defences including Firewalling systems to

Protect Systems and Networks.

Answer –

1.Security Policy

- The first step toward improving the security of any aspect of computing is to have a security policy.
- Without a policy in place, it is impossible for users and administrators to know what is permissible, what is required, and what is not allowed. The policy is a road map to security, and if a site is trying to move from less secure to more secure, it needs a map to know how to get there.
- Once the security policy is in place, the people it affects should know it well. It should be their guide. The policy should also be a living document that is reviewed and updated periodically to ensure that it is still pertinent and still followed.

2.Vulnerability Assessment

- Used to determine whether a security policy has been implemented correctly.
- Risk assessment is done to value the assets of the entity in question (a program, a management team, a system, or a facility) and determine the odds that a security incident will affect the entity and decrease its value. When the odds of suffering a loss and the amount of the potential loss are known, a value can be placed on trying to secure the entity.
- The core activity of most vulnerability assessments is a penetration test, in which the entity is scanned for known vulnerabilities.
- Networked computers are much more susceptible to security attacks than are standalone systems. Rather than attacks from a known set of access points, such as directly connected terminals, we face attacks from an unknown and large set of access points—a potentially severe security problem. To a lesser extent, systems connected to telephone lines via modems are also more exposed.
- Many programmers believe in security through obscurity, that is no tools should be used to find out vulnerabilities as these programs can be misused in wrong hands.

3.Intrusion Detection

- Strives to detect attempted or successful intrusions into computer systems and to initiate appropriate responses to the intrusions.
- intrusion-detection systems raise an alarm when an intrusion is detected and intrusion-prevention systems blocks the traffic in the system.
- First signature-based detection is called which examines system input or network traffic for specific behaviour patterns (or signatures) known to indicate attacks.
- The second approach, typically called anomaly detection, attempts through various techniques to detect anomalous behaviour within computer systems.

4.Virus Protection

- Antivirus programs are often used to provide this protection. Some of these programs are effective against only particular known viruses. They work by searching all the programs on a system for the specific pattern of instructions known to make up the virus. When they find a known pattern, they remove the instructions, disinfecting the program.
- The best protection against computer viruses is prevention, or the practice of safe computing. Purchasing unopened software from vendors and avoiding free or pirated copies from public sources or disk exchange offer the safest route to preventing infection.

5. Auditing, Accounting and Logging

- All system-call executions can be logged for analysis of program behaviour (or misbehaviour). More typically, suspicious events are logged.
- Authentication failures and authorization failures can tell us quite a lot about break-in attempts.
- Accounting is another potential tool in a security administrator's kit. It can be used to find performance changes, which in turn can reveal security problems.

6. Firewalling to Protect Systems and Networks

- A firewall is a computer, appliance, or router that sits between the trusted and the untrusted. A network firewall limits network access between the two security domains and monitors and logs all connections.
- Some types of Firewalls: -
    1. Personal firewall
    2. Application proxy firewall
    3. XML firewall
    4. System-call firewall

Question 4 - Discuss strategies for Implementing the Access Matrix

Answer -

Access Matrix is a security model of the protective state of a computer system. For each object, the permissions for every process executing in the domain are specified using an access matrix. Access matrix in OS is shown as a two-dimensional matrix, where the columns of the matrix represent objects, while the rows represent domains. Each matrix cell represents a specific set of access rights that are granted to processes of the domain this indicates that each entry access (i, j) specifies the set of actions that a process executing in domain Di may invoke on object O_j. The access matrix implements policy decisions and these policy decisions involve which rights should be included in the (i, j)th entry like reading, writing, and executing.

Various Methods Used to Implement the Access Matrix in OS: -

1. Global Table

The global table is the most basic and simple implementation of the access matrix in the operating system which consists of a set of an ordered triple <domain, object, right-set>. When an operation M is being executed on an object O_j within domain D_i, the global table searches for a triple <Domain (D_i), Object (O_j), right-set (R_k)> where M € R_k. If the triple is present, the operation can proceed to continue, or else a condition of an exception is thrown. There are various drawbacks to this implementation, the main drawback of global table implementation is that because the table is sometimes too large, it cannot be stored in the main memory, that's why input and output are required additionally.

2. Access Lists

In the Access Lists method, the access matrix in OS is divided into columns (Column wise decomposition). When an operation M is being executed on an object O_j within domain D_i, we search for an entry <Domain (D_i), right-set (R_k)**>** with M € R_k in the access list for object O_j. If the triple is present, the operation can proceed to continue, or else we check the initial set. If M is included in the default set, access is allowed; otherwise, access is denied, and an exception is raised.

3. Capability Lists

In the access matrix in the operating system, Capability Lists is a collection of objects and the operations that can be performed on them. The object here is specified by a physical name called capability. In this method, we can associate each row with its domain instead of connecting the columns of the access matrix to the objects as an access list. A capability list is itself a protected object maintained by OS and accessed by the user indirectly.

4. Lock-Key Mechanism

It is a comparison between capability lists and access lists. Every domain has a distinct bit pattern called keys, and every object has a distinct bit pattern called locks. Only if a domain's key matches one of the locks of the object, A process can access it.

In simple words, when a process running in a specific domain (D_i) try to access an object (O_j) then the key of that D_i must match with the lock of that O_j, then only an object can be accessed. The operating system should handle the keys and locks in such a way that any unauthorized access should not be allowed on them.

Question 5: - Explain different techniques for Free-Space Management under Disk Management

Answer –

The system keeps tracks of the free disk blocks for allocating space to files when they are created. Also, to reuse the space released from deleting the files, free space management becomes crucial. The system maintains a free space list which keeps track of the disk blocks that are not allocated to some file or directory. The free space list can be implemented mainly as:

1. **Bitmap or Bit vector –**

A Bitmap or Bit Vector is series or collection of bits where each bit corresponds to a disk block. The bit can take two values: 0 and 1: 0 indicates that the block is allocated and 1 indicates a free block.

- Advantages –Simple to understand.
- Finding the first free block is efficient. It requires scanning the words (a group of 8 bits) in a bitmap for a non-zero word. (A 0-valued word has all bits 0). The first free block is then found by scanning for the first 1 bit in the non-zero word.

2. **Linked List –**

In this approach, the free disk blocks are linked together i.e. a free block contains a pointer to the next free block. The block number of the very first disk block is stored at a separate location on disk and is also cached in memory. A drawback of this method is the I/O required for free space list traversal.

3. **Grouping: -**

This approach stores the address of the free blocks in the first free block. The first free block stores the address of some, say n free blocks. Out of these n blocks, the first n-1 blocks are actually free and the last block contains the address of next free n blocks.

An advantage of this approach is that the addresses of a group of free disk blocks can be found easily.

4. **Counting: -**

This approach stores the address of the first free disk block and a number n of free contiguous diskblocks that follow the first block.

Every entry in the list would contain:

a. Address of first free disk block

b. A number n

5. **Space Maps**

• Oracle’s ZFS file system (found in Solaris and other operating systems) was designed to encompasshuge numbers of files, directories, and even file systems.

In its management of free space, ZFS uses a combination of techniques to control the size of datastructures and minimize the I/O needed to manage those structures.

• First, ZFS creates meta slabs to divide the space on the device into chunks of manageable size. Agiven volume may contain hundreds of meta slabs. Each meta slab has an associated space map.

• ZFS uses the counting algorithm to store information about free blocks. Rather than write countingstructures to disk, it uses log-structured file-system techniques to record them.

• The space map is a log of all block activity (allocating and freeing), in time order, in counting format.

• When ZFS decides to allocate or free space from a meta slab, it loads the associated space map into memory in a balanced-tree structure (for very efficient operation), indexed by offset, and replays the loginto that structure.

• The in-memory space map is then an accurate representation of the allocated and free space in the meta slab. ZFS also condenses the map as much as possible by combining contiguous free blocks into asingle entry.