5 System Security

Module 5

Syllabus

At the end of this unit, you should be able to understand and comprehend the following syllabus topics:

- Operating System Security
  - Memory and Address Protection
  - File Protection Mechanism
  - User Authentication
  - Linux and Windows
    - Vulnerabilities
    - File System Security
- Database Security
  - Database Security Requirements
  - Reliability and Integrity
  - Sensitive Data
  - Inference Attacks
  - Multilevel Database Security

5.1 Operating System Security

- Operating system architecture can be viewed as made up of several layers. Each of these layers operate in a ring mechanism where lower the ring the higher the privileges. The operating system plays a vital role in ensuring that not only access to the hardware devices is secured but also the overall system is stable, reliable and operational to serve the legitimate purposes.
- The OS developers do not know in advance about what the application developers are going to develop in their applications. Hence, the protected ring mechanism ensures that the higher privilege commands and instruction sets are not available to the applications directly. Applications needing to execute privilege commands must make the appropriate system calls to the operating system which can execute the commands on the behalf of such applications.
- Additionally, the ring architecture ensures that malware cannot directly interact with the hardware devices. They must be able to bypass OS security measures in order to compromise the system and carry out attacks.

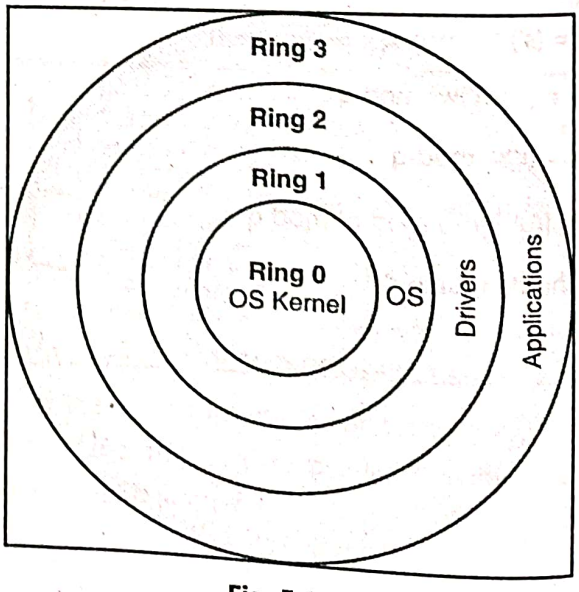


Fig. 5.1.1

- The OS performs several security functions. Some of them are
  - (i) Protect data
  - (ii) Control inter process communication
  - (iii) Isolate processes from each other
  - (iv) Protect hardware
  - (v) Protect OS Kernel
  - (vi) Secure user management [identity, authenticate, authorise and audit]
  - (vi) Secure operations [logs, system calls, interrupts, application management, etc.]

You will learn about specific OS security measures in this section. Let's begin.

5.1.1 Memory and Address Protection

- The core objective of memory protection is to ensure that the processes running on the system are attached and associated with only the memory they are authorised to. One process cannot read or write to the memory address allocated to another process until and unless there is inter-process communication that authorises so. This ensures the integrity of the process data and the process itself.
- There are several mechanisms using which memory protection can be provided. Some of them are shown in Fig. 5.1.2.

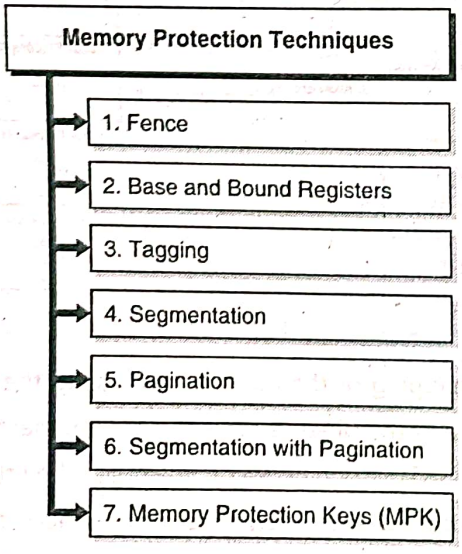


Fig. 5.1.2 : Memory protection techniques

1. Fence

✍ ***Definition :*** *Fencing is a technique in which the memory is divided into two parts – one for the OS and one for the user.*

- This is one the oldest forms of memory protection and is suited only for single-user OS. The entire memory is divided into two parts – one for the use of OS and another one for the user processes. The division of memory could be implemented in hardware or in the OS software. A fence register keeps track of where the user memory starts.

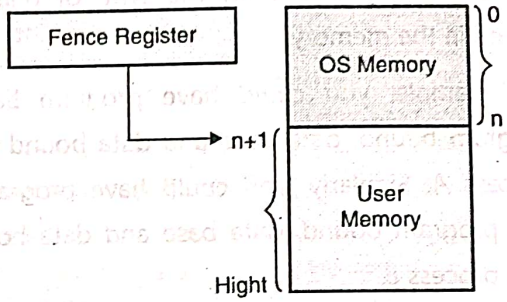


Fig. 5.1.3

> **Note :** A single-user OS means that all the applications and processes run as the same user. So, for example, if you as a user run two processes *A* and *B* then both these processes run with the same user id and can access the entire user memory without any restrictions between the two processes.

2. Base and Bound Registers

✍ ***Definition :*** *Base and Bound Register technique uses two registers to divide the memory and can be used in a multi-user OS environment.*

- One of the limitations of the fence register technique is that it works only for the single-user OS. Most of the OSs today are built for multi-user. The base and bound register technique solves the problem by using two registers :
  - (i) **Base Register :** This register holds the value of the starting address.
  - (ii) **Bound Register :** This register holds the value of the ending address.
- Each of the processes is assigned a base and a bound value. The processes are only permitted to read and write to the address range as defined by their respective base and the bound registers.

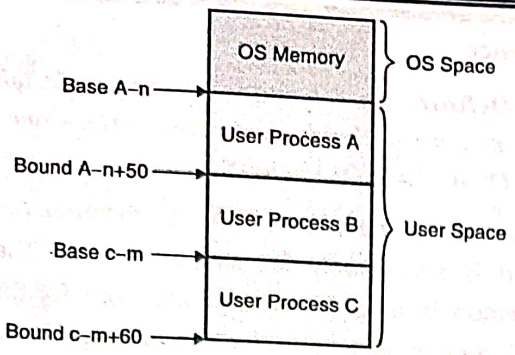


Fig. 5.1.4

- The base and bound mechanism can also be used to divide memory based on the type of data to be stored in the memory.
- For example, you could have program base and program bound, data base and data bound for user process *A*. Similarly, you could have program base and program bound, data base and data bound for user process *B*.

3. Tagging

✍ ***Definition** : Tagging technique assigns properties, using several markings, to each memory address.*

- One of the limitations of the base and bound register technique is that it requires consecutive memory address allocation.
- In tagging technique, each memory address is assigned tags (markings) to denote properties of that memory address. You can add as many tags (markers) for a memory address as required. For example, you could add a tag for the process name, a tag for access rights such as read, write and execute or a tag for the data type such as characters or numerals. Any process that is trying to access a memory address is checked to verify if the tags allow the requested access.

Memory Address	Tags
A	Read-only, process A
B	Read-only, process A
C	Read-only, process A
D	Read and Write, process B
E	Read, Any
F	Write, Any

- This technique has a significant overhead in terms of performance. It is not widely used. This technique is also called Dynamic Tainting.

4. Segmentation

✍ ***Definition** : Segmentation divides the memory by the type of objects it is allowed to hold.*

- You have learnt about segmentation in the buffer overflow section. There are various segments such as heap, stack, and data that hold various objects as required by a process. Each segment is protected from the other segments in the memory.
- Memory in each segment is addressed using two values :
  - **(i) Name :** This provides the segment name in reference.
  - **(ii) Offset :** This provides the address from the base of the segment name in reference.
- So, for example, addressing a memory *M* in the heap segment at the offset of say *10* would mean that you are talking about *M (heap, 10)*.So, if the heap segment starts from the address *1,000* then *M* would translate to *1,000 + 10 = 1,010*. So, the location of *M* would be at *1,010* in the heap segment.

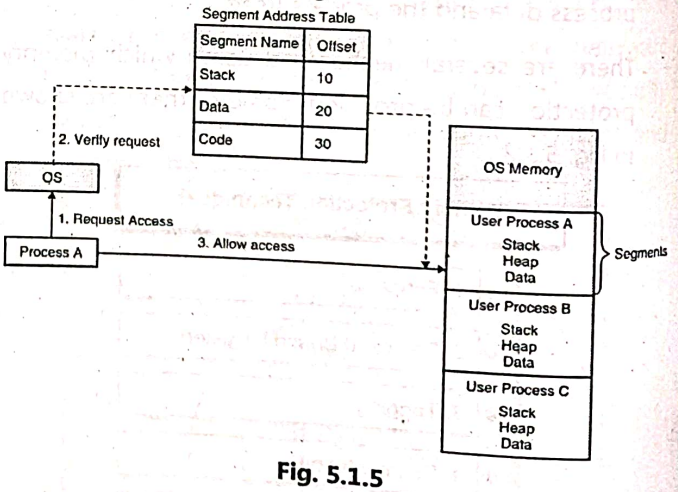


Fig. 5.1.5

- This mapping of the memory address to the segment name and the offset is stored in the Segment Address Table (SAT), as a pair. SAT is maintained by the OS for each process separately. The process does not really need to know or manage the addresses itself. The OS

refers to the SAT on every request and allows the access only if the address is within the scope of the process's SAT. If a process's SAT entries do not hold the requested address, then the access is denied.

- Segmentation provides the following security benefits.
  1. Each process is tied to its own segment.
  2. Each request is checked for permissions in SAT before access is granted.
  3. Segments can hold objects based on respective privileges and permissions.
- Segmentation could lead to memory fragmentation by not using some parts of the memory.

5. Pagination

✍ ***Definition :*** *Pagination divides the physical memory into virtual memory address pages.*

- Pagination works very similar to the table of contents in a book.
- The entire physical memory on the system is divided into equal-sized memory blocks called pages. Each page consists of several equal-sized memory units called page frames. A program is assigned a page (denoted by a page number) and page frames (denoted by page offsets). The program can then only access those page frames in the assigned pages. The OS maintains the list of (page, offset) that is assigned to each program in the Page Address Table (PAT).

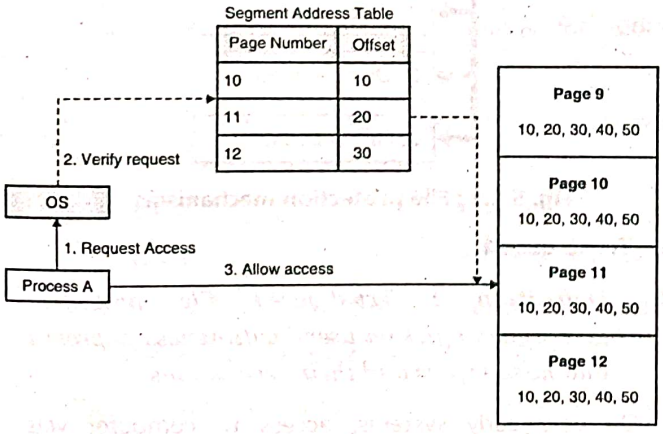


Fig. 5.1.6

- If a process tries to access a page which is not assigned to it, it results into a page fault exception very similar to the segmentation fault exception as in the case of segmentation technique.
- Since all the pages are of same size, memory fragmentation is not a problem. However, unlike segmentation, pagination does not provide a mechanism to differentiate between the types of data on a page. All data is treated equally, and the access decisions are taken based on the page numbers and page offsets.

6. Segmentation with Pagination

- Often times, because of the respective benefits that segmentation and pagination provide, they are combined.

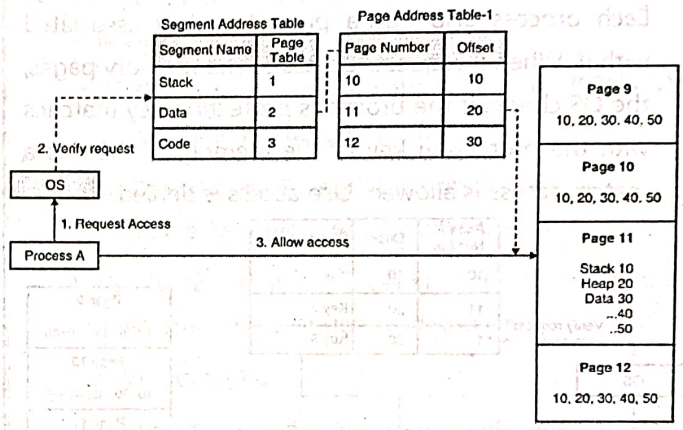


Fig. 5.1.7

7. Memory Protection Keys (MPK)

✍ ***Definition :*** *The Memory Protection Keys technique assigns a key to each memory page.*

- MPK is a CPU feature. MPK provides a mechanism for enforcing page-based protections without requiring modification of the page tables. Normal page permissions using page tables require expensive system calls and Translation Look a side Buffer (TLB) invalidations when changing permissions. MPK provides a mechanism for changing protection without requiring modification of the page tables on every permission change.
- On x86 processors, it dedicates four bits in each page table entry to a "protection key", giving sixteen

possible keys. One of sixteen key values is assigned to any given page. Each key also has two separate bits that decide whether read or read-write is allowed on the page.

(i) "Write disable" bit : If this bit is set for a given protection key then all the write attempts to the pages marked with that protection key are disabled.

(ii) "Access disable" bit : If this bit is set for a given protection key then all the read attempts to the pages marked with that protection key are disabled.

- The sixteen keys are with respect to each process. So, each process can have its own sixteen memory regions that can be protected with the keys.
- Each process also has a protection key associated with it. When it requests access to the memory pages, the OS checks if the process's protection key matches with the protection key of the memory. If there is a match, access is allowed, else access is denied.

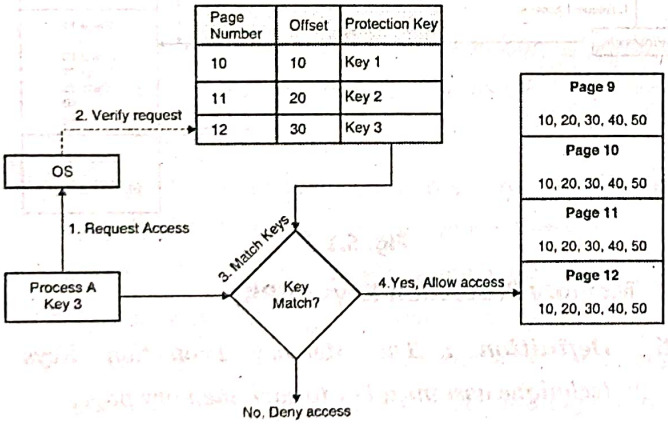


Fig. 5.1.8

5.1.1(A) Comparison between Memory Protection Mechanisms

Table 5.1.1

Mechanism Name	OS Type	Memory Allocation	Security Key	Security Benefit	Efficiency
Fence	Single-user	Sequential	None	Low	Low
Base and Bound	Multi-user	Sequential	None	Low	Low
Tagging	Multi-user	Non-Sequential	None	Medium	Very Low
Segmentation	Multi-user	Segments and offsets	None	Very High	High
Pagination	Multi-user	Pages and offsets	None	High	Very High
Protection Keys	Multi-user	Pages	Yes	High	Very High

5.1.2 File Protection Mechanisms

The core objectives of file protection mechanism are

(i) Confidentiality : Avoid disclosure of file data to unauthorised entities

(ii) Integrity : Allow only the authorised entities to make changes

(iii) Availability : Ensure the presence of file as and when needed

- You have already learnt about the several ways to control access such as DAC, MAC, RBAC and ABAC. In this section, you would specifically learn about how file protection mechanisms evolved and how are files protected by OS today. These protection mechanisms are implemented using the access control models such as DAC, MAC, RBAC and ABAC.
- File protection mechanisms can be majorly categorised as following.

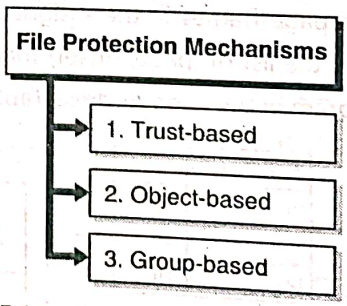


Fig. 5.1.9 : File protection mechanisms

1. Trust-based

✍ ***Definition*** *: Trust-based file protection mechanism relies on users' willingness to protect and honour files and their permissions.*

- On very early systems, access to computer was restricted to only a selected few individuals or users. Computers were not inter-connected, and you really

needed to access the computer physically to interact with them. The storage, commands, and various capabilities on the computers were also very limited.

- In those days, because there were not many users and not many capabilities, the file protection was left to the users. It was assumed that the users, who have got access to the computers and the files on them, are indeed the legitimate users and there was no further need felt to secure the files amongst the users. All files on a computer could be accessed by anyone who could use that computer. It was further assumed that a user would only be interested in the file she owns and not everything else on the system.

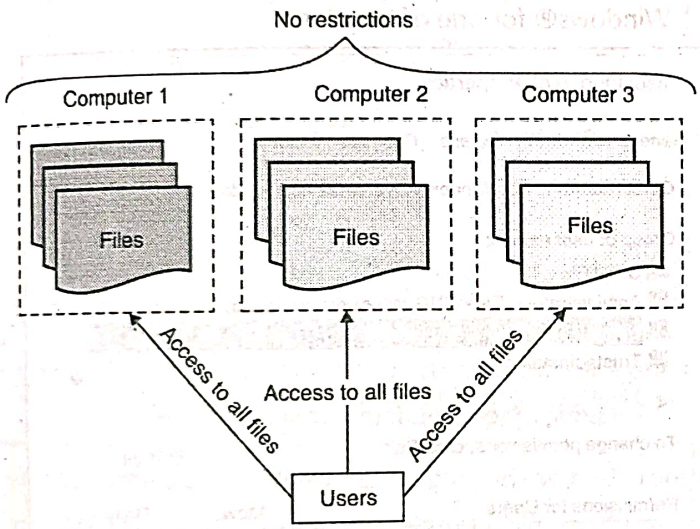


Fig. 5.1.10

- However, there used to be a capability to lock the files using password. If a file was locked, it would require the user who locked the file to unlock it before anyone else could access the file.
- This approach has several limitations for today's computing needs –
  1. There could be many users and not necessarily with equal trustworthiness.
  2. Files could have sensitive information that does require access control and cannot be left open.
  3. The systems are inter-connected and could be attacked to reveal all sensitive information.
- Trust-based approach is obsolete is no more used in modern computing.

2. Object-based

✍ ***Definition** : Object-based file protection mechanism relies on file security labels for protection.*

- Each file on the system is marked with the security properties (or permissions) using labels. A user who requires access to the file must possess equal or higher security properties (or permissions) for accessing the file. This is precisely how systems using MAC (Mandatory Access Control) policy work that you learnt earlier. MAC-based systems require files to have the labels associated with them and the access decisions for entities requiring access to the file are taken based on the label on the file and the label of the entity.
- Two examples enforcing such a mechanism are as following.
  - **(a) SELinux** : Security Enhanced Linux (SELinux) is an implementation on Linux OS.
  - **(b) Mandatory Integrity Control (MIC)** : It is an implementation on Windows OS.
- Let's dive slightly deeper in the Windows implementation to understand it better.
- On Windows OS,
  - (i) There are four integrity levels: low, medium, high, and system.
  - (ii) Standard users are treated at the medium level.
  - (iii) Privileged users, such as administrators, are treated at the high level.
  - (iv) System files and resources are treated at the system level.
  - (v) Processes you start and objects (or files) you create receive your integrity level.
  - (vi) Objects that lack an integrity label are treated at the medium level.
- This ensures that an entity can only access or modify files that have file levels matching or lower than its own level.

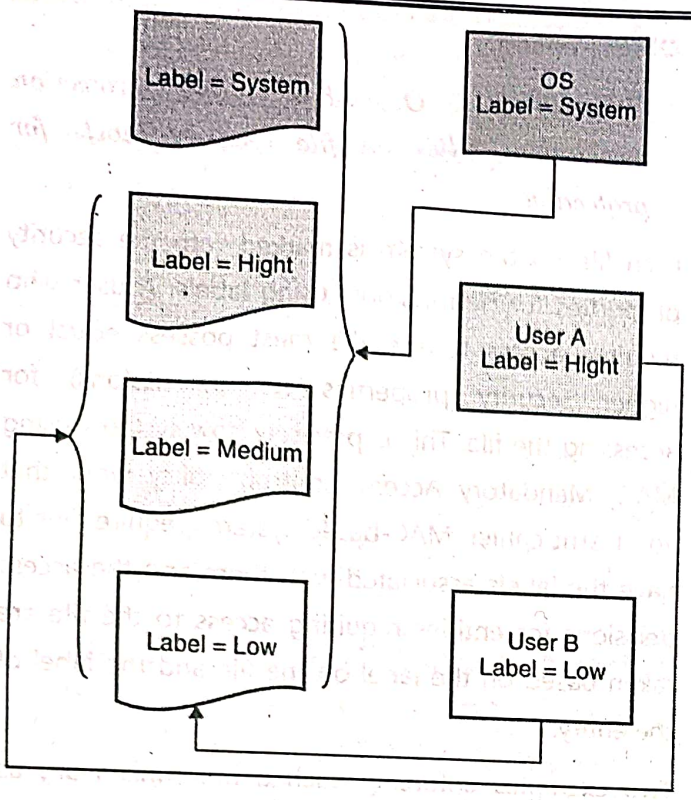


Fig. 5.1.11

- It is for this reason that the administrator, even though privileged, is restricted to modify system files.
- This method of file protection provides adequate security and protection for files. It is used in the modern operating systems and in high-security environments.

3. Group-based

✍ ***Definition*** *: Group-based file protection mechanism relies on assigning permissions to entities for file protection.*

- Typically on an operating system, various individual users and groups of users are present. In group-based protection, the users and groups present on the operating system are mapped to the files present on the system and the respective permissions are assigned to them. Group-based file protection mechanism is implemented using DAC (Discretionary Access Control) policy that you learnt earlier.
- The file permissions are typically set by the file owner during file creation or during the lifetime of the file. If the file owner does not explicitly set the file permissions, the operating system sets the default permissions as configured by the administrator.

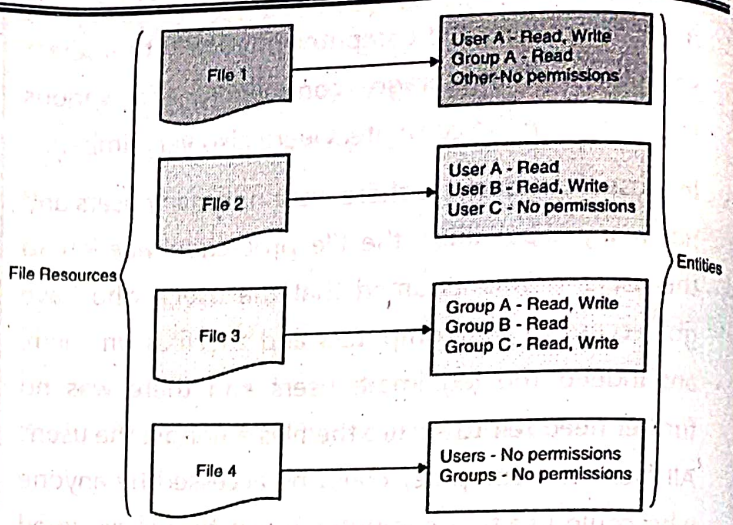
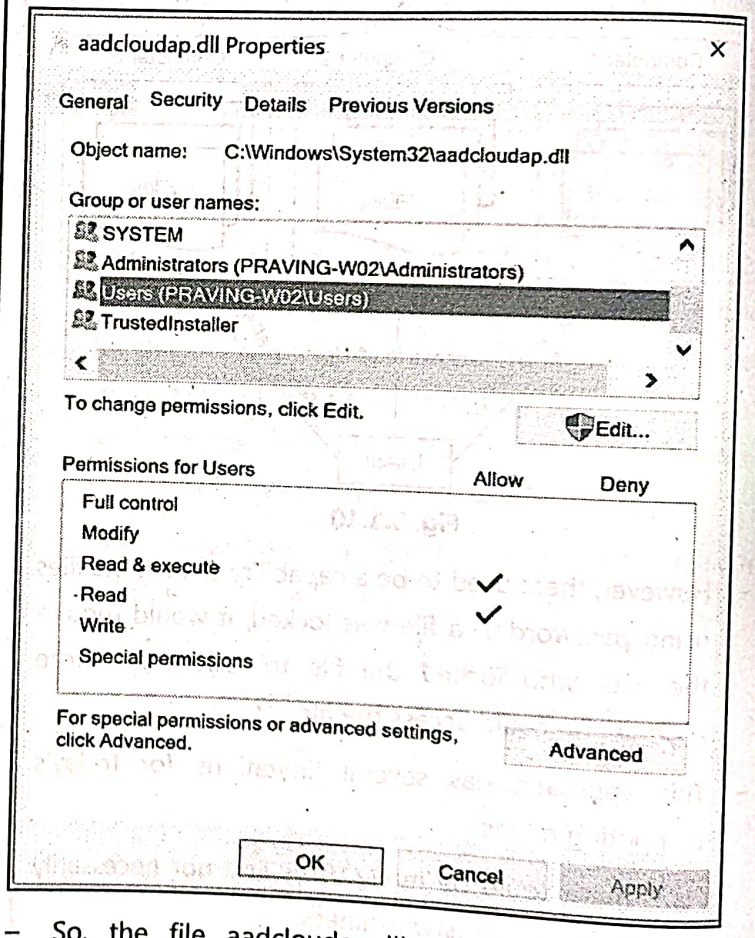
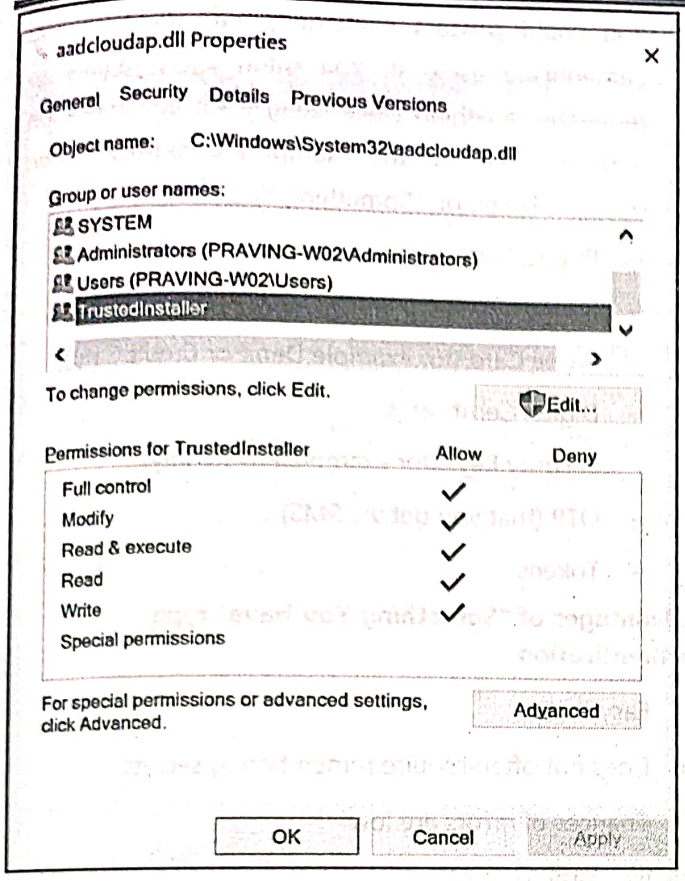


Fig. 5.1.12

- For example, here is a snapshot from Microsoft Windows® for one of the files.



- So, the file aadcloudap.dll has Read & Execute permissions for group name Users. This is typically how file permissions are assigned in the group-based protection mechanism. Several users and groups of users are assigned the respective permissions. For example, the same file aadcloudap.dll has different permissions for the `TrustedInstaller` group.



- Similarly, on a Linux system, permissions are assigned to files based on the individual users or groups.

```
ubuntu@vrni-platform:/etc$ 1s-1 passwd-rw-r—1 root root 2244 Feb 1 11 : 42 passwd ubuntu@vrni-platform : /etc$
```

- In the above snapshot, the file `/etc/passwd` is owned and group-owned by `root` and has the following permissions.
  - (i) Read, Write for the user root
  - (ii) Read only for the group root
  - (iii) Read only for everyone else (others)
- Group-based mechanism provides more flexibility than the Object-based file protection mechanism. Individual users or group of users can be assigned respective and varied permissions based on the requirements. But, as the number of users or groups increase, it becomes increasingly difficult to manage the permissions individually.

5.1.2(A) Comparison between file Protection Mechanisms

Table 5.1.2

Sr. No	Comparison Attribute	Trust-based	Object-based	Group-based
1.	Complexity	None	High	Low
2.	Protection	Low	High	Medium
3.	Modern Use	No	Yes	Yes
4.	Scalability (handle many users)	Low	High	Medium
5.	Flexibility	Low	Low	High

5.2 User Authentication

Today, you use various forms of authentication – passwords, OTP, fingerprint sensor, biometrics, etc. In Unit 1, I briefly talked about identification, authentication, authorisation, and non-repudiation. Let's elaborate on user authentication.

5.2.1 Types of Authentication Methods

Overtime, various authentication methods have evolved to address

- the ease of authentication
- make it hard to break authentication
- make authentication techniques suitable for various devices
- At a high level, authentication methods are categorised as shown in Fig. 5.2.1.

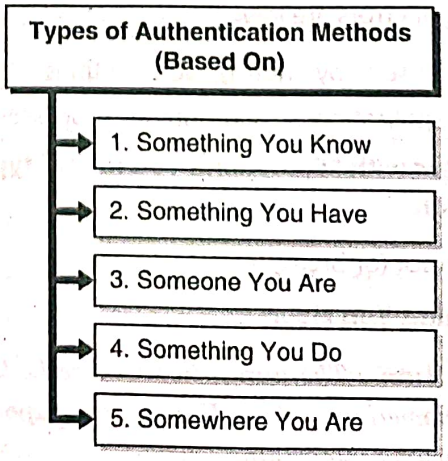


Fig. 5.2.1 : Types of authentication method (Based on)

1. Something You Know

✍ ***Definition :*** *The authentication methods based on "Something You Know" rely upon your secret knowledge about something.*

These secrets could be memorised or practiced and are often easy to recall for you. Some of the examples of authentication methods based on "Something You Know" are

- Passwords
- PIN
- Passphrases
- Secret questions
  - Mother's maiden name
  - First pet name
  - First car purchase year
  - First school name
  - City in which you were born
  - Or other secret questions whose answers would be likely known just to you
- Lock key combination

Advantages of "Something You Know" type authentication

(i) Easy to implement by developers in the product (OS, Applications or Websites)

(ii) Easy to recall for the user

(iii) Easy to authenticate for the user

(iv) Easy to change for the user

(v) Chances of errors are low

(vi) Can be used by non-person entities (hardware devices, applications, OS, programs, processes, etc.)

Disadvantages with "Something You Know" type authentication

(i) Easy to crack (or break)

2. Something You Have

✍ ***Definition :*** *The authentication methods based on "Something You Have" rely upon your possession of something.*

You could possess something that could let you authenticate using it. You often don't require to remember anything while using it for authentication purpose. Some of the examples of authentication methods based on "Something You Have" are

- Physical keys
- Badge
- Swipe Card (for example Debit or Credit Card)
- Digital Certificates
- Security Keys (for example Private Key)
- OTP (that you get via SMS)
- Tokens

Advantages of "Something You Have" type authentication

(i) Easy to use

(ii) Does not often require remembering secrets

(iii) Chances of errors are low

(iv) Not easy to crack

(v) Can be used by non-person entities (hardware devices, applications, OS, programs, processes, etc.)

Disadvantages of "Something You Have" type authentication

(i) Difficult to change

(ii) Possibility of loss or theft

(iii) Requires distribution methods (provisioning) to reach to the user securely

3. Someone You Are

✍ ***Definition :*** *The authentication methods based on "Someone You Are" rely upon your physical characteristics.*

Your body has several physical characteristics that can be used to uniquely identify and authenticate you. These characteristics do not much change over time (as you age) and can serve authentication purpose for near lifetime. These characteristics are called **Static Biometrics.**

✍ ***Definition :*** *Static Biometrics are physical characteristics that can be used for authentication.*

Generally speaking,

✍ ***Definition :*** *The measurement and analysis of unique physical or behavioural characteristics is called biometrics.*

However, there can be scenarios where re-provisioning (re-calibrating) your biometric details might be required. For example, if you use a particular finger for fingerprint and you're that particular finger is damaged permanently, you might have to choose another finger or another type of biometrics for authentication. Some of the examples of authentication methods based on "Someone You Are" are

- Fingerprint
- Palm Scan
- Hand Geometry
- Retina Scan
- Iris Scan
- Facial recognition

Advantages of "Someone You Are" type authentication

(i) Easy to use

(ii) Does not often require remembering secrets

(iii) Difficult to crack

Disadvantages of "Someone You Are" type authentication

(i) Difficult to implement correct

(ii) Chances of errors are high (recall you trying fingerprints several times at Aadhar enrolment centre?)

(iii) Difficult to change (requires physical presence for re-provisioning)

(iv) Cannot be used by non-person entities (hardware devices, applications, OS, programs, processes, etc.)

4. Something You Do

✍ ***Definition :*** *The authentication methods based on "Something You Do" rely upon your way of performing a given task.*

This authentication method also collects biometric patterns but while performing a given task. Unlike static biometric patterns, these are dynamic biometric patterns which are used for authentication purpose.

✍ ***Definition :*** *Dynamic Biometrics are job performing characteristics that can be used for authentication while an individual is performing a given task.*

Some of the examples of authentication methods based on "Something You Do" are

- Voice print (or pattern)
- Keystroke Dynamics (how hard you press the keys and how fast)
- Handwriting characteristics (remember old movies where handwriting matches were done?)

Advantages of "Something You Do" type authentication

(i) Easy to use

(ii) Does not often require remembering secrets

Disadvantages of "Something You Do" type authentication

(i) Difficult to implement correct

(ii) Not too difficult to crack (consider replaying a recorded voice)

(iii) Chances of errors are high (have you tried voice to text yet?)

(iv) Difficult to change (requires physical presence for re-provisioning)

(v) Cannot be used by non-person entities (hardware devices, applications, OS, programs, processes, etc.)

5. Somewhere You Are

✍ ***Definition :*** *The authentication methods based on "Somewhere You Are" rely upon your physical location.*

Increasingly, the devices and systems have location awareness. Your mobile phone or laptops (via network connectivity) precisely know where you are located at a particular moment. "Somewhere You Are" uses the location information for authentication.

For example, Google Smart Lock for Android allows you to set Trusted Locations. Say, if you are at home, your phone may not require you to unlock it (via PIN, pattern, password or fingerprint) before use.

Set your Android device to automatically unlock

You can keep your Android phone or tablet unlocked in some situations, like when your phone is in your pocket or you're near home. When you use Smart Lock, you won't need to unlock with your PIN, pattern, or password. The features you can use depend on your device.

If you want to change your screen lock, learn more about screen lock settings.

Note: Some of these steps work only on Android 9 and up. Learn how to check your Android version.

Turn on automatic unlock

1. Make sure you have a screen lock. Learn how to set a screen lock.
2. Open your device's Settings app.
3. Tap **Security & location** > **Smart Lock**.
4. Enter your PIN, pattern, or password.
5. Pick an option and follow the on-screen steps.

After setup, when you turn on your screen, you'll see a pulsing circle at the bottom around the Lock.

Important: When you don't use your device for 4 hours, and after it restarts, you'll need to unlock it.

Turn off automatic unlock

1. Open your device's Settings app.
2. Tap **Security & location** > **Smart Lock**.
3. Enter your PIN, pattern, or password.
4. Turn off **On-body detection** and remove all trusted devices, trusted places, trusted faces, and Voice Match voices.
5. Optional: If you want to turn off your screen lock, learn how to change your screen lock.

"Somewhere You Are" is also quite widely used in corporate IT. In many environments, if you are on an office network (LAN or Wi-Fi), you can login using only a password, but if you are out of the office you must use VPN or an additional mechanism for authentication.

Advantages of "Somewhere You Are" type authentication :

(i) Easy to use

(ii) Does not often require remembering secrets

(iii) Can be used for authenticating non-person entities (mobile, laptops or other location aware devices)

Disadvantages of "Somewhere You Are" type authentication :

(i) Not too difficult to crack (consider theft and someone carrying the device to a trusted location)

(ii) Cannot be used for authenticating individuals

(iii) Requires network connectivity for location awareness

5.2.1(A) Comparison between the Authentication Types

Type of authentication	Ease of use	Ease of Change	Ease of implementation	Error rate	Support non-person entities
Something You Know	High	High	High	Low	Yes
Something You Have	High	Low	Medium	Low	Yes
Someone You Are	High	Low	Low	High	No
Something You Do	Medium	Low	Low	High	No
Somewhere You are	High	High	Medium	Medium	Yes

5.2.2 Factors of Authentication

Each type of authentication that you learnt in this section is called a factor of authentication. Based on the number of factors you choose for effectively carrying out and completing the authentication process, you have three types of authentication scenarios as shown in Fig. 5.2.2.

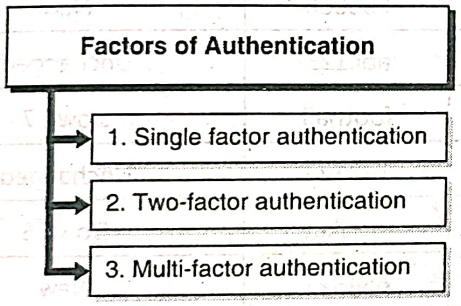


Fig. 5.2.2 : Factors of authentication

1. Single factor authentication

✍ ***Definition** : Single factor authentication requires only ONE of the types of authentication for successfully carrying out the authentication process.*

For example, you could just use password or token. This is most widely used. It is often treated as a weak form of authentication.

2. Two-factor authentication

✍ ***Definition** : Two-factor authentication requires you to use any TWO types of authentication, one after another, for successfully carrying out the authentication process.*

It is considered as a strong form of authentication. For example, for your online transactions, you are first required to give the account password and then you receive an OTP. You are required to put the correct OTP for successfully authenticating your account details and carrying out and completing your transaction. Another example could be ATM. You must possess you Debit Card (Something You Have) and put in the right PIN (Something You Know) for withdrawing money.

3. Multi-factor authentication

✍ ***Definition** : Multi-factor authentication requires you to use MORE than two types of authentication.*

It is often used in a high security environment. For example, you may be first required to give your fingerprint, after which you can access an application where you are required to provide username and password. For carrying out a transaction on that application, you might require an OTP.

Note : If two authentication techniques from the same type of authentication are used, it is not considered two-factor authentication. It would be considered a single factor authentication. For example, you cannot consider consecutive requirement of two passwords or a password and a PIN to be two-factor authentication. Two-factor authentication essentially requires two different ways to authenticate.

5.2.3 Password based Authentication

Definition : *A password is a protected sequence of characters that is used to authenticate an entity and provide access to a protected system.*

Passwords are the most widely used form of authentication. You can use it on computers, mobile devices, web portals, applications and nearly everything else that supports any form of user interface and requires authentication before use. Passwords fall into "Something You Know" type of authentication that you learnt earlier. The use of passwords requires effective password management that is strong enough to keep the passwords protected.

5.2.3(A) Choosing a Password

- Passwords, even though are the easiest to use amongst other types of authentication techniques, are also weakest form of authentication. Why? Because, users usually choose weak passwords which are easily breakable or guessable.
- Passwords can either be user chosen or be automatically generated by a trusted system.

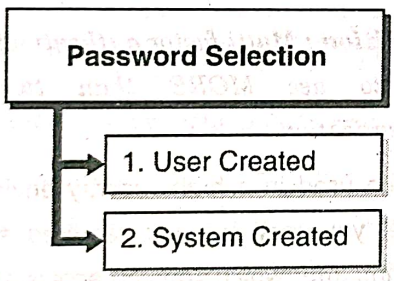


Fig. 5.2.3 : Password selection

1. User Created

- A user can typically choose her own password when signing up on a system. The problem here is that the users typically choose something that is extremely easy to guess.
- Some of the common and worst password choices of 2018 are shown in the Table 5.2.1.
- **Source:** Splash Data's Top 100 Worst Passwords of 2018

Table 5.2.1

Rank	Password	Change from 2017
1	123456	Unchanged
2	password	Unchanged
3	123456789	Up 3
4	12345678	Down 1
5	12345	Unchanged
6	111111	New
7	1234567	Up 1
8	sunshine	New
9	qwerty	Down 5
10	iloveyou	Unchanged
11	princess	New
12	admin	Down 1
13	welcome	Down 1
14	666666	New
15	abc123	Unchanged
16	football	Down 7
17	123123	Unchanged
18	monkey	Down 5
19	654321	New
20	!@#$%^&*	New
21	charlie	New
22	aa123456	New
23	donald	New
24	password1	New
25	qwerty123	New

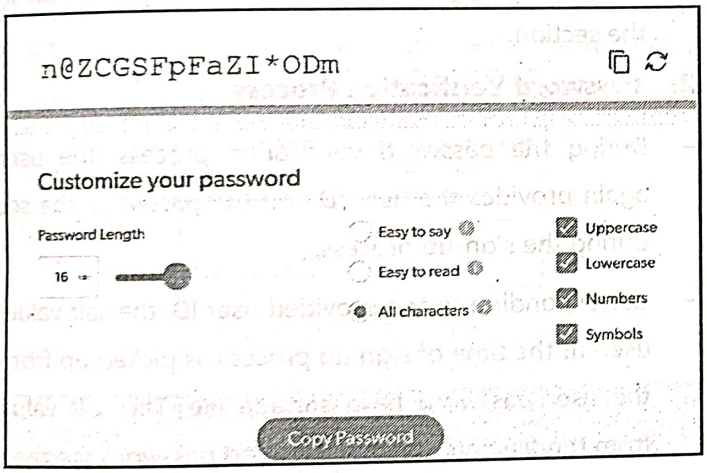
Is your password on the list?

Let's pick the first password from this list (123456) and find out how much time it would take a computer to break this password. https://howsecureismypassword.net/ provides a good estimate of password cracking times with respect to the provided password.

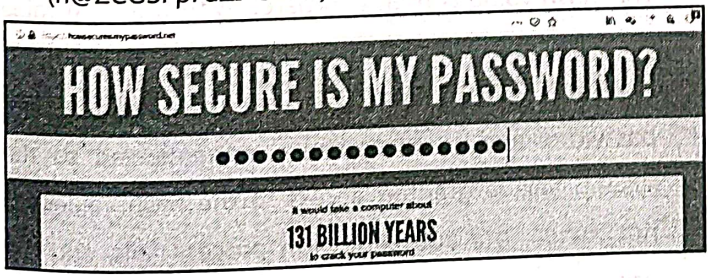
- The provided password (123456) can be instantly broken. Would you choose that as your password still?

2. System Created

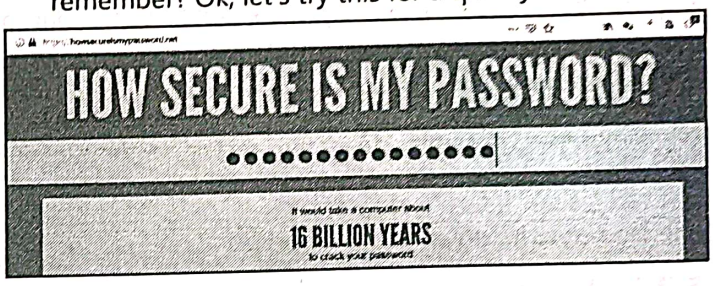
- Quite a few systems today offer passwords that are
  (i) Randomly generated (ii) Difficult to crack and
  (iii) Easy to remember
- Such systems ensure that the important quality parameters for a good and strong password are matched in the suggested passwords.
- Here is a snapshot of LastPass password generator tool.



- It allows you to choose length, mix of characters and creates a random and strong password. Let's find the strength of the password it generated (n@ZCGSFpFaZI*ODm).



- It would take 131 billion years for a computer to crack this password - n@ZCGSFpFaZI*ODm. Wow, now that is something called as a strong password! I would suggest choosing that if you are protecting something worthwhile!
- But, do you say that the password (n@ZCGSFpFaZI*ODm) is hard to remember? Ok, I hear you.
- Let me give you a tip. How about choosing an Indian festival as a password but slightly altering the characters? Let's take "Ganesh Chaturthi". I will replace "a" with "@"and "i" with "1". So, "Ganesh Chaturthi" would be "G@neshCh@turth1". Easy to remember? Ok, let's try this for a quality test.



- Not bad, right? I see you smiling. You will learn how to create good password very soon. Just read on!
- By the way, don't choose "G@neshCh@turth1" as your password now because I and many other readers of this book would then know your password. Create something of your own and don't tell it to anyone.

a. Password Selection Criteria (Quality Guidelines)

Here are some general guidelines when choosing passwords.

(i) Choose at least 12 characters in your password. The lengthier the password, the harder it is to crack it.

(ii) Use mix of characters
  (a) At least one digit
  (b) At least one uppercase letter
  (c) At least one special character such as (@,#, $, %, ^, &, *, (,), etc.)

(iii) Do not use the same password for all your accounts.

(iv) Do not use easy to guess passwords. Do not use your name, date of birth, school name etc. as your password.

(v) Check your password quality before using it.

(vi) Do not use words found in the dictionaries. For example, avoid creating a password such as "Apple".

b. General Password Usage Guidelines

Here are some general guidelines to follow when working with passwords.

(i) Do not send your password in clear text. Avoid entering them on sites that do not use https.

(ii) Do not share your password with anyone in your family and friends.

(iii) Change your passwords periodically to avoid overuse.

(iv) Do not tell your password to anyone over phone or email howsoever the conversation may sound legitimate.

(v) Wherever possible, use two-factor authentication with password.

(vi) Do not write down your password!

5.2.3(B) Storing Passwords on System

When you enter your password on a Windows® or a Linux machine, how does the system know that you entered the right password? Does it compare your provided password, character by character, to the one previously stored on the system that you chose during sign up? No, it does not store your password in cleartext at anywhere on the system. The passwords are adequately protected using hashes.

1. Sign-up Process

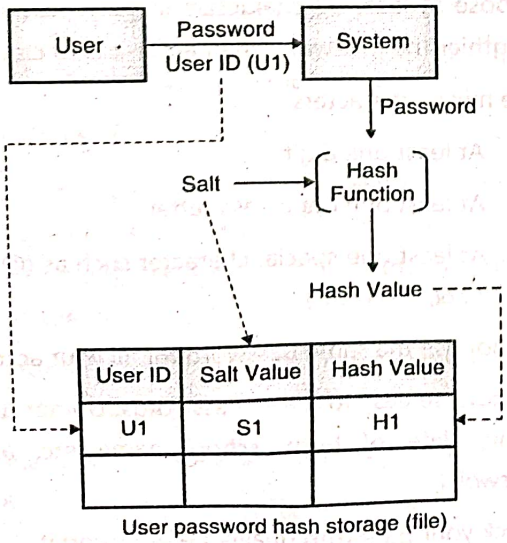


Fig. 5.2.4

- During the sign-up process, the user creates an ID and chooses a password that she would like to use when authenticating to the system.
- The system chooses a random salt value (nonce or any other random value) and passes that along with the user provided password to a hashing function. The hashing function computes the hash value. Then, the hash value and the corresponding salt value with the user ID is saved on the system in the form of a file. The system does not store password in clear text at anywhere on the system.
- The purpose of the salt value is to bring randomness to the hashing process. So, if any two users provide the same password, even then, the entries in the password hash table file would not be the same. The random salt value also serves a purpose of defending the passwords from attack as you would see later in the section.

2. Password Verification Process

- During the password verification process, the user again provides the user ID and the password she set during the sign-up process.
- Corresponding to the provided user ID, the salt value, used at the time of sign up process, is picked up from the user password hash storage file. The salt value from the file and the user provided password are then fed to the same hashing function. The hashing function computes the hash value with the provided inputs.
- This hash value is then compared with the hash value stored in the user password hash storage file. If the two hash values match, the user provided the right password and is successfully authenticated to the system and is granted access. If the hash values do not match, the authentication process fails, and the user is denied access to the system. The following schematic explains the verification process.

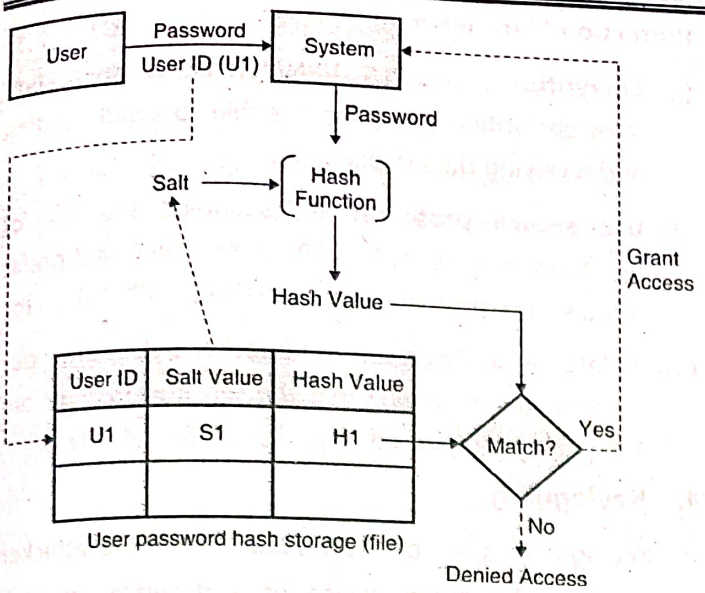


Fig. 5.2.5

5.2.3(C) Attacks on Password Based Authentication

As you know, password is the weakest form of authentication. There have been several mechanisms to attack on it as shown in Fig. 5.2.6. Let's learn about them.

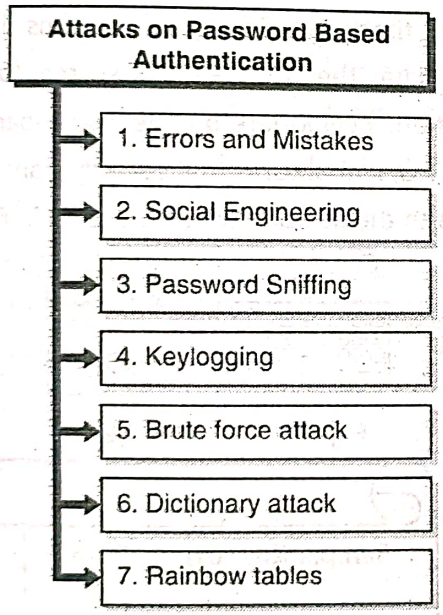


Fig. 5.2.6 : Attacks on password based authentication

1. Errors and Mistakes

This is probably the most common reason due to which cracking passwords is not that hard. These are common pitfalls against the password creation and general password usage guidelines that you learnt earlier. The common mistakes are

(i) Choosing a weak password that does not align with the password selection guidelines

(ii) Writing down passwords

(iii) Sharing them with others

(iv) Transferring or storing them in cleartext

(v) Using the same password for multiple accounts

(vi) Choosing common passwords (something as in the top 25 worst passwords list that you saw earlier)

– Such errors and mistakes make it quite easy to crack password-based authentication and gain unauthorised access to the system.

Protection Mechanism

(i) Avoid common pitfalls and adhere to the password selection and password usage guidelines.

(ii) Use two-factor authentication to ensure that the authentication is not processed only based on the provided password. Another form of authentication would be required before any access is granted to the system.

2. Social Engineering

As you learnt earlier, social engineering attacks can be carried out on the users. Users can be tricked to

(i) Reveal the current password (say by posing as your manager needing access to something very quickly)

(ii) Reset the password (say by posing as someone calling from IT department and asking for quick reset)

(iii) Ask to install software that could read user passwords or send password hash storage file to the attacker

(iv) Click on links that could reset or reveal the current password

(v) Or carry out anything else that helps the attacker to perform attacks on the password-based authentication

Protection Mechanism

(i) Periodic user education and training on social engineering and cybersecurity.

(ii) Users should be repeatedly made aware of social engineering attacks and how not to get trapped.

3. Password Sniffing

Passwords that are transferred without encryption (in cleartext) are prone to sniffing.

✍ ***Definition** : Packet Sniffing is the act of intercepting (capturing) of network traffic and logging it for further analysis*

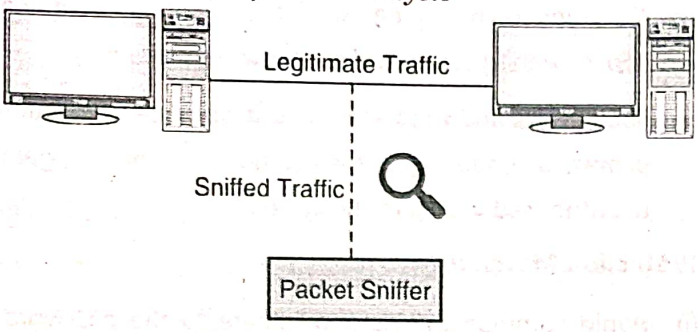
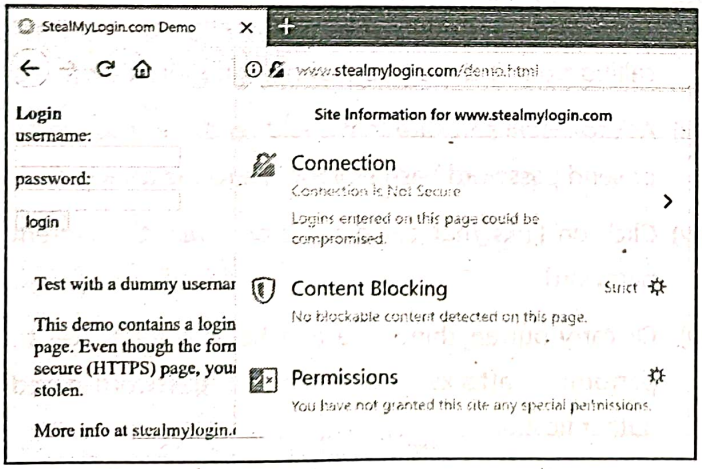


Fig. 5.2.7 : Packet sniffing

- Packet sniffing is carried out using either software programs or hardware devices. These are called packet analysers or just **packet sniffers**. Wireshark is one of the most widely used Packet Sniffing program.
- Hence, as you see, as a best practice, never send or store your passwords in cleartext. These days browsers give you a warning when you are entering a password in a HTTP (and not HTTPS) site. The HTTP protocol sends information to the webserver in cleartext. You should not ignore such browser warnings. Avoid entering sensitive information such as password on such sites.



- You cannot, to a great extent, avoid network sniffing attempts since it is external to the network and mostly beyond your control. For example, how do you protect someone with Wi-Fi analyzing device to capture your Wi-Fi packets? However, some of the common protection mechanisms to reduce the impact from sniffing are as follows.

Protection Mechanism

(i) **Encryption :** Use TLS, VPN tunnels or application level encryption wherever possible to avoid sending and receiving data in plaintext.

(ii) **Use secure protocols :** Discourage the use of protocols such as HTTP, FTP, TELNET, RPC and prefer using secure protocols such as HTTPS, FTPS, SSH, etc.

(iii) **Isolate and Segment networks :** Design your network in such a way that the sensitive systems are adequately segmented from rest of the network.

4. Keylogging

- Keylogging is an old technique where the attacker installs a hardware device or a malware on your system that captures all your keystrokes.

✍ ***Definition** : Keyloggers or simply keystroke loggers are malware programs that capture your keystrokes on the keyboard.*

- The simple idea behind keylogging is that any user would definitely type sensitive information at some point in time. By identifying patterns in the key logged data, the attacker can search for sensitive information. Keyloggers are generally part of other virus or Trojan malwares that secretly transmit logged key data to the attacker which is shown in Fig. 5.2.8.

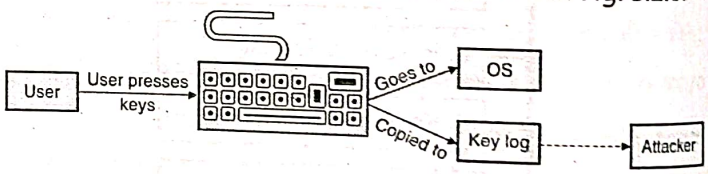


Fig. 5.2.8 : Key logger

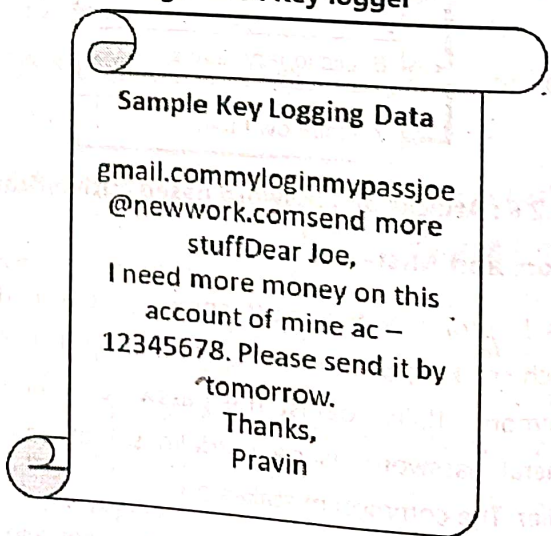


Fig. 5.2.9

- Fig. 5.2.9 shows a sample keylogging data.
- Can you identify some patterns and figure out some of my sensitive information? What If I told you that the data belongs to my activity of sending an email via my Gmail account to a friend? Can you figure out the sensitive information?
- Let me help you. Assume the following sequence of activities.
  (i) I opened my favorite browser Mozilla and typed "gmail.com".
  (ii) The Gmail login page appeared and then I typed my username "mylogin" and my password "mypass".
  (iii) Then, I clicked on *compose* and typed the email address of my friend "joe@newwork.com".
  (iv) Then, I typed my email subject "send more stuff".
  (v) Then, I composed my email and hit the *send* button.
- Does it sound simple but scary? That's precisely what keylogging is. All keys are captured as you press them. Some modern and advanced keyloggers also capture mouse clicks (x and y coordinates of the screen or pixel position information on the screen) along with screenshots.

Protection Mechanism

(i) Regularly use anti-malware programs to scan your systems.

(ii) Set multi-factor authentication to avoid account hijack even if your password is compromised.

(iii) Use virtual keyboard specially when dealing with financial related logins.

(iv) Regularly check your installed program inventory to identify any potential malware.

(v) Regularly check the services running on your system to identify any potential malware service.

(vi) Physically inspect your system wires and attachments to ensure that there are no unwanted hardware plugins.

5. Brute Force Attack

✍ ***Definition :*** *Bruce force attack is performed using tools that cycle through various combinations of characters until a successful combination of characters is found.*

- These tools continuously send various character combinations in the hope of finding a successful combination. The time it takes to crack passwords this way is usually large.
- Some of the common tools to carry out the brute force attack are :
  (i) Brutus (ii) RainbowCrack
  (iii) Cain and Abel (iv) Wfuzz
  (v) John the ripper (vi) Medusa
  (vii) Aircrack-ng

Protection Mechanism

(i) Use lengthy and complex passwords. The more complex the password, the longer it takes to brute force it.

(ii) Introduce a time delay after a certain number of failed logins. For example, the system can lock the account after 3 failed attempts and requires administrator to unlock the account or wait for a period of 1-24 hours before automatically unlocking the account.

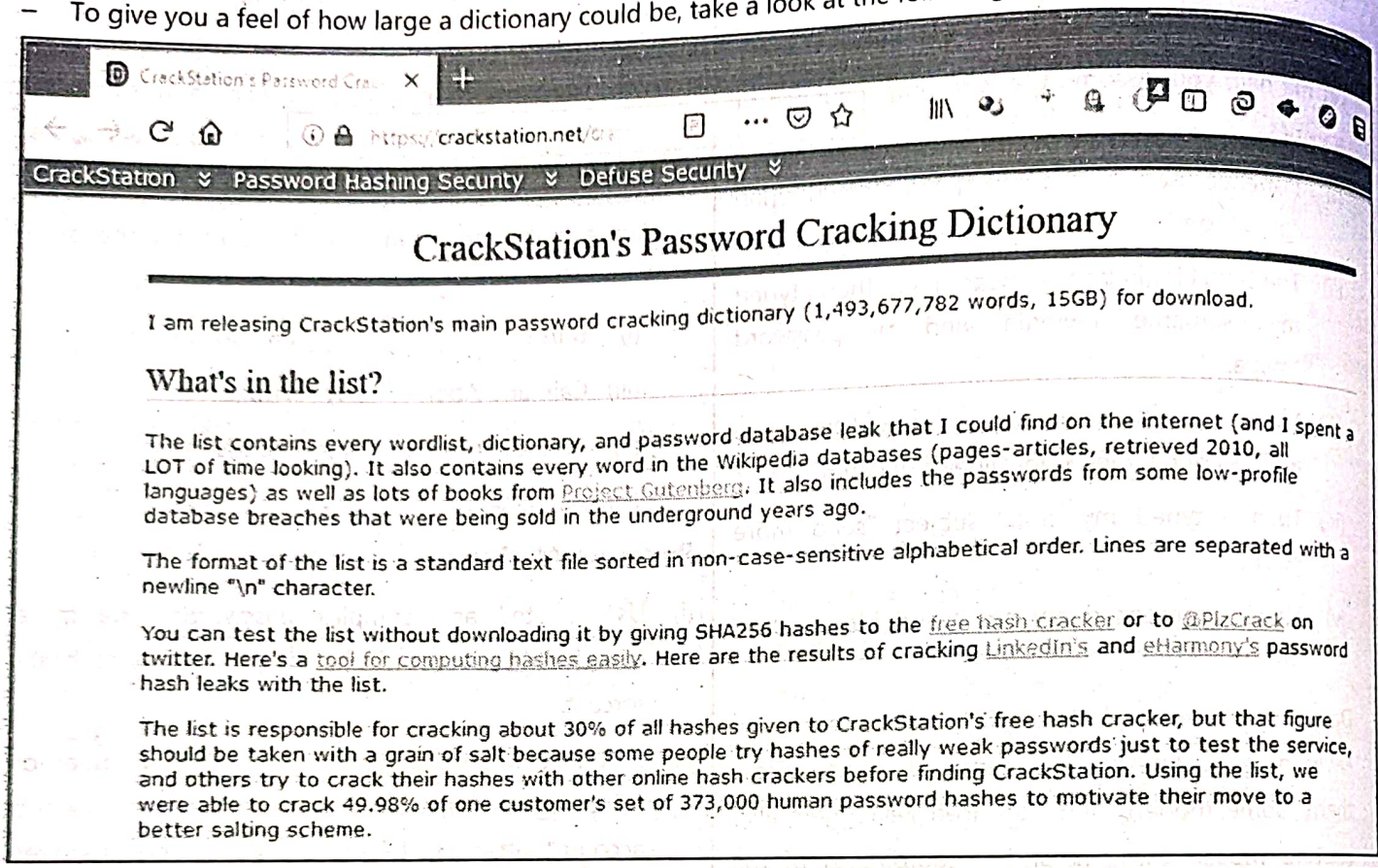
(iii) Send login notification. Whenever your account is attempted to log in, you could be notified via SMS or email. You could then take corrective actions if the login activity is not found to be legitimate.

(iv) Protect the password hash storage files from any unauthorised access or download. The attacks can be carried out on the files thus bypassing all forms of controls.

6. Dictionary Attack

✍ ***Definition :*** *Dictionary attack is a variation of brute force attack where instead of trying random character combinations, dictionary words and their variations are tried out for cracking the password.*

- Dictionaries could be used from the user's natively spoken language or the multiple languages that a user may know. Apart from the regular dictionaries, there are other dictionaries which are collections from the hacker community. When a large user account database is stolen, such information is available with the hacker community that use the information to form a listing of new passwords that people might be using.
- To give you a feel of how large a dictionary could be, take a look at the following snapshot.



- Imagine a text file of 15 GB!!! Impressive, isn't it?
- Dictionary attack might take lesser time than the actual brute force attack since not all combinations are tried. Also, if the user has not chosen a dictionary-based password, then the dictionary attack may not be able to crack it.

Protection Mechanism

(i) Follow password selection guidelines and choose a complex and non-dictionary password.

(ii) Introduce a time delay after a certain number of failed logins. For example, the system can lock the account after 3 failed attempts and requires administrator to unlock the account or wait for a period of 1-24 hours before automatically unlocking the account.

(iii) Send login notification. Whenever your account is attempted to log in, you could be notified via SMS or email. You could then take corrective actions if the login activity is not found to be legitimate.

(iv) Protect the password hash storage files from any unauthorised access or download. The attacks can be carried out on the files thus bypassing all forms of controls.

7. Rainbow tables

✍ ***Definition** : Rainbow tables consist of passwords in the hash format.*

- Rainbow tables are minor variations of dictionaries used in the dictionary attacks. Unlike dictionary attacks that use the pre-defined character combinations, rainbow tables contain pre-computed hash values for every possible character combination

in those dictionaries. That way, you need not waste system resources and time in computing the hash values (again and again) and then comparing with the hash value on the system (or the stolen user password hash storage file). The Table 5.2.2 shows a simplistic rainbow table.

Table 5.2.2

Character Combination	SHA-1 Hash
NewPassword	735b535bd8148743e2e19b08c7db8ea60142a0b3
MyPassword	daa1f31819ed4928fd00e986e6bda6dab6b177dc
Iamhappy	b0177566098815e0555830d7a68d2352614fe9bc
Apple@123	d88f1b3fb247e4ad3f6b821e4da902d1c91f0864

- Your problem is simplified. All you need to find then is any hash value in the rainbow table that could match with the system's hash value. When such a match is found, you can find the corresponding character combination from the rainbow table. This character combination is the cracked password.

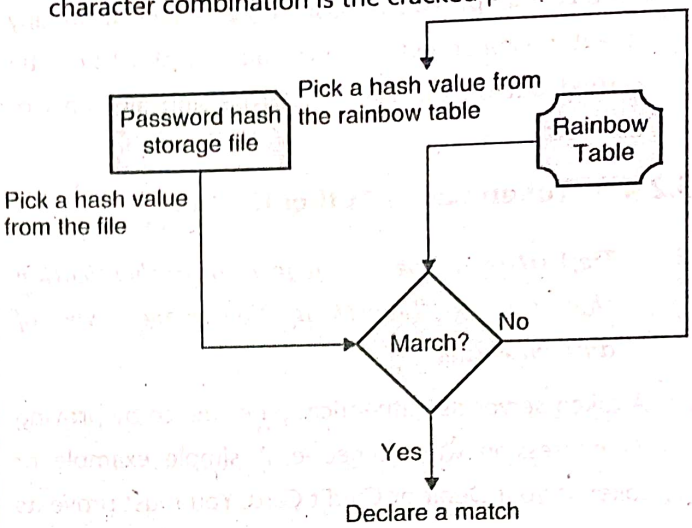
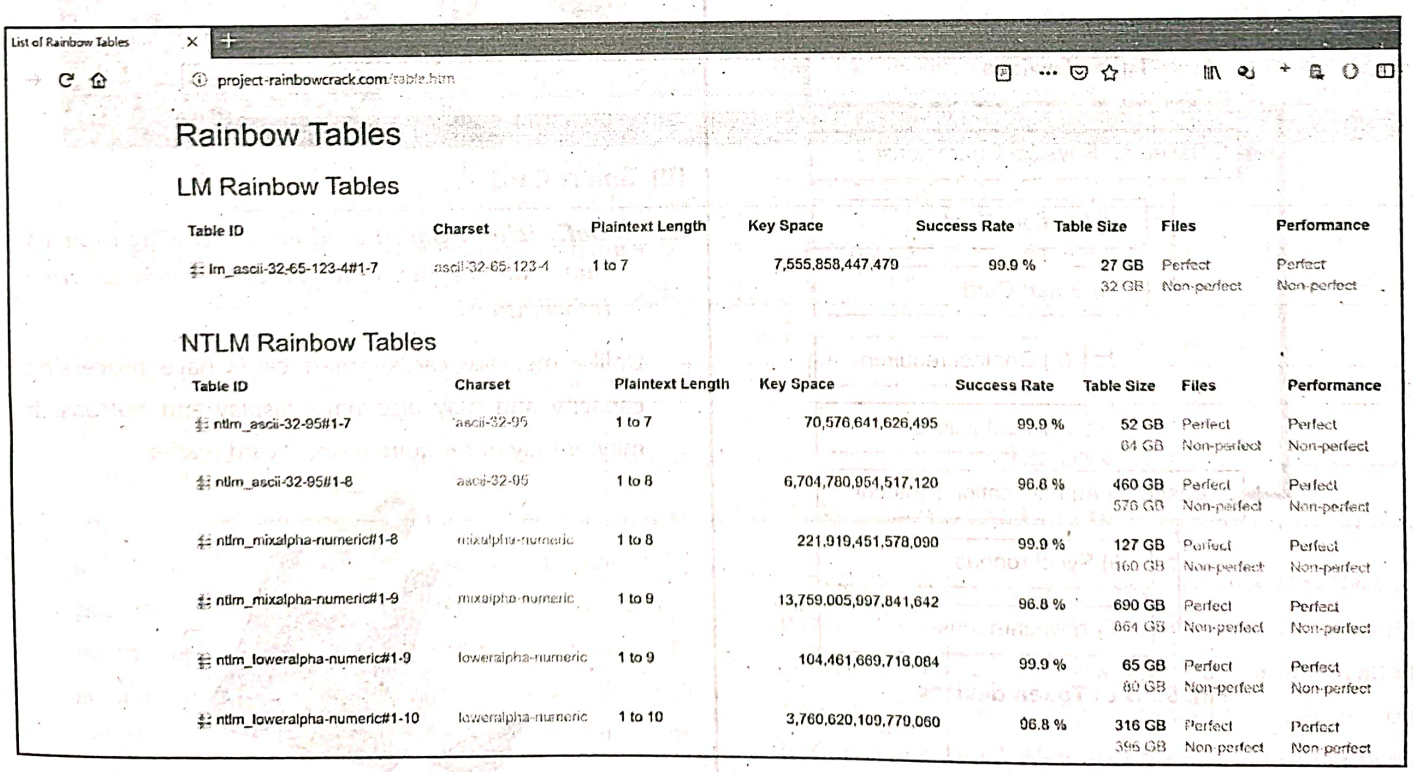


Fig. 5.2.10

- Rainbow tables are much faster when compared with the brute force or dictionary attacks. RainbowCrack is one of the tools that uses rainbow tables to crack passwords. The rainbow tables are also huge in size as they contain several hash values for dictionary words.



Protection Mechanism

(i) Follow password selection guidelines and choose a complex and non-dictionary password.

(ii) Protect the password hash storage files from any unauthorised access or download. The attacks can be carried out on the files thus bypassing all forms of controls.

5.2.4 Token based Authentication

✍ ***Definition :*** *Token is a form of authentication that utilizes "Something You Have" type of authentication.*

- A token serves as authentication evidence by proving its possession with someone. A simple example of token is your Debit or Credit Card. You must prove its possession to use it. A physical transaction may require a swipe of the card to prove possession whereas an online transaction may require various details about the card such as the card number, expiry date, CVV number and card holder's name.
- There are various types of token devices used in different ways to provide authentication as shown in Fig. 5.2.11.

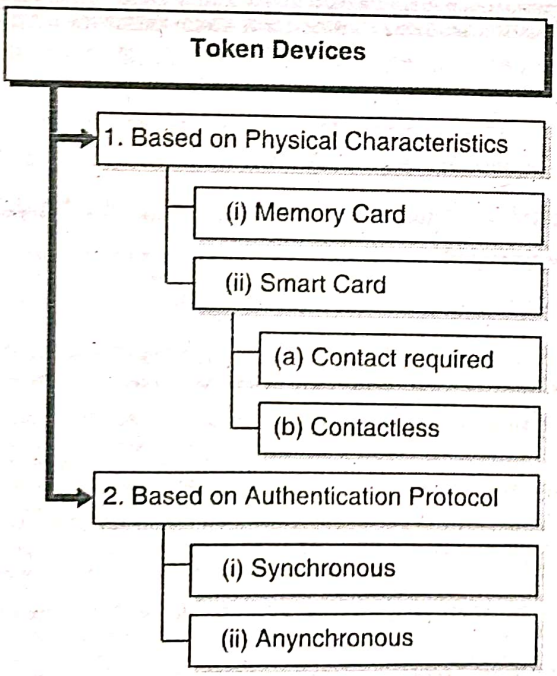


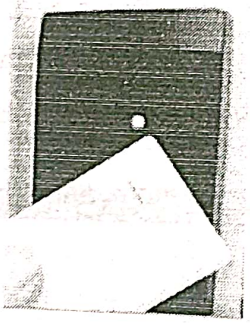
Fig. 5.2.11 : Token devices

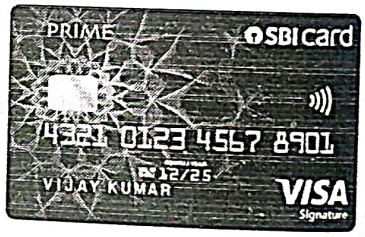
1. Based on Physical Characteristics

(I) Memory Card

✍ ***Definition :*** *Memory cards physically contain the required authentication information.*

- It usually has a magnetic stripe (or a chip) and the authentication information is contained in this stripe (or chip). You require a card reader using which you can swipe the card (very much like your Debit Card) and authenticate yourself. Memory cards themselves do not possess any form of processing power.
- Sometimes, the use of memory cards also requires a password or a PIN to validate that it is actually in the possession of the right individual. This is done to avoid circumstances where stolen cards can be used just by their possession without the knowledge of the card owner. The use of a PIN or a password along with the card possession, makes authentication as two-factor authentication.





(II) Smart Card

✍ ***Definition :*** *Smart card has processing capacity that can be used to provide authentication information.*

- Unlike memory cards, smart cards have processing capacity and may also have display and buttons. It may or may not require a smart card reader.

- There are two types of smart cards.

  (a) Contact required : The smart card needs to be inserted into the smart card reader for authentication.

  (b) Contactless : The smart card just needs to be in the close proximity of the reading device.

2. Based on Authentication Protocol

(I) Synchronous Tokens

✍ ***Definition*** *: A synchronous token device synchronizes (works tightly coupled) with the authentication service for providing authentication information.*

- The synchronization could be based on time or a counter. RSA SecurID is an example of time-based synchronous token.
- If the synchronization is time-based, the token device and the authentication service must hold the same time within their internal clocks. The time value on the token device and a secret key (pre-loaded in the token device and the authentication server also has a copy of it) are used to create the one-time password, which is displayed to the user. The user enters this value and a user ID into the computer, which then passes them to the server running the authentication service. The authentication service compares it to the value it expected. If the two match, the user is authenticated and allowed to use the computer and resources.
- If the token device and authentication service use counter-based synchronization, the user will need to initiate the creation of the one-time password by pushing a button on the token device. This causes the token device and the authentication service to advance to the next authentication value. The user enters this resulting value along with a user ID to be authenticated.

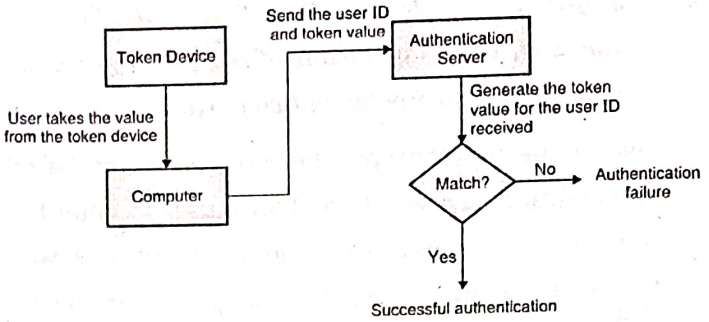


Fig. 5.2.12

(II) Asynchronous Tokens

✍ ***Definition*** *: Asynchronous token device uses challenge-response mechanism for providing authentication information.*

- Asynchronous token is not time dependent. It uses a challenge-response mechanism (very much like handshake protocols) for providing authentication information.
- The authentication server sends the user a challenge which is usually a random value. The user needs to enter this value into the token device. The token device encrypts the value entered and returns a value that the user uses as a one-time password. The user sends this value along with the user ID to the authentication server. If the authentication server can decrypt the value and gets the same challenge value that it sent earlier, the user is authenticated.

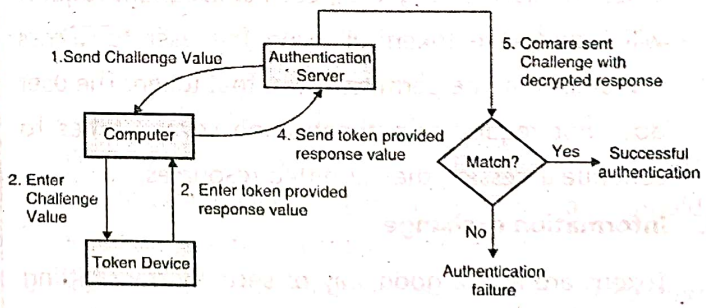


Fig. 5.2.13

5.2.4(A) Software Tokens

- Similar to physical tokens there also exist software-based tokens. Software tokens are used for security claims (authorisation) between two communicating parties. One such implementation, based on RFC 7519, is JSON Web Token or JWT (pronounced Jot).

***Definition** : JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties.*

- The general concept behind a token-based authentication system is to allow a user to enter her username and password in order to obtain a web token that allows her to fetch a specific resource - without using her username and password again. Once her token has been obtained, she can offer the token that provides access to the specific resources for a time period. Tokens come with an expiry time and need to be refreshed periodically.

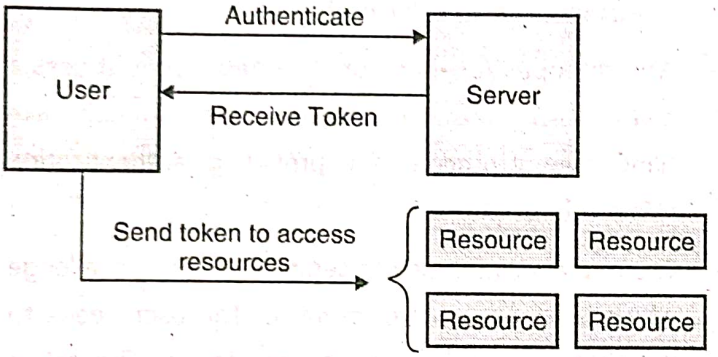


Fig. 5.2.14

Usage of software tokens

Two common scenarios where software tokens are used are

1. Authorisation

Once the user is logged in, each subsequent request will include the token, allowing the user to access resources that are permitted with that token. The user does not require re-authentication several times to continue accessing the permitted resources.

2. Information exchange

Tokens are also a good way of securely transmitting information between parties. Tokens can be signed, and non-repudiation can be enforced.

JWT Structure

A JWT consists of 3 parts separated by dot (".") –

1. Header

The header typically consists of two parts: the type of the token, which is JWT, and the signing algorithm being used, such as HMAC SHA256 or RSA.

For example,

```
{
 "alg": "HS256",
 "typ": "JWT"
}
```

Here algorithm is HMAC SHA 256 written HS256 in short. Token type is JWT.

2. Payload

The second part of the token is the payload, which contains the claims. Claims are statements about an entity (typically, the user) and additional data.

For example,

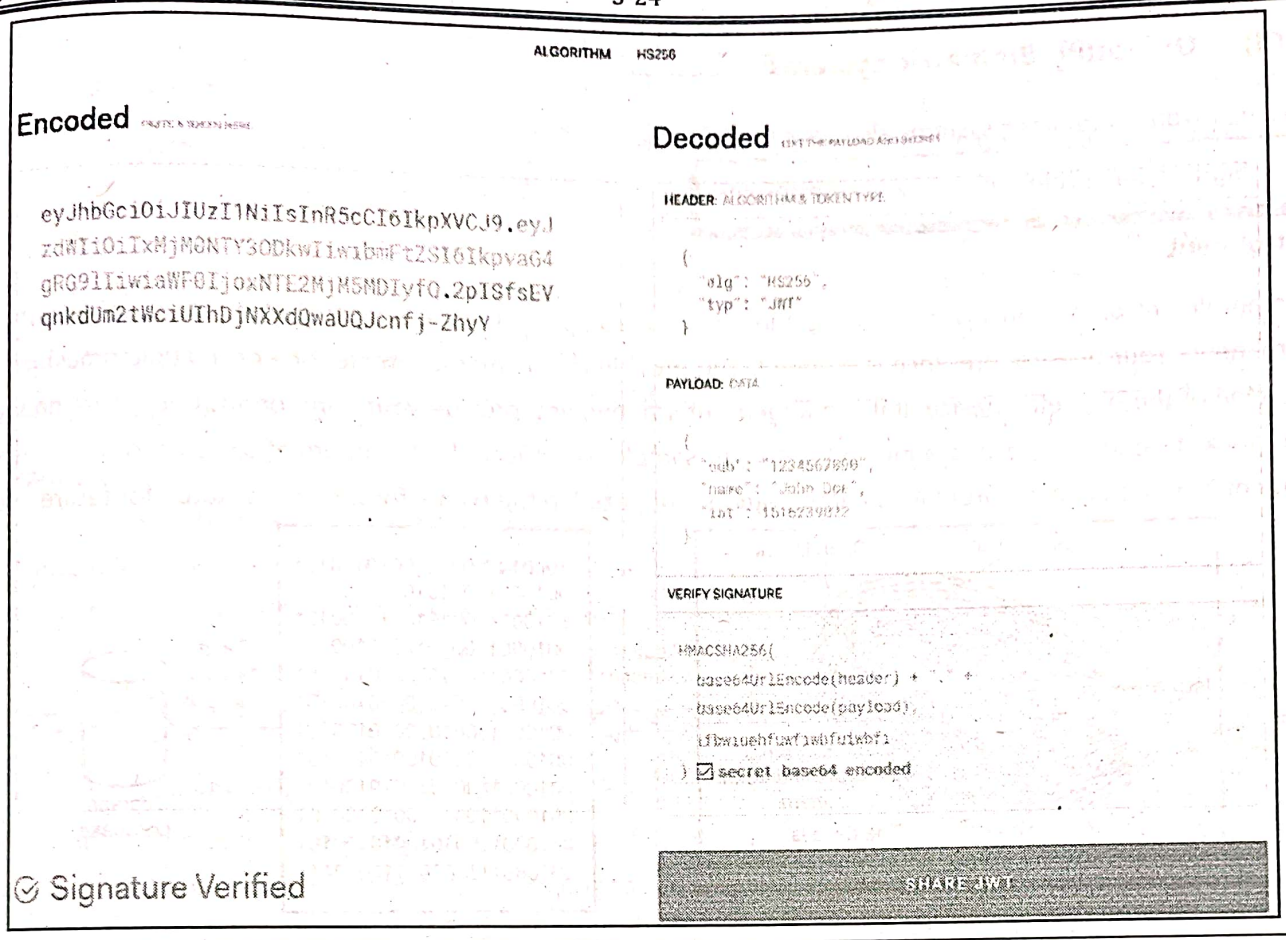
```
{
 "sub": "1234567890",
 "name": "John Doe",
 "admin": true
}
```

3. Signature

To create the signature part you have to take the encoded header, the encoded payload, a secret, the algorithm specified in the header, and sign that. For example if you want to use the HMAC SHA256 algorithm, the signature will be created in the following way:

```
HMACSHA256(
 base64UrlEncode(header) + "." +
 base64UrlEncode(payload),
 secret)
```

- The signature is used to verify the message wasn't changed along the way, and, in the case of tokens signed with a private key, it can also verify that the sender of the JWT is who it says it is.
- All the 3 parts are Base64Url encoded. The following example shows a JWT that has the previous header and payload encoded, and it is signed with a secret.



Note : You can go to https://jwt.io/ and try out JWT.

5.2.5 Biometric based Authentication

The biometric-based authentication relies upon "Someone You Are" and "Something You Do". "Something You Are" utilizes physical characteristics of your body whereas "Something You Do" utilizes your behavioural characteristics. Biometric is the most expensive way of authentication. It requires specialized readers, processing, storage and comes with errors.

5.2.5(A) Components of Biometric Systems

Any typical biometric system comprises of the following components as shown in Fig. 5.2.15.

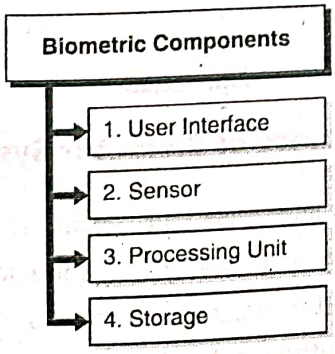


Fig. 5.2.15 : Biometric components

1. **User Interface :** This is the equipment (or the part of the overall biometric system) which serves as the input and output device. The user interacts with it. It could be the glass where you place your fingers for scanning or could be a microphone where you give your voice print.
2. **Sensor :** Sensor is the most critical part of the biometric system. It extracts the authentication related information from the provided input and passes it to the processing unit. Sensor should be able to adequately read the information and should be error free as much as possible. If the sensor has problems, it could either mean accepting unauthorised individuals or rejecting authorised individuals. Both conditions are dangerous.
3. **Processing Unit :** The processing unit evaluates the captured information and performs any processing required for overall working of the biometric system.
4. **Storage :** A storage mechanism (or unit) is required to keep the collected biometric samples from the individuals for matching them as and when needed for authentication purpose.

5.2.5(B) Operating Biometric Systems

Like passwords, biometric systems also have two phases.

1. Sign up (or enrolment) 2. Verification

1. Enrolment

- The enrolment process involves collecting the biometric sample from the individual. Remember your Aadhar card enrolment? Your physical presence is required. You provide the biometric sample (one or multiple times) based on the type of the biometric device. If it is a fingerprint scanner, you provide your fingerprints. If it is a retina scanner, you look through an eye scanner. Once your sample is collected, the information that can be used for authentication is extracted from it. The information is digitized in the binary format and is stored for future use.

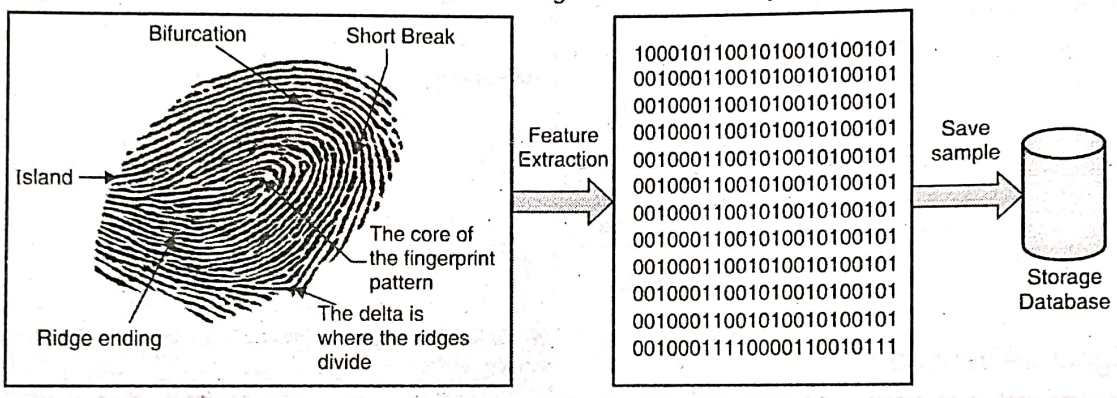


Fig. 5.2.16

- You can understand the enrolment process using a simplistic schematic diagram.

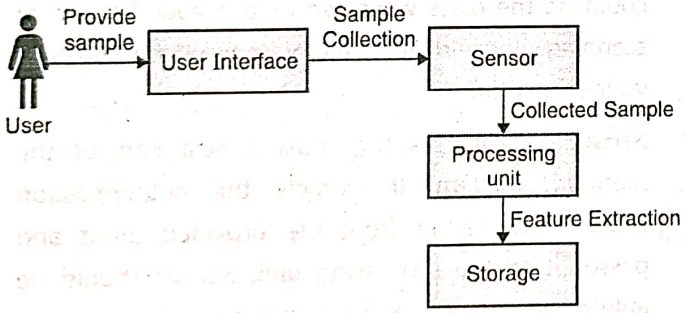


Fig. 5.2.17

2. Verification

- The verification process is quite straightforward. Your physical presence is again required. You offer the same type of sample that you provided during the enrolment process. For example, if you enrolled with fingerprints, you would need to provide fingerprints again for verifying.
- Your provided verification sample is compared against the enrolment sample based on your identity. Once the two samples match, you are successfully verified. If the samples do not match, the authentication is rejected. This might be due to some error. In that case, you can retry providing the sample again.

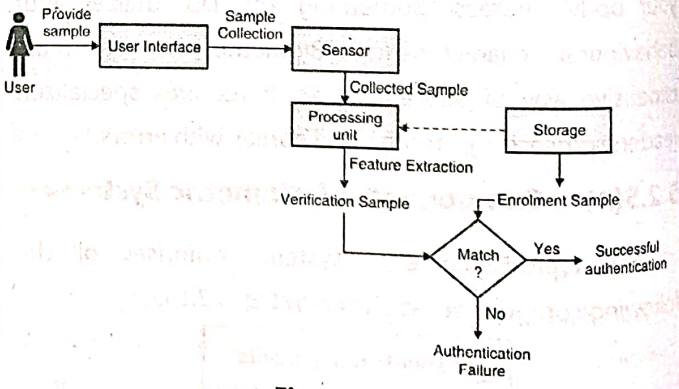


Fig. 5.2.18

5.2.5(C) Accuracy of Biometric Systems

- You understand that an enrolment sample is compared against the corresponding verification sample to find a match. If a match is found, you are successfully authenticated.

- Biometric systems are prone to two types of errors as shown in Fig. 5.2.19

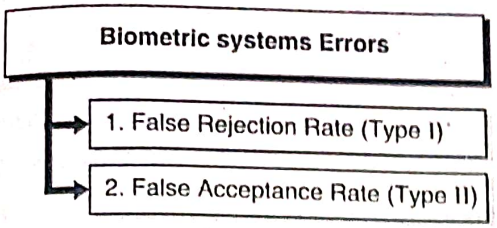


Fig. 5.2.19

1. False Rejection Rate (FRR) or Type I error

✍ ***Definition :*** *When a biometric system rejects an authorised individual it is called Type I error.*

✍ ***Definition :*** *False Rejection Rate (FRR) is the ratio of number of incorrect rejections to the total number of authentication attempts made.*

$$FRR = \frac{\text{Number of Incorrect Rejections}}{\text{Total Number of Authentication Attempts Made}}$$

The higher the FRR the more error prone is the biometric system. The authorised individuals would likely get frustrated from repeated attempts required to pass authentication.

2. False Acceptance Rate (FAR) or Type II Error

✍ ***Definition :*** *When a biometric system accepts an unauthorised individual it is called Type II error.*

✍ ***Definition :*** *False Acceptance Rate (FAR) is the ratio of number of incorrect acceptances to the total number of authentication attempts made.*

$$FAR = \frac{\text{Number of Incorrect Acceptances}}{\text{Total Number of Authentication Attempts Made}}$$

The higher the FAR the more error prone is the biometric system. The unauthorised individuals would likely get successfully authenticated. Such unauthorised individuals may cause harm to the system.

3. Crossover Error Rate (CER)

✍ ***Definition :*** *Crossover Error Rate is the point at which False Rejection Rate (FRR) is equal to the False Acceptance Rate (FAR).*

- This means that the biometric system is not more likely to produce one type of error than the other type. If the system rejects too many authorised individuals the FRR would be high. Similarly, if the system accepts too many unauthorised individuals the FAR would be high. CER is also called as Equal Error Rate (EER).
- When comparing various biometric systems, choose the one with lower CER value. The lower the CER value the more accurate the biometric system is. For example, a biometric system having CER value 3 would be more accurate than the biometric system having CER value of 5.

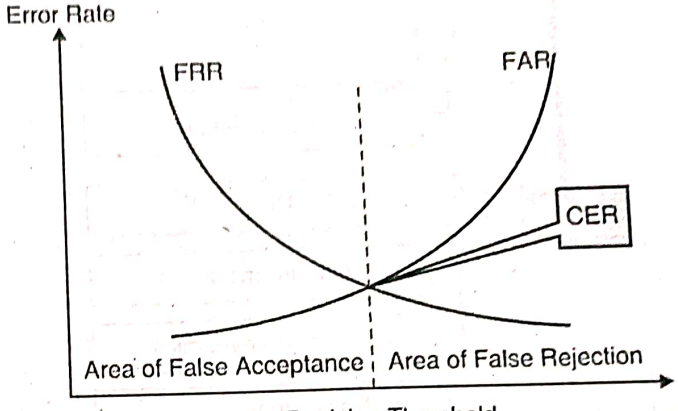


Fig. 5.2.20

Ex. 5.2.1 : A biometric system rejects 2 authorised attempts out of 10. Calculate its FRR.

Soln. :

$$FRR = \frac{\text{Number of Incorrect Rejections}}{\text{Total Number of Authentication Attempts Made}}$$

Hence, FRR = 2/10 = 0.2

Ex. 5.2.2 : A biometric system accepts 4 unauthorised attempts out of 10. Calculate its FAR.

Soln. :

$$FAR = \frac{\text{Number of Incorrect Acceptances}}{\text{Total Number of Authentication Attempts Made}}$$

Hence, FAR = 4/10 = 0.4

5.2.5(D) Types of Biometric Systems

There are several types of biometric systems in use today. These systems are used depending upon the type of application, the level of authentication required, processing speed and authentication accuracy (Type I errors and Type II errors). Let us learn about some of the most commonly used biometric systems as shown in Fig. 5.2.21.

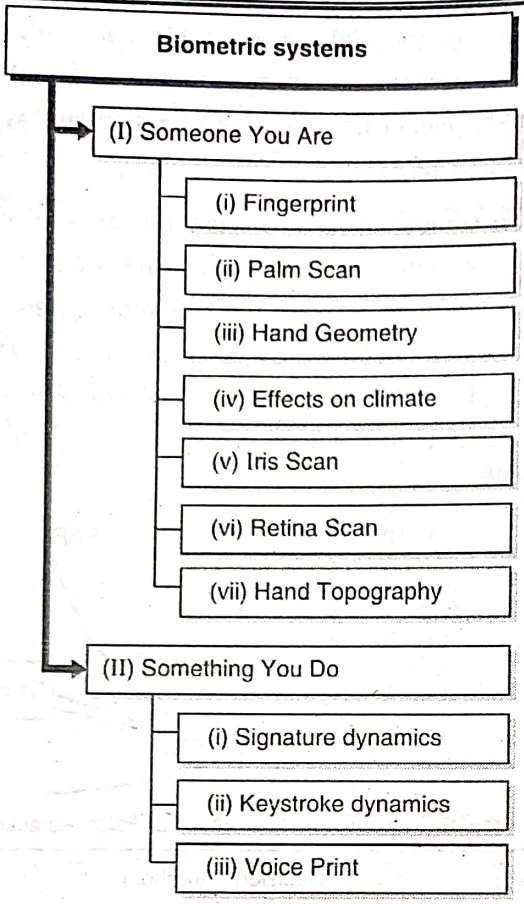


Fig. 5.2.21 : Biometric systems

1. Someone You Are

(i) Fingerprint : Fingerprint is one of the most commonly used type of biometric system. Your fingerprint is made up of several curves and endpoints. These when combined together are unique enough to identify you.

(ii) Palm Scan : You would have seen this in several movies. Your palm contains several curves, lines, endpoints, folds and textures that can uniquely identify you. Unlike fingerprints, you need to place your entire palm on the biometric sensor to enroll and verify yourself.

(iii) Hand Geometry : Your hand holds several key attributes such as shape, length, width, size etc. These attributes could fulfill biometric requirements to provide authentication information.

(iv) Retina Scan : Retina scan involves reading the blood-vessel pattern of retina on the backside of the eyeball. This pattern is unique enough to identify you. You are required to look into an eye scanner (fitted with a high-resolution camera). The scanner captures the pattern and stores it for authentication.

(v) Iris Scan : The iris is the colored portion of the eye that surrounds the pupil. The iris has unique patterns, rifts, colors, rings, coronas, and furrows. When you look into an eye scanner, these characteristics are captured and stored for authentication. Iris scan is the most accurate form of biometric system.

(vi) Facial Scan : Your face can be digitally identified as well. It has got several key features such as nose ridges, eye widths, chin shape, forehead size, bone structures, etc. These features are extracted during a facial scan and stored for authentication.

(vii) Hand Topography : Hand topography captures overall hand shape and structure. The peaks and valleys in your hand provide biometric traits that can be used for authentication. The entire hand is placed on a scanner and there are several cameras that capture the hand information from various angles.

2. Something You Do

(i) Signature Dynamics : You sign at a particular speed making similar strokes each time. The signing process generates electrical signals that can be captured to provide biometric authentication.

(ii) Keystroke Dynamics : Keystroke dynamics capture electrical signals when you type using a keyboard. You type at a particular speed applying a specific key pressure, timing and rhythm for each key that you press. These dynamics can be captured and used for authentication.

(iii) Voice Print : Aren't you using Alexa, Siri or Ok Google yet? How does that work? Basically, you train the device to recognize your voice pattern. Your voice pattern has specific pitch, tone, amplitude and frequency that can be used to create a voice print (similar to fingerprint). These attributes are stored and when you are required to authenticate you are asked to speak a set of words or sentences to capture your voice print and compare it with the previously stored voice print information.

5.2.5(E) Comparison of Biometric Systems

Table 5.2.3

Biometric System	Processing Speed	Accuracy	Ease of Enrolment
Fingerprint	High	High	High
Palm Scan	Medium	High	Medium
Hand Geometry	Low	Medium	Low
Retina Scan	Medium	High	High
Iris Scan	Medium	High	High
Facial Scan	Medium	Low	Medium
Hand Topography	Low	Low	Low
Signature dynamics	Medium	Medium	Medium
Keystroke dynamics	Medium	Medium	Medium
Voice Print	Medium	Medium	Medium

5.3 Linux and Windows Vulnerabilities

5.3.1 Vulnerabilities

- Linux and Windows OSs are being widely used for over 20 years now. These OSs support several programs and functionalities and have over million lines of code. Their wide adoption and the large code base attract attackers world-wide to find vulnerabilities and carry out exploits. At the same time, the security community and the vendor companies run several research and bug bounty programs to identify the vulnerabilities beforehand and fix them before they are widely exploited.
- Let us look at some of the statistics around Linux and Windows vulnerabilities as per NVD (National Vulnerability Database - https://nvd.nist.gov/vuln/search).
- The number of vulnerabilities reported in Microsoft Windows® Server 2016 (OS only) since launch are as following.

Year	Vulnerability Count
2016	47
2017	416
2018	383
2019	581
2020	867
2021	510
2022	522

- The number of vulnerabilities reported in Red Hat Enterprise Linux 8 (OS only) are as following.

Year	Vulnerability Count
2018	3
2019	177
2020	87
2021	121
2022	166

- So, as you see, there are several vulnerabilities that are found every year in these OSs. These vulnerabilities can be categorised at a high level as per the STRIDE model as shown in the Table 5.3.1.

Table 5.3.1

Attack Category	Security Property that is attacked
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-repudiation
Information Disclosure	Confidentiality
Denial of service	Availability
Elevation of Privilege	Authorisation

1. **Spoofing :** When someone tries to steal your identity, it is called spoofing. Spoofing is also called as impersonating someone or something.

- For example, if I try to login to your Facebook account using your email address and password guesses, I am trying to spoof or impersonate your identity. Spoofing can also be used for fake news, fake

websites, fake or malicious files or anything else that could be mistaken for being real. The target of spoofing attacks is to crack authenticity (of person, file, website, news or anything real).

2. **Tampering :** When someone tries to do an unauthorised modification to something, it is called tampering.

– You would have heard of cricketers tampering with the ball to suit their requirements. It is a punishable offense. Similarly, in information systems, if you try to tamper with say files, emails, or any other information in an unauthorised way, it is called tampering. Basically, you are trying to attack the integrity of the object by making such alterations.

3. **Repudiation :** In this, you are trying to falsely claim that you didn't carry out a particular action.

– For example, you didn't send an email or didn't visit a website. Remember your childhood days when you broke something, and you try to escape the punishment saying that you didn't actually break it and then your parents finding proofs that it was indeed you who broke it. That's precisely what is non-repudiation.

– In repudiation attacks, you are trying to destroy evidences that someone can use to falsify your denial claims.

4. **Information Disclosure :** This pertains to unauthorised revealing of any confidential information. In these attacks, the attacker wishes to know the confidential information and tries to crack controls around it to get hold of such confidential information.

5. **Denial of service :** In this type of attacks, the purpose of attack is to make the information system or its services unstable or unavailable to perform its assigned activities.

– For example, if you can succeed to bring Flipkart website temporarily down, it might mean severe loss of business for Flipkart and the customers may go to a different website for placing their urgent orders.

6. **Elevation of Privileges :** In this category of attack, the attacker tries to get elevated (higher) privileges (permissions / authority) over resources.

– For example, if you are only a user on a system and you try to attack the system to become an administrator on the system, it is called an elevation of privileges attack.

– The vendors are constantly providing security patches that when installed would mitigate the reported vulnerability.

– So, what can you do about these vulnerabilities? How can you protect your systems? The answer is simple, by following security risk management framework that you learnt earlier.

(i) Identify : Identify what security controls your OS requires

(ii) Protect : Put those identified security controls to safeguard the OS

(iii) Detect : Put controls in place that can find out if the OS is attacked

(iv) Respond : Have controls in place that can respond to the attacks

(v) Repair : Have controls in place that can restore the system functionality and security

– Out of these steps, let's focus specifically on protection for OS from vulnerabilities.

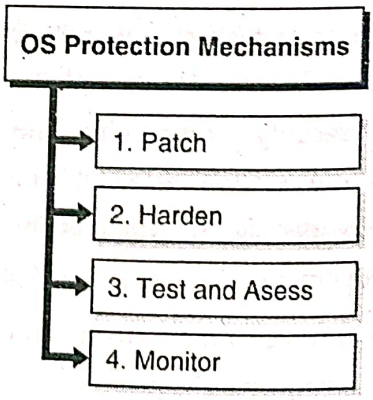


Fig. 5.3.1

1. Patch

✍ ***Definition :*** *A software patch is a top-up capability to install any corrections, repairs or new functionalities on a system.*

- Patching allows to make selective corrections or introduce new functionalities in the software without significantly changing the underlying software or user data. There are majorly two kinds of patch.

(i) **Security Patch :** This type of patch fixes a known vulnerability in the software.

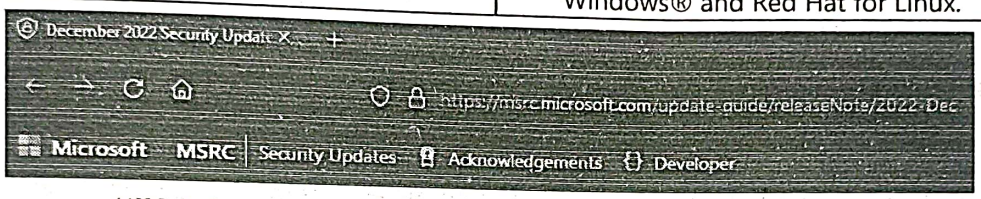
(ii) **Functional Patch :** This type of patch introduces new functionalities or makes any other desired changes to the existing functionalities.

- I would be limiting our discussion to only security patches here.
- Patching the system regularly with security patches is perhaps the most crucial way to mitigate software vulnerabilities. As new vulnerabilities are discovered, the software vendors release the security patches and let the users know which vulnerabilities those patches fix. The Following Table 5.3.2 is a typical patch release cycle (or schedule) for major commercial software vendors.

Table 5.3.2 : A typical patch release cycle

Vendor Name	Patch Cycle
Microsoft	2nd Tuesday of every month
Adobe	2nd Tuesday of every month
Red Hat	No fix cycles. Release as available
Apple	No fix cycles. Release as available
Oracle	Once per quarter
Google	No fix cycles. Release as available

- Vendors release security advisories to detail about the vulnerability and the patches. Following are the examples of security advisories from Microsoft for Windows® and Red Hat for Linux.



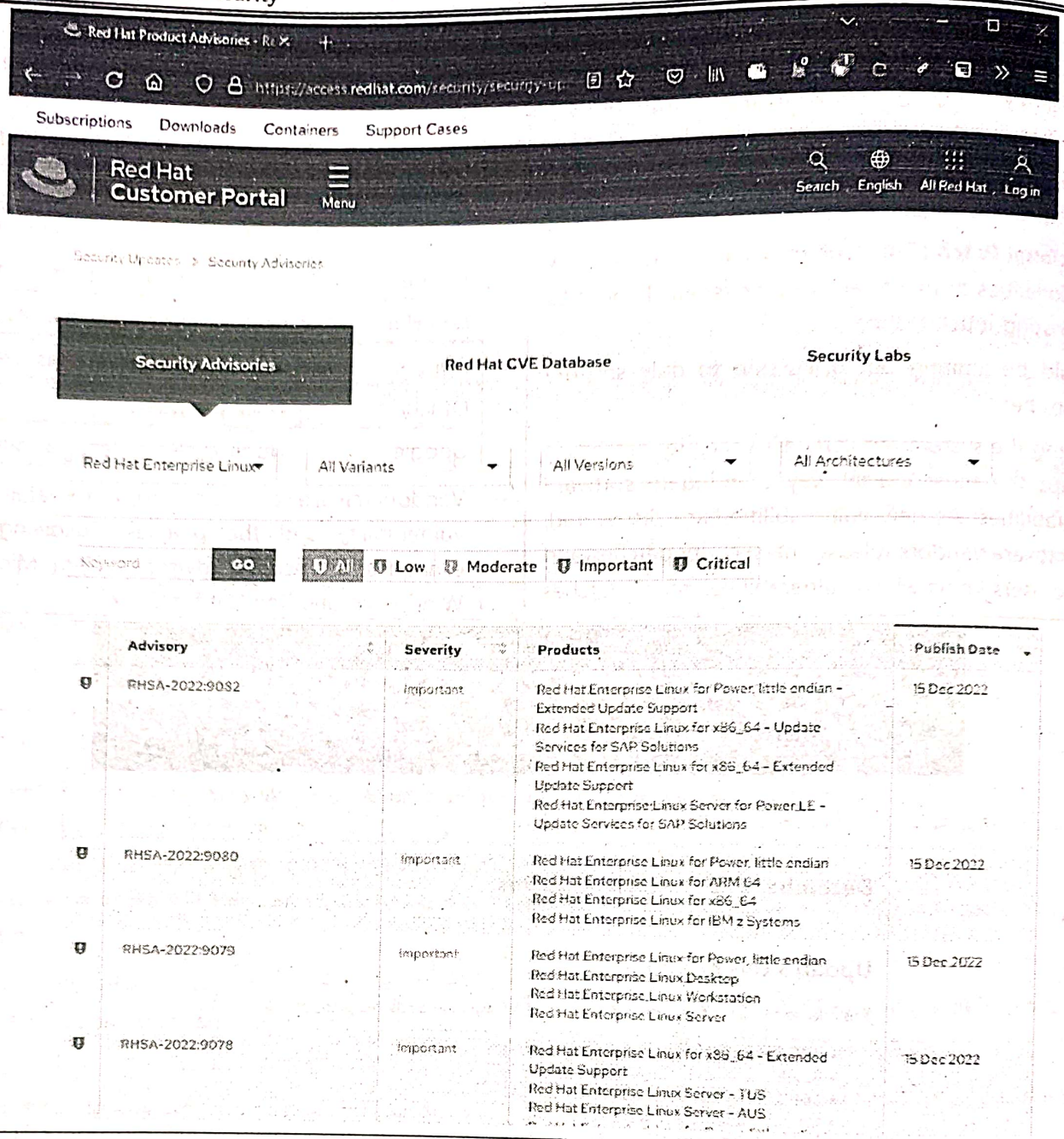
MSRC > Customer Guidance > Security Update Guide > Release Notes > 2022 Dec

December 2022 Security Updates

Updates this Month

This release consists of security updates for the following products, features and roles.

- .NET Framework
- Azure
- Client Server Run-time Subsystem (CSRSS)
- Microsoft Bluetooth Driver
- Microsoft Dynamics
- Microsoft Edge (Chromium-based)
- Microsoft Graphics Component
- Microsoft Office
- Microsoft Office OneNote
- Microsoft Office Outlook
- Microsoft Office SharePoint
- Microsoft Office Visio
- Microsoft Windows Codecs Library
- Role: Windows Hyper-V
- Sysinternals
- Windows Certificates
- Windows Contacts
- Windows DirectX
- Windows Error Reporting
- Windows Fax Compose Form
- Windows HTTP Print Provider
- Windows Kernel
- Windows PowerShell
- Windows Print Spooler Components
- Windows Projected File System
- Windows Secure Socket Tunneling Protocol (SSTP)
- Windows SmartScreen
- Windows Subsystem for Linux
- Windows Terminal



Steps in Patch Management

– Typically, the patch management process involves the following steps as shown in Fig. 5.3.2.

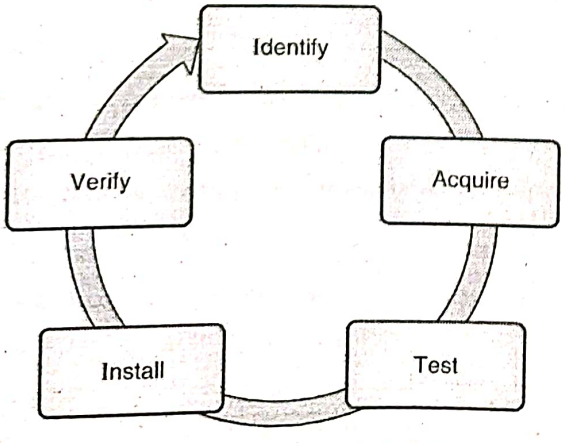


Fig. 5.3.2 : Patch management process

– You would need to carry out these steps periodically or at a frequency defined by your organisation. You might also need to carry out these steps on urgent basis for any critical security patches that are released out of the regular release schedule or cycle. These attacks are usually called Zero-day Attacks because they are not previously known, and the vendor has to plan for it in a hurry and on urgent basis.

– Let's look through these steps in detail.

(A) Identify : You start by evaluating what patches are required for your system. The analysis could be manually done or could be preferably done by a patch management software. You not only need to

evaluate OS-specific patches but also any applications that you have installed on top of it. For example, if you have installed Java on a Windows® OS, you would need security patches for both Java and Windows® OS, if applicable. From this step, you get a list of security patches that are applicable for your system and would require installation to fix the identified security vulnerabilities.

(B) **Acquire :** The next step is to acquire or get those identified security patches. It is crucial that these patches are downloaded only from a secure website as directed in the vendor's security advisory. You should not download patches from any site that claims to be hosting security patches. Always download the security patches from the official and vendor recommended location only. The patches downloaded from elsewhere may contain malicious code that might get installed as you install those malicious patches. In that scenario, instead of securing your system, the installation of malicious code would cause further harm.

(C) **Test :** Before installing the acquired patches on the production systems (systems that are used by your customers or by many people in your organisation), you should test those patches on a non-production system (system used only for testing) that has a similar configuration as your production system to identify any issues that might occur from installing the acquired patches.

- If there are no potential issues and your system and applications on it continue to work as before installing the patch, the acquired patch is safe to install.
- If you see any issues after installing the patch, you should report the issue details to the vendor so that it can fix the patch and re-release the patch so that you can safely install the patch without issues or without affecting your desired system functionality.

(D) **Install :** Once your tests are successful for the acquired patches, you can go ahead and install those patches on your production systems. Some patches require reboot of the system. You should adequately plan for the downtime of the system so that the patches can be installed, and the system functionality can be resumed.

(E) **Verify :** Once you have installed the patches, you should re-verify the patch installation status and also the system functionality. The patch management software usually gives you the patch installation status. If there are any undesirable results, patches can be rolled back (uninstalled) to restore the previous behaviour.

2. Harden

- In Chapter 1, under the topic "Secured and hardened software baselines", you learnt about the importance of security baseline and the hardening process. The basic premise behind hardening is to reduce the attack surface area. As you know, there are millions of lines of code in the OS, and if you lock down certain features or uninstall things that you don't plan to use, you are indirectly reducing the number of lines of codes that could be potentially exploited for any vulnerability. Uninstalling unused features and locking down the remaining features ensure that you have a secure baseline.

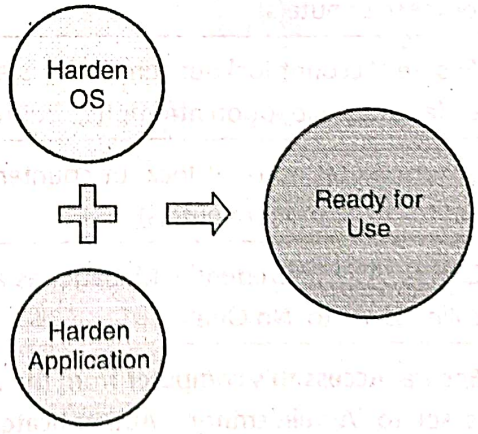


Fig. 5.3.3

- It is important to note that like patching, the hardening process must be applied not only to the OS but also to any other applications that you might have installed on it. There are various guidelines that detail out what you could potentially do to establish a secure baseline for Windows® and Linux OS. One

such guideline vendor is CIS (Center for Internet Security - https://www.cisecurity.org/). Let's review some of the guidelines for hardening for both Windows® and Linux as per CIS Security Benchmarks.

- A part of the CIS Security Benchmark for Windows® 2016 is listed in the Table 5.3.3 for your reference. The benchmark contains around 500 recommendations for hardening Windows® 2016.

Table 5.3.3

Control Category	Recommendation
Password Policy	Ensure 'Enforce password history' is set to '24 or more password(s)'
	Ensure 'Maximum password age' is set to '60 or fewer days, but not 0'
	Ensure 'Minimum password age' is set to '1 or more day(s)'
	Ensure 'Minimum password length' is set to '14 or more character(s)'
	Ensure 'Password must meet complexity requirements' is set to 'Enabled'
	Ensure 'Store passwords using reversible encryption' is set to 'Disabled'
Account Lockout Policy	Ensure 'Account lockout duration' is set to '15 or more minute(s)'
	Ensure 'Account lockout threshold' is set to '10 or fewer invalid logon attempt(s), but not 0'
	Ensure 'Reset account lockout counter after' is set to '15 or more minute(s)'
User Rights Assignment	Ensure 'Access Credential Manager as a trusted caller' is set to 'No One'
	Ensure 'Access this computer from the network' is set to 'Administrators, Authenticated Users' (MS only)
	Ensure 'Act as part of the operating system' is set to 'No One'
	Ensure 'Adjust memory quotas for a process' is set to 'Administrators, LOCAL SERVICE, NETWORK SERVICE'
Accounts	Ensure 'Accounts: Administrator account status' is set to 'Disabled' (MS only)
	Ensure 'Accounts: Block Microsoft accounts' is set to 'Users can't add or log on with Microsoft accounts'
	Ensure 'Accounts: Guest account status' is set to 'Disabled' (MS only)
	Ensure 'Accounts: Limit local account use of blank passwords to console logon only' is set to 'Enabled'
	Configure 'Accounts: Rename administrator account'
	Configure 'Accounts: Rename guest account'
Interactive logon	Ensure 'Interactive logon: Do not display last user name' is set to 'Enabled'
	Ensure 'Interactive logon: Do not require CTRL+ALT+DEL' is set to 'Disabled'
	Ensure 'Interactive logon: Machine inactivity limit' is set to '900 or fewer second(s), but not 0'
Microsoft network client	Ensure 'Microsoft network client: Digitally sign communications (always)' is set to 'Enabled'
	Ensure 'Microsoft network client: Digitally sign communications (if server agrees)' is set to 'Enabled'
	Ensure 'Microsoft network client: Send unencrypted password to third-party SMB servers' is set to 'Disabled'
Network security	Ensure 'Network security: Configure encryption types allowed for Kerberos' is set to 'AES128_HMAC_SHA1, AES256_HMAC_SHA1, Future encryption types'
	Ensure 'Network security: Do not store LAN Manager hash value on next password change' is set to 'Enabled'
	Ensure 'Network security: Force logoff when logon hours expire' is set to 'Enabled'
	Ensure 'Network security: LAN Manager authentication level' is set to 'Send NTLMv2 response only. Refuse LM & NTLM'

Control Category	Recommendation
	Ensure 'Network security: LDAP client signing requirements' is set to 'Negotiate signing' or higher
Account Management	Ensure 'Audit Application Group Management' is set to 'Success and Failure'
	Ensure 'Audit Computer Account Management' is set to 'Success and Failure'
	Ensure 'Audit Other Account Management Events' is set to 'Success and Failure'
	Ensure 'Audit Security Group Management' is set to 'Success and Failure'
	Ensure 'Audit User Account Management' is set to 'Success and Failure'

- A part of the CIS Security Benchmark for Red Hat Enterprise Linux 7 is listed in the Table 5.3.4 for your reference. The benchmark contains around 200 recommendations for hardening Red Hat Enterprise Linux 7.

Table 5.3.4

Control Category	Recommendation
Filesystem Configuration	Ensure nodev option set on /tmp partition
	Ensure nosuid option set on /tmp partition
	Ensure noexec option set on /tmp partition
	Ensure sticky bit is set on all world-writable directories
	Disable Automounting
Disable Unused Filesystems	Ensure mounting of cramfs filesystems is disabled
	Ensure mounting of freevxfs filesystems is disabled
	Ensure mounting of jffs2 filesystems is disabled
	Ensure mounting of hfs filesystems is disabled
Configure Software Updates	Ensure package manager repositories are configured
	Ensure gpgcheck is globally activated
	Ensure GPG keys are configured
	Ensure Red Hat Subscription Manager connection is configured
Services	Ensure Avahi Server is not enabled
	Ensure CUPS is not enabled
	Ensure DHCP Server is not enabled
	Ensure LDAP server is not enabled
	Ensure NFS and RPC are not enabled
	Ensure DNS Server is not enabled
	Ensure FTP Server is not enabled
	Ensure HTTP server is not enabled
	Ensure IMAP and POP3 server is not enabled
	Ensure Samba is not enabled
Set Shadow Password Suite Parameters	Ensure minimum days between password changes is 7 or more
	Ensure password expiration warning days is 7 or more
	Ensure inactive password lock is 30 days or less
	Ensure all users last password change date is in the past
System File Permissions	Ensure permissions on /etc/passwd are configured
	Ensure permissions on /etc/shadow are configured
	Ensure permissions on /etc/group are configured
User and Group Settings	Ensure no duplicate UIDs exist
	Ensure no duplicate GIDs exist
	Ensure no duplicate user names exist
	Ensure no duplicate group names exist

– So, do you realize the importance of hardening now? The OS and the applications have several such features that require to be carefully configured before use so that they do not open security vulnerabilities. Failing to adequately harden the OS and the application could compromise the system by exploiting the vulnerabilities.

3. Test and Assess

✍ ***Definition :*** *Security assessment is the process of determining how effectively an entity being examined (e.g. system, network, procedure, person) meets the specific security objectives.*

– As you test your system for functionality, similarly, you should test your system for security often. Testing ensures that your system meets the security baseline that you desired to establish, and its behaviour is as you expected to be.

– Once you have applied the security patches and hardened the system, you should conduct penetration tests to find out if there are any gaps that you should fill based on your system. Without testing, your system might still be vulnerable.

4. Monitor

✍ ***Definition :*** *Continuous Monitoring is a process of ensuring ongoing awareness of information security, vulnerabilities, and threats to support organisational risk management decisions.*

– You should continuously monitor your systems to ensure that any deviation from the previously established security baselines, as new threats and vulnerabilities evolve is considered. You might have installed all the patches that are available today but tomorrow? New vulnerabilities would be found out and there could be new business requirements that need you to change certain features or configuration on the system.

– Typically, you deploy monitoring solutions that continuously assess your systems against the pre-established security baselines. So, for example, if you have disabled a service and someone enables it, such monitoring systems automatically detect that undesired change and raise a notification for you to take appropriate actions. You can also automate your actions where undesired changes are automatically rolled back or reverted to the desired configuration.

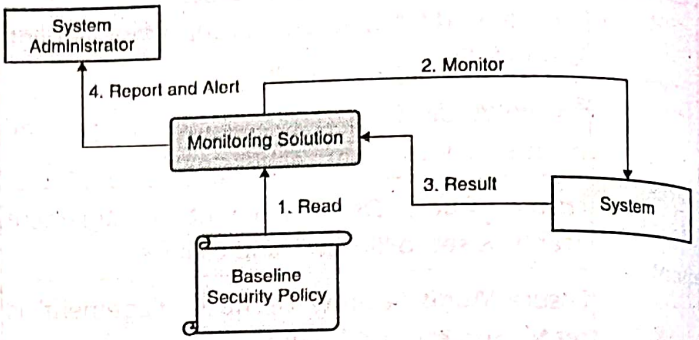


Fig. 5.3.4

– Monitoring is crucial for ensuring that your systems continue to run securely, and any vulnerability is properly mitigated within the appropriate time period.

5.3.2 File System Security

✍ ***Definition :*** *File system is a software mechanism that defines the way in which files are named, stored, organised, and accessed.*

– OS requires a file system to manage the several files present on the system. There are many operations on files that need to be properly carried out such as

(i) Opening the file

(ii) Closing the file

(iii) Reading / Writing the file

(iv) Storing the file

(v) Setting file permissions

(vi) Securing the file

– Let's learn about the file systems from security perspective for Windows® and Linux OS.

Note : The detailed explanation about file systems is out of scope for this book. I would be limiting the scope to the file system security only. Also, there were several file systems that have evolved over time. I would be restricting the discussion to only the latest file system on both Windows® and Linux.

5.3.2(A) File System Security on Windows®

(I) NTFS

NTFS (New Technology File System) is the default file system on Windows®. It brings several key advantages over other file systems including security features. Shows Fig. 5.3.5 the security features of NTFS.

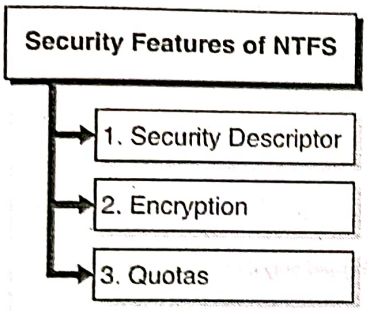
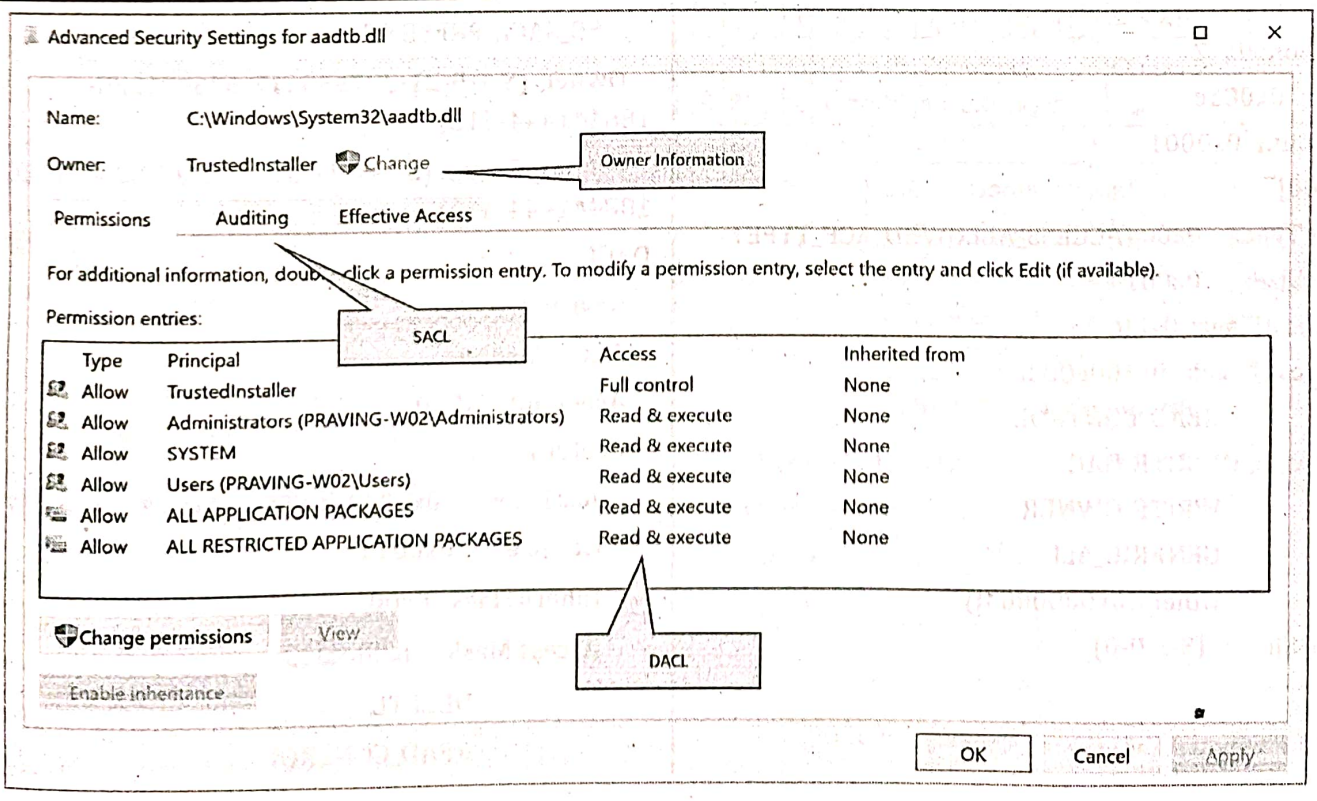


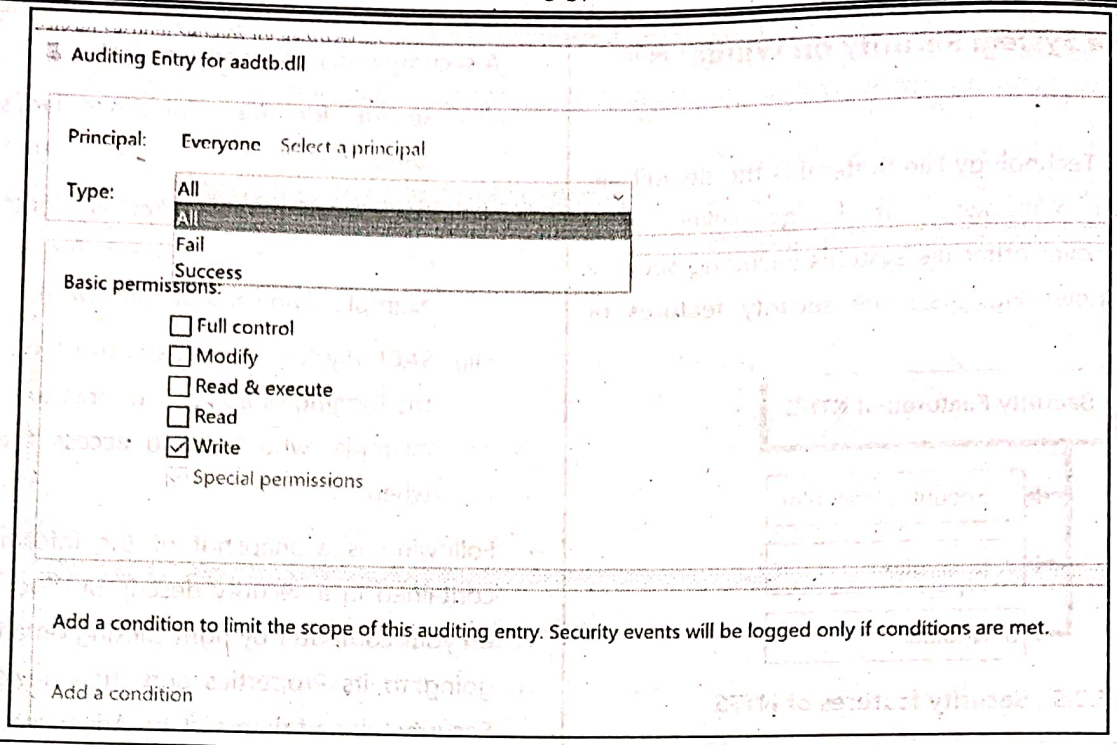
Fig. 5.3.5 : Security features of NTFS

1. Security Descriptor

✍ ***Definition** : Security descriptor contains the security information associated with a securable object.*

- Security descriptors can be associated with any named objects, including files, folders, shares, registry keys, processes, threads, named pipes, services, job objects and other resources.
- A security descriptor consists of the
  - (i) Security Identifiers for the object's owner and primary group
  - (ii) DACL (Discretionary Access Control List) that controls the access rights for an object. For example, who can read and write to the file.
  - (iii) SACL (System Access Control List) that controls the logging of access attempts to the object. For example, who tried to access the object and when.
- Following is a snapshot of the information that is contained in a security descriptor. You can view this on your computer by right clicking on a file name and going to its Properties and then navigating to the Security tab and then clicking Advanced.
- Going to the Auditing tab gives you the details of which access attempts will be logged (failed access attempts, successful access attempts or both). So, if a particular entity tries to access the object that access attempt would be logged in the logs based on whether the access attempt was successful or failure.





– Following is an example of how a security descriptor is internally defined for an object.

Revision: 0x00000001
Control: 0x0004
SE_DACL_PRESENT
Owner: (S-1-5-32-548)
PrimaryGroup: (S-1-5-21-397955417-626881126-188441444-512)
DACL
Revision: 0x02
Size: 0x001c
AceCount: 0x0001
Ace[00]
AceType: 0x00 (ACCESS_ALLOWED_ACE_TYPE)
AceSize: 0x0014
InheritFlags: 0x00
Access Mask: 0x100e003f
READ_CONTROL
WRITE_DAC
WRITE_OWNER
GENERIC_ALL
Others(0x0000003f)
Ace Sid : (S-1-0-0)
SACL
Not present

– DACL consists of the Access Control Entries (ACE) that determines which users have what level of permissions for a given object. There could be multiple ACE defined in a DACL as shown in the following example.

Revision: 0x00000001
Control: 0x0014
SE_DACL_PRESENT
SE_SACL_PRESENT
Owner: (S-1-5-21-397955417-626881126-188441444-512)
PrimaryGroup: (S-1-5-21-397955417-626881126-188441444-512)
DACL
Revision: 0x04
Size: 0x0104
AceCount: 0x0007
Ace[00]
AceType: 0x00 (ACCESS_ALLOWED_ACE_TYPE)
AceSize: 0x0014
InheritFlags: 0x00
Access Mask: 0x000f003f
DELETE
READ_CONTROL

WRITE_DAC
WRITE_OWNER
Others(0x0000003f)
Ace Sid: (S-1-5-18)
Ace[01]
AceType: 0x00 (ACCESS_ALLOWED_ACE_TYPE)
AceSize: 0x0024
InheritFlags: 0x00
Access Mask: 0x000f003f
DELETE
READ_CONTROL
WRITE_DAC
WRITE_OWNER
Others(0x0000003f)
Ace Sid: (S-1-5-21-397955417-626881126-188441444-512)
Ace[02]
AceType: 0x05 (ACCESS_ALLOWED_OBJECT_ACE_TYPE)
AceSize: 0x002c
InheritFlags: 0x00
Access Mask: 0x00000003
Others(0x00000003)
Flags: 0x00000001, ACE_OBJECT_TYPE_PRESENT
ObjectType: GUID_C_USER
InhObjectType: GUID ptr is NULL
Ace Sid: (S-1-5-32-548)
Ace[03]
AceType: 0x05 (ACCESS_ALLOWED_OBJECT_ACE_TYPE)
AceSize: 0x002c
InheritFlags: 0x00
Access Mask: 0x00000003
Others(0x00000003)
Flags: 0x00000001, ACE_OBJECT_TYPE_PRESENT
ObjectType: GUID_C_GROUP
InhObjectType: GUID ptr is NULL
Ace Sid: (S-1-5-32-548)
SACL
Revision: 0x02
Size: 0x001c
AceCount: 0x0001
Ace[00]
AceType: 0x02 (SYSTEM_AUDIT_ACE_TYPE)
AceSize: 0x0014
InheritFlags: 0xc0
SUCCESSFUL_ACCESS_ACE_FLAG
FAILED_ACCESS_ACE_FLAG
Access Mask: 0x000d002b
DELETE
WRITE_DAC
WRITE_OWNER
Others(0x0000002b)
Ace Sid: (S-1-1-0)

– Let's understand how access control works with security descriptor. Do you see `Access Mask` in the examples?

✍ ***Definition :*** *An access mask is a 32-bit value whose bits correspond to the access rights supported by an object.*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GR	GW	GE	GA	Reserved			AS	Standard access rights								Object specific access right															

GR → Generic _ Rad
GW → Generic _ Write
GE → Generic _ Execute
GA → Generic _ ALL]
AS → Right to access SACL

Fig. 5.3.6 : Access matric

– There are four types of access rights that an access mask defines :

(i) Generic access rights
(ii) Standard access rights
(iii) SACL access rights
(iv) Directory services access rights

- When a process tries to access an object, it typically specifies an access mask to request a set of access rights it desires on the object. For example, a process that needs to set and query the values of a registry key can open the key by using an access mask to request the KEY_SET_VALUE and KEY_QUERY_VALUE access rights. If the ACE allows for the requested access rights to the entity, the access is granted else it is denied.
- The following Table 5.3.5 summarizes the various access rights.

Table 5.3.5 : Generic access rights

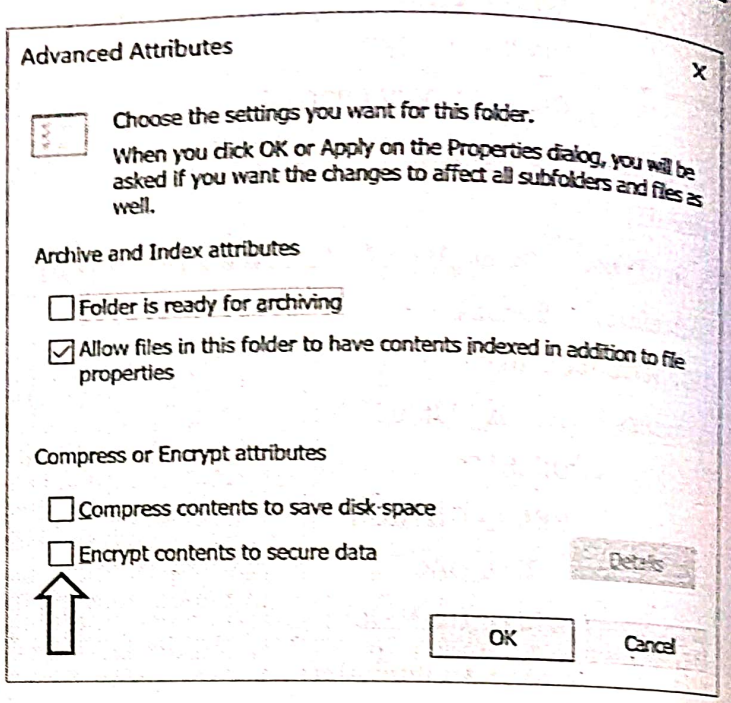
Constant	Right
GENERIC_ALL	All possible access rights
GENERIC_EXECUTE	Execute access
GENERIC_READ	Read access
GENERIC_WRITE	Write access

Table 5.3.6 : Standard access rights

Constant	Right
DELETE	The right to delete the object.
READ_CONTROL	The right to read the information in the object's security descriptor, not including the information in the System Access Control List (SACL).
SYNCHRONIZE	The right to use the object for synchronization.
WRITE_DAC	The right to modify the Discretionary Access Control List (DACL) in the object's security descriptor.
WRITE_OWNER	The right to change the owner in the object's security descriptor.

2. Encryption

- NTFS works with Encrypting File System (EFS) to provide file and folder level encryption per user. Encryption provides additional security in case a device is stolen, or a system is compromised.
- You can enable encryption by right clicking on a folder or a file, and then clicking on Advanced.



- EFS works using public key and symmetric key cryptography. EFS supports AES 256 and 3DES encryption algorithms.

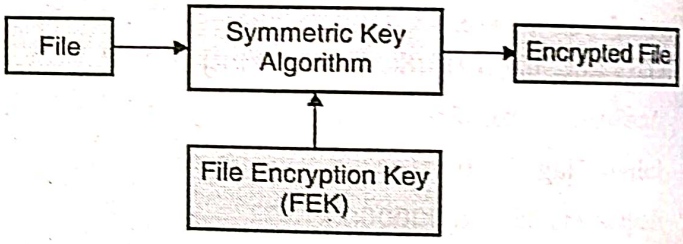


Fig. 5.3.7 (a)

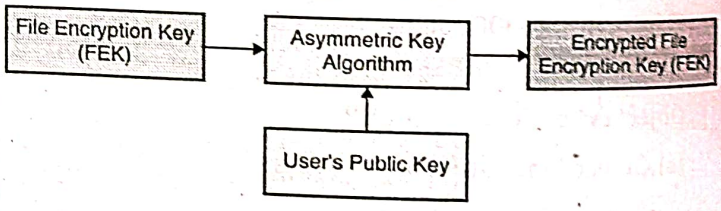


Fig. 5.3.7 (b)

- When you enable encryption for a file or a folder, EFS generates a random symmetric key called File Encryption Key (FEK) that is used to encrypt the file (or folder and its contents). The FEK is then encrypted using the user's public key and stored in a secured area.

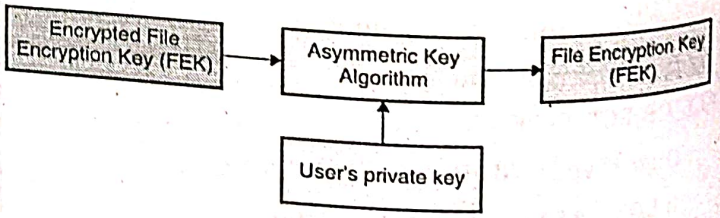


Fig. 5.3.8 (a)

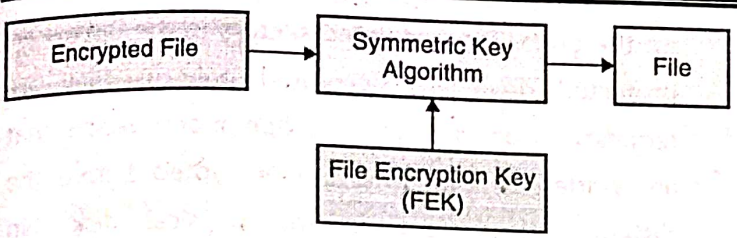
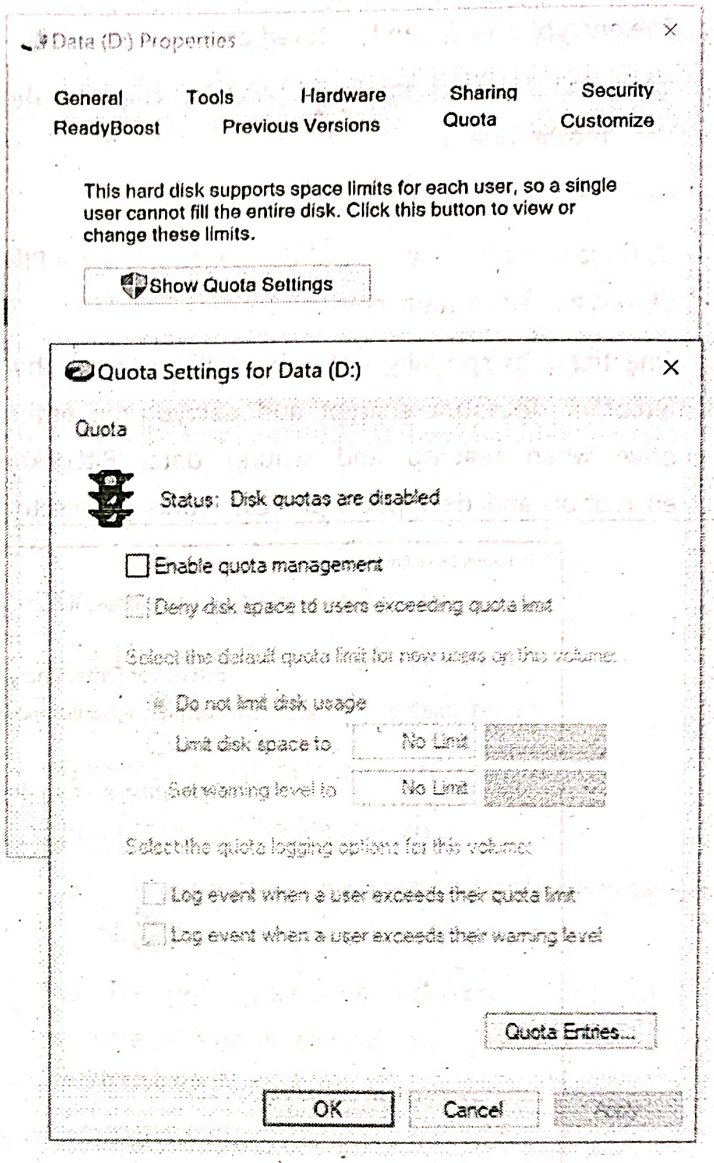


Fig. 5.3.8 (b)

- When an application or the user requires to read the file, the user's private key is used to decrypt the FEK. Once the FEK is decrypted, it can be used to further decrypt the file contents. The entire process is seamless (invisible) to the user and the EFS driver takes care of it internally.

3. Quota

- One of the tenets of security is availability. Quota ensures that each user can consume resources only up to the limit assigned to her.
- The NTFS file system supports disk quotas. Disk quotas allow administrators to control the amount of data that each user can store on an NTFS file system volume. Administrators can optionally configure the system to log an event when users are near their quota, and to deny further disk space to users who exceed their quota.
- Disk quotas are transparent to the user. When a user asks how much space is free on a disk, the system reports only the available quota allowance the user has available. If the user exceeds this allowance, the system returns the disk full error.
- To obtain more free disk space after exceeding the quota allowance, the user must do one or more of the following :
  - (i) Delete some files
  - (ii) Have another user claim ownership of some files
  - (iii) Have the administrator increase the quota allowance
- To set quota on a disk, you can right click on a disk and go to `Properties`. Under the Quota tab, click on Show `Quota Settings`. Configure the settings as appropriate.
- Disk quota ensures that a particular user cannot consume the entire disk space and make system unusable for other users.



(II) BitLocker

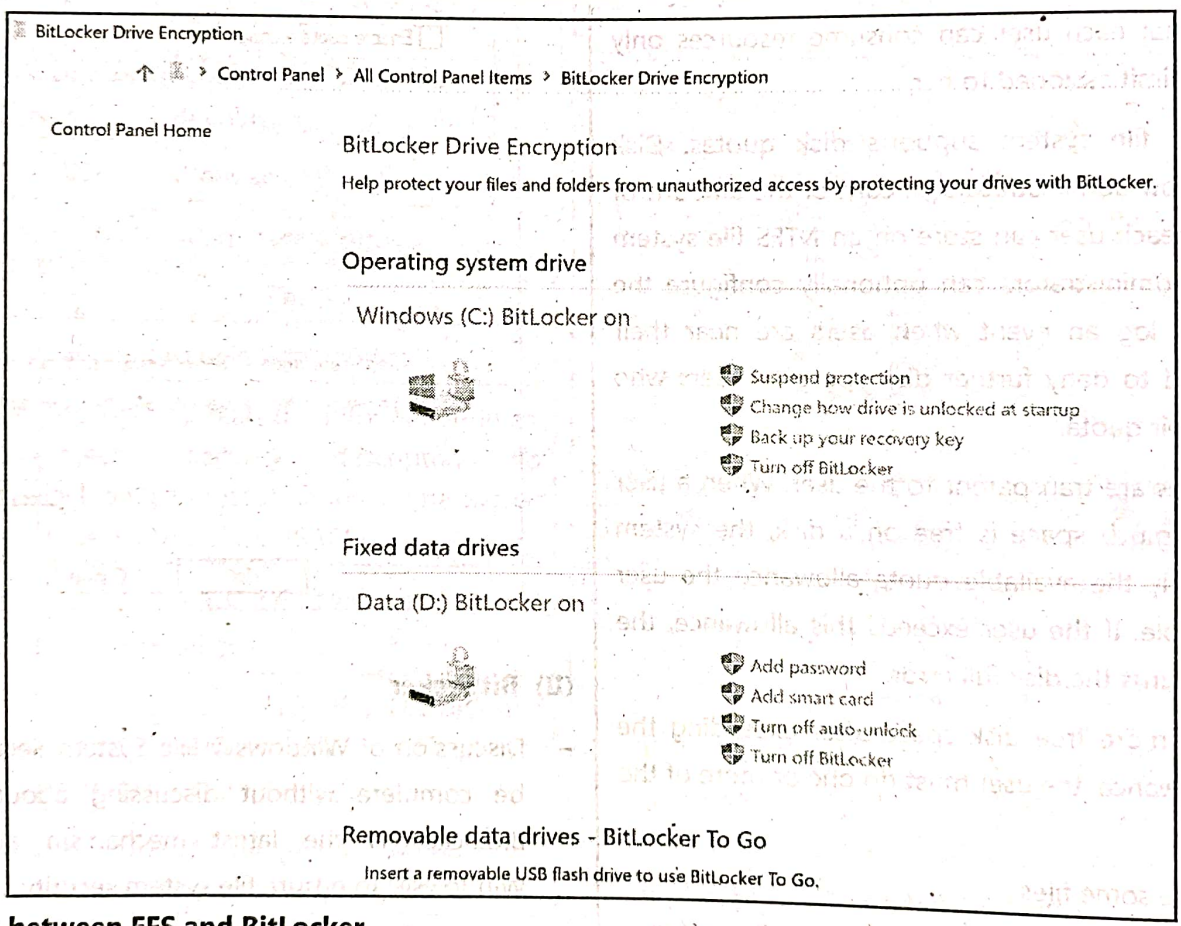
- Discussion of Windows® File System Security cannot be complete without discussing about BitLocker. BitLocker is the latest mechanism available on Windows® to ensure file system security.

✍ ***Definition :*** *BitLocker provides Full Volume Encryption (FVE) for operating system volumes, as well as for fixed and removable data volumes.*

- Unlike EFS that provides per file per user encryption, BitLocker encrypts the entire disk volume. It is not dependent on the user or one file over another. BitLocker is enabled at a disk drive level and encrypts

the entire disk drive. By default, it uses the AES encryption algorithm in Cipher Block Chaining (CBC) or XTS mode with either a 128-bit or 256-bit key.

- The encryption key can be stored on
  (i) TPM (Trusted Platform Module) chip on the motherboard
  (ii) USB drive
- Optionally, you can also set BitLocker to require a PIN or a password to start the boot process.
- One thing to specially note about BitLocker is that BitLocker does not encrypt and decrypt the entire drive when reading and writing data. BitLocker encryption and decryption process works per sector on the drive. The encrypted sectors in the BitLocker-protected drive are decrypted only as they are requested from system read operations. Blocks that are written to the drive are encrypted before the system writes them to the physical disk. No unencrypted data is ever stored on a BitLocker-protected drive.
- If your system meets the required license, hardware and other dependencies, you can turn on BitLocker by right clicking on a drive and choosing to Turn on BitLocker. You can also see status of your drives by going to BitLocker status as shown the following snapshot.



Comparison between EFS and BitLocker

Table 5.3.7

Sr. No.	Comparison Attribute	EFS	BitLocker
1.	Encryption	Per file Per user	Entire disk, Per Sector
2.	Protection Keys	User dependent	System Dependent
3.	Specialized hardware	Not required	Required
4.	Administrator role	None, user can enable / disable	Only administrator can enable / disable

5.3.2(B) File System Security on Linux

- ext4 is the latest file system on Linux. Unlike Windows®, Linux capabilities are grown and developed by open source community instead of just one company. The capabilities are built in the Linux kernel and then extended to other components of the Linux OS.
- Linux provides the following options for file system security as shown in Fig 5.3.9.

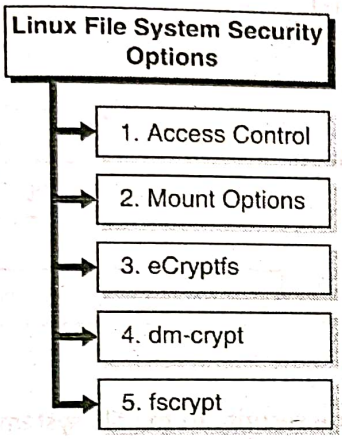


Fig. 5.3.9 : Linux file system security options

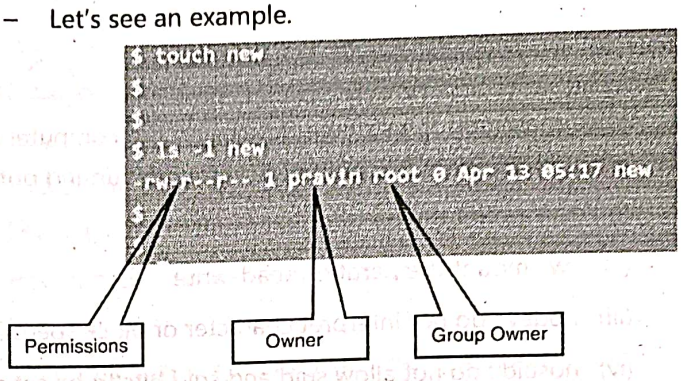
1. Access Control

- Like a typical system based on DAC, Linux provides access control for files and folders.
- Each file is assigned
  - (i) An owner and a group-owner
  - (ii) Set of permissions for owner, group and rest of the world
- Permissions are set as per the following Table 5.3.8.

Table 5.3.8

Permission	Alphabetic representation	Numeric Representation
Read	r	4
Write	w	2
Execute	x	1
No Permission	-	0

- So, if a file is to be given read and write permissions, you can either specify *r* and *w* in the command or provide the combined numeric value (4+2 = 6). You can use ls -l <filename> to see the file permissions. You can use chmod to assign file permissions.
- Let's see an example.



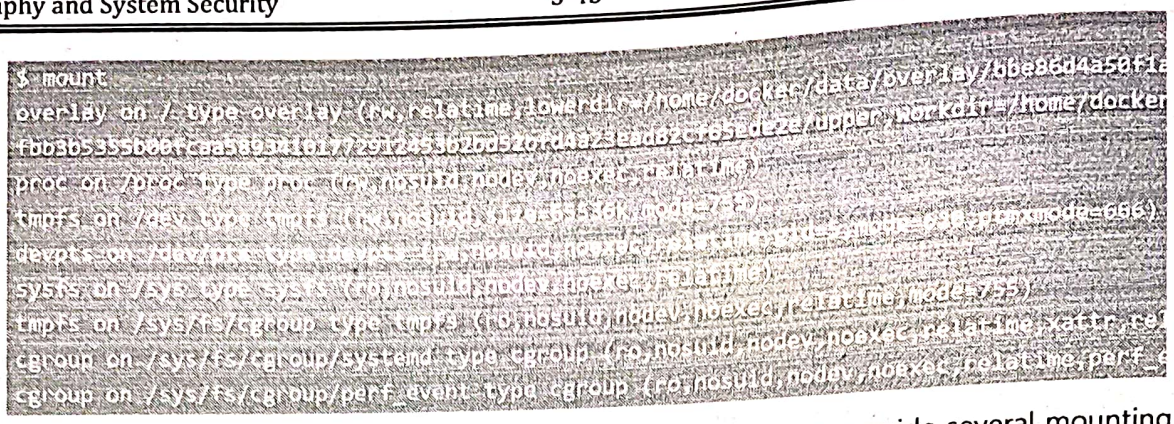
In the provided example:

- The file name is new
- Owner of the file new is the user pravin
- Group-owner of the file new is the group root
- Permissions of the file new are
  - o rw- : user pravin has read and write permission on the file new but no execute permission
  - o r-- : group members of the group root have read permission on the file new
  - o r-- : rest of the world (or others) has read permission on the file new
- Similarly each file and directory on the system has ownership information and permissions defined. Various entities must have adequate permissions to gain access to the file and perform a required task.

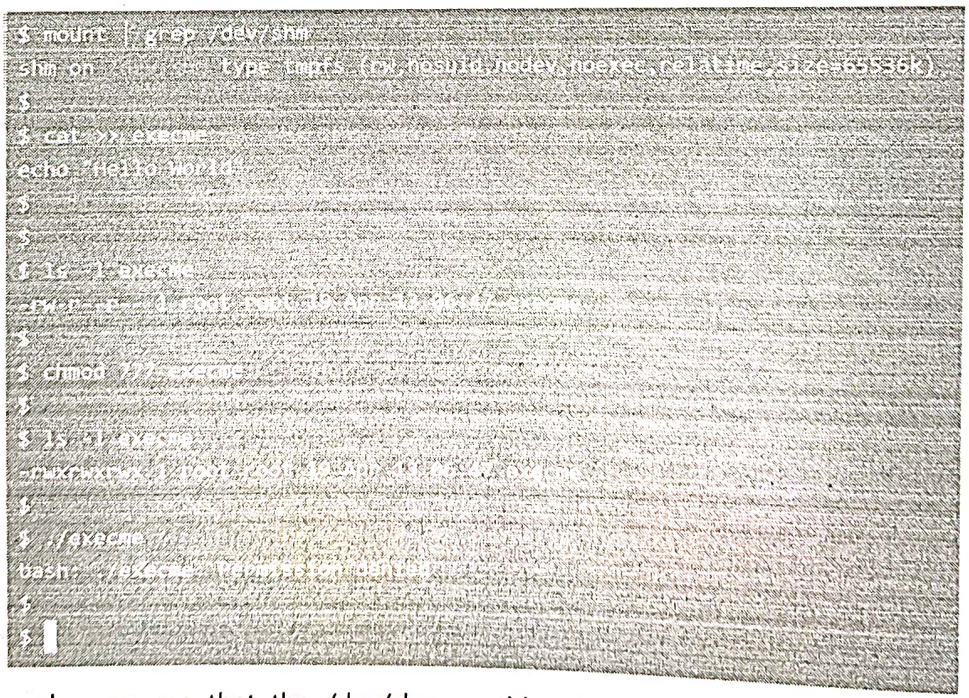
2. Mount Options

- Various disk partitions are mounted (mapped) to various directories.

✍ ***Definition :*** *Mounting is a process by which the operating system makes files and directories on a storage device (such as hard drive, CD-ROM, or network share) available for users to access via the computer's file system.*



- While mounting the storage devices on the computer's file system, you can provide several mounting options that provide security. A few examples of such mounting options are as following.
  - (i) ro : mount the partition read-only
  - (ii) rw : mount the partition read-write
  - (iii) nodev : do not interpret character or block special devices on the file system
  - (iv) nosuid : do not allow suid and sgid bits to be set on files
  - (v) noexec : do not allow execution of binary files from this partition
  - (vi) context : set security context (label) for MAC based systems
  - (vii) nouser : permit only the root user to mount / unmount the partition
- Often the partitions are mounted using the given mount options to provide security to the file system. For example, /dev/shm partition is mounted with the noexec option so that no one can copy a random executable binary file on the system and run it. If the mount option is set, even if the file has executable permission, it would be not executed.



- In the given example, you see that the /dev/shm partition is mounted with the noexec option. I create an executable file execme and give it executable permission for everyone. When I try executing it, it gives me Permission denied error. So, even if I have executable permission set on the file execme, I am not allowed to execute it because of the noexec mount option using which /dev/shm partition was mounted during the system boot process.

3. eCryptfs

✍ ***Definition :*** *eCryptfs is a stacked cryptographic file system.*

- The eCryptfs takes a stacking approach in which eCryptfs sits on top of another file system (say ext4). It provides a seamless translation layer that takes the unencrypted files, encrypts them and remounts them in an encrypted file system.

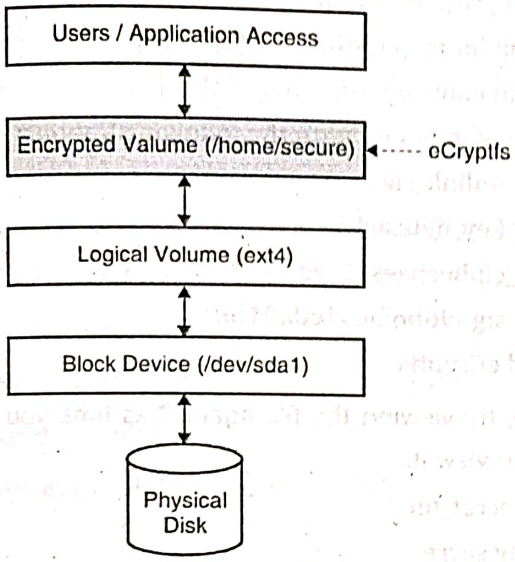


Fig. 5.3.10

- So, you can create encrypted file systems on top of whatever file system is in use locally, or even over a network-mounted file system. It works at the individual file or directory level. Rather than encrypting the entire file system, eCryptfs works on each file individually. Different files can be encrypted in different ways. The metadata required for decryption is placed in the first block of the file itself. Storing metadata with the file provides flexibility in backing up, copying, or even moving the file to another system without losing the metadata needed to decrypt the file in the future.
- It supports the following ciphers and key sizes.

Ciphers	Key Sizes
AES	128 – 256
Blowfish	128 – 448
3DES	192
Twofish	128 – 256
Cast6	128 – 256
Cast5	40 - 128

- Let's see an example of how it works. The following example is on the Ubuntu Linux OS.

Install eCryptfs

apt-get -y install ecryptfs-utils

Let's encrypt a directory by name myfolder.

sudo mount -t ecryptfs /myfolder /myfolder

Once you execute that command you are prompted to enter a passphrase (a password of your choice) and choose other encryption parameters such as cipher and key size.

Passphrase: Enter your passphrase here

Select cipher:

1) aes: blocksize = 16; min keysize = 16; max keysize = 32

2) blowfish: blocksize = 8; min keysize = 16; max keysize = 56

3) des3_ede: blocksize = 8; min keysize = 24; max keysize = 24

4) twofish: blocksize = 16; min keysize = 16; max keysize = 32

5) cast6: blocksize = 16; min keysize = 16; max keysize = 32

6) cast5: blocksize = 8; min keysize = 5; max keysize = 16

Selection [aes]: Press Enter

Select key bytes:

1) 16

2) 32

3) 24

Selection [16]: Press Enter

Enable plaintext passthrough (y/n) [n]: n

Enable filename encryption (y/n) [n]: n

Attempting to mount with the following options:

ecryptfs_unlink_sigs

ecryptfs_key_bytes=16

ecryptfs_cipher=aes

ecryptfs_sig=96b6fac91e0a01b8

Mounted eCryptfs

The my folder directory is then encrypted and mounted. You can verify it by issuing the mount command.

```
sudo mount | grep myfolder
/myfolder on /myfolder type ecryptfs
(rw,relatime,ecryptfs_sig=96b6fac91e0a01b8,ecryptfs_c
ipher=aes,ecryptfs_key_bytes=16,ecryptfs_unlink_sigs)
```

You can now add files in the encrypted my folder directory.

```
cat >> my_secret_file
This is my secret

cat my_secret_file
This is my secret
```

To check if encryption is working, unmount the myfolder directory.

```
sudo umount /myfolder
```

Now, if you try to see the file content, it would show garbage.

```
cat my_secret_file
r6AZy3._T`ã0FfỸ%O{@[ь8wLRosr&]^
XoeA"m?SVb&/Nrgθo?&[sj!
"lL
CЯ5)+H^rA2aKxf.a+}LiA|gSOj2y_x~Ma^p.PmPUvdjv34
c5-F\hFQ½CTLB#OI+5.WXsDlb"
f50)> AJa7$~4V!R;udP]ƹABSyT!qg
D;fY(&4!aX@"JyɬPwvJ"h}B<nS>e
)@[K~w39PK^ j~p"G'eQEK&3Ywe,Þ,AVjʒ!yUoX
·@\:!{mQfó'wţap]4}% αu[0M#>S麗g9t_A0k=ʒ;
wHGZN$V2"HI-4olhEoK?]9sIr0J:{uTJ8籲D ·+
nı7CI&iCELŸāỹK⌒n)-]}C
/ƒ46}u4bo\B4Iȯ+1hdH2:)>Vj{#B8<p6)3XHwluDkQ{
```

Remount the myfolder directory by providing your previously chosen passphrase and parameters.

```
sudo mount -t ecryptfs /myfolder /myfolder
Passphrase:  Enter your passphrase here
Select cipher:
1) aes: blocksize = 16; min keysize = 16; max keysize
= 32
2) blowfish: blocksize = 8; min keysize = 16; max keysize
= 56
3) des3_ede: blocksize = 8; min keysize = 24; max
keysize = 24
4) twofish: blocksize = 16; min keysize = 16; max keysize
= 32
5) cast6: blocksize = 16; min keysize = 16; max keysize
= 32
6) cast5: blocksize = 8; min keysize = 5; max keysize = 16
Selection [aes]:  Press Enter
Select key bytes:
1) 16
2) 32
3) 24
Selection [16]:  Press Enter
Enable plaintext passthrough (y/n) [n]: n
Enable filename encryption (y/n) [n]: n
Attempting to mount with the following options:
ecryptfs_unlink_sigs
ecryptfs_key_bytes=16
ecryptfs_cipher=aes
ecryptfs_sig=96b6fac91e0a01b8
Mounted eCryptfs
```

Now, try viewing the file again. This time you should be able to view it.

```
cat my_secret_file
This is my secret
```

4. dm-crypt

- Device Mapper provides a generic way to create virtual layers of block devices that can do different things on top of real block devices.

✍ ***Definition :*** *dm-crypt (device mapper crypt) provides block device encryption (full volume encryption).*

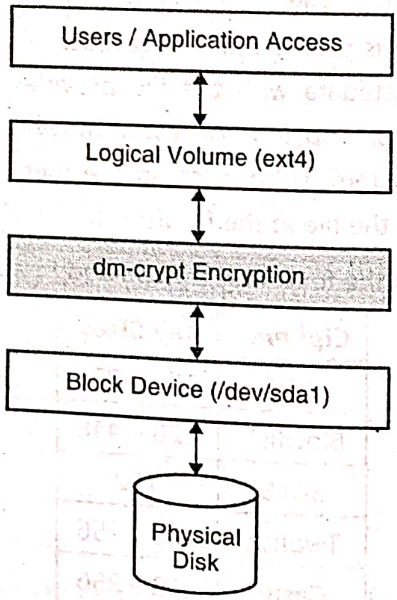


Fig. 5.3.11

- dm-crypt supports the following ciphers with their respective block and key sizes.
  (i) AES
  (ii) Twofish
  (iii) Serpent
  (iv) Cast5
  (v) Cast6
- Setting up dm-crypt directly could be complex for many users. Hence, LUKS (Linux Unified Key Setup) is the preferred way to set up disk encryption with dm-crypt using the cryptsetup utility.
- Let's see an example of how it works. The following example is on the Ubuntu Linux OS.
- Install cryptsetup.

```
sudo apt-get install cryptsetup
```

- Format the partition.

```
sudo cryptsetup -y luksFormat /dev/sda1

WARNING!
========
This will overwrite data on /dev/sda1 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter passphrase:
Verify passphrase:
```

- Map the encrypted partition to a logical volume.

```
sudo cryptsetup luksOpen /dev/sda1 secure

Enter passphrase for /dev/sda1:
```

- Create a file system on the logical volume

```
sudo mkfs.ext4 /dev/mapper/secure
```

- Create a mount point to mount the file system created on the logical volume

```
sudo mkdir /home/secure
sudo mount /dev/mapper/secure /home/secure
```

- Now, whatever you write to the `/home/secure` directory would remain encrypted.

5. fscrypt

- fscrypt is the latest way using which files can be encrypted on a Linux OS.
- Performance suffers in eCryptfs as a result of the stacked nature of the filesystem. Imagine a system running eCryptfs over ext4.If a process wants to read a page from an encrypted file, eCryptfs must first instruct ext4 to read that page into the page cache. It then decrypts the data into another page-cache page. The extra copies of the data can consume a lot of memory and slow things down unnecessarily.
- Hence, encryption support directly into ext4 (or at file system level) can eliminate performance issues. You can choose either of the following for protecting your file data.
  (i) Your login password
  (ii) Your chosen passphrase
  (iii) A raw key file

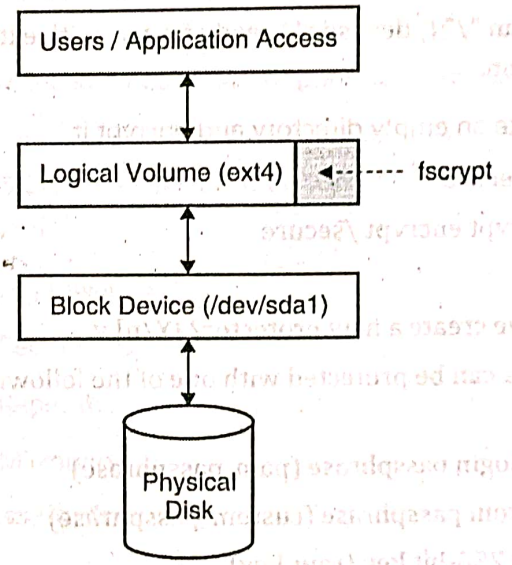


Fig. 5.3.12

- fscrypt allows one encryption mode to be specified for file contents and one encryption mode to be specified for filenames. Different directories are permitted to use different encryption modes. Currently, the following pairs of encryption modes are supported :
  (i) AES-256-XTS for contents and AES-256-CTS-CBC for filenames

(ii) AES-128-CBC for contents and AES-128-CTS-CBC for filenames

- To start using fscrypt, you need to first create an empty directory and encrypt it. Then you can place files in that directory to encrypt them.
- Let's see an example of how it works. The following example is on the Ubuntu Linux OS.
- Install fscrypt.

```
sudo apt-get install libpam-fscrypt
```

Setup fscrypt.

```
sudo fscrypt setup
Create "/etc/fscrypt.conf"? [Y/n] y
Customizing passphrase hashing difficulty for this system...
Created global config file at "/etc/fscrypt.conf".
```

- Initialize the file system to use fscrypt.

```
sudo fscrypt setup /
Metadata directories created at "/.fscrypt".
Filesystem "/" (/dev/sda1) ready for use with ext4 encryption.
```

- Create an empty directory and encrypt it.

```
mkdir /secure
sudo fscrypt encrypt /secure

Should we create a new protector? [Y/n] y
Your data can be protected with one of the following sources:
1 - Your login passphrase (pam_passphrase)
2 - A custom passphrase (custom_passphrase)
3 - A raw 256-bit key (raw_key)
Enter the source number for the new protector
[2 - custom_passphrase]: 2
Enter a name for the new protector: My Secret
Enter custom passphrase for protector "My Secret":
Confirm passphrase:
"/secure" is now encrypted, unlocked, and ready for use.
```

- You can then place your files inside the encrypted directory to keep them protected.

Comparison between eCryptfs, dm-crypt and fscrypt

Table 5.3.9

Sr. No.	Comparison Attribute	eCryptfs	dm-crypt	fscrypt
1.	Stack Partition	Yes	Yes	No
2.	Actual file system lies	Below	Above	Same level
3.	Encryption at	File level	Block device level	File level
4.	Formatting	Not required	Required	Not required
5.	Encryption per	User	Device	User

5.4 Database Security

✍ ***Definition :*** *A database is a repository of information organised in a logical meaningful way.*

- Database is used in almost all applications. Keeping it secure is a prime concern of organisations to ensure that the sensitive data is not exposed, altered or destroyed.
- Let us begin with some of the high-level database security requirements.

Note : This section assumes that you have sufficient understanding of basic database concepts. The focus of this section is purely on security aspects of database.

5.4.1 Database Security Requirements

- The database security requirements can be categorised into the following.

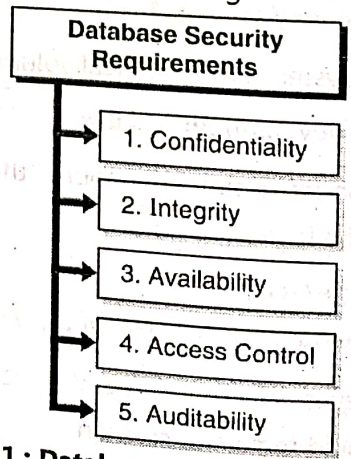


Fig. 5.4.1 : Database security requirements

1. Confidentiality

✍ ***Definition :*** *Confidentiality of data ensures that it is not exposed to entities that do not have sufficient authorisation to view the data.*

- For example, who should be allowed to see your account details online? The banking application must ensure that it authenticates you and then lets you view only the account data that is tied to your account. Even if you are successfully authenticated to the banking application, you cannot see account details of any other user.
- There are several mechanisms using which confidentiality of data can be preserved. Some of them are as following.
  (i) Data encryption
  (ii) Database views
  (iii) Access control and permissions

2. Integrity

- Integrity of database is perhaps the most crucial part of database security.

✍ ***Definition :*** *Integrity of data ensures that the data stored in the database is not altered in any unauthorised way and its structure, value and meaning is preserved as it was desired to be.*

- The database should be protected from any damage arising out of physical, logical, structural or programmable aspects. For example, suppose your account has Rs.10,000 and you withdraw Rs.2,000 from ATM. As soon as the cash is dispensed, the power goes off. What would happen? If the bank's database is not updated with your transaction, it would still show that you hold Rs.10,000 in your account and bank would be in loss. It is crucial to ensure that the database records are accurate.
- There are several mechanisms using which integrity of data can be preserved. Some of them are as following.
  (i) Save points
  (ii) Two-phase commits
  (iii) Backup and Restore
  (iv) Error detection

3. Availability

- The database is crucial for delivering the data relevant to the user or the application.

✍ ***Definition :*** *Availability of database ensures that the data contained in the database is usable when someone is required to use it.*

- If for some reason, the database is not available, the user would not be able to use the application. For example, you want to urgently make a payment for your online purchase, but the banking application is not available because there is some database maintenance activity going on. What do you do? Don't you feel frustrated?
- Similarly, if there are thousands of users requesting data at the same time, the database has to prioritize one request over the another or be capable of serving multiple requests in parallel. If the database response is slow, it could again lead to poor user experience.
- There are several mechanisms using which availability of database can be ensured. Some of them are as following.
  (i) High Availability
  (ii) Redundancy
  (iii) Replication
  (iv) Mirroring

4. Access Control

- By now, you understand the importance of access control.

✍ ***Definition :*** *Access Control in database ensures that only the entities authorised to use the required data from the database are allowed access.*

- Access Control is also one of the mechanisms to provide confidentiality and integrity of data in the database. Like OS, there are different users with different capabilities on database server. There could

be administrators and general users. There could also be roles that define a pre-selected set of capabilities that someone assuming the role would have.

- For example, can the administrator of the banking database server increase the account balance amount for anyone? Can she delete the transaction records and rollback the database to a previous state? So, even though someone might be an administrator on the database, it does not mean that she has unrestricted rights over the data contained in the database. For example, the database administrator's job might be to just ensure that the database server is up and running, it is properly patched, and the desired network access is permitted. She may not have any rights on the data at all. How do you ensure this separation? Let's answer it later in the chapter. Access control in database could be little tricky.
- There are several mechanisms using which access control on database can be ensured. Some of them are as following.
  (i) Authentication and Authorization
  (ii) Roles and Permissions
  (iii) Data views
  (iv) Firewalls

5. Auditability

Auditability provides the capability of tracing and evidences.

✍ ***Definition :*** *Auditability of database ensures that any access to it is appropriately logged and reported.*

- Audit trails are evidences and records of who has done what on the database server. For example, if you find that a table is missing from the database, then you can check the audit logs and find out who has deleted the table and when. Using audit logs, you can hold someone accountable for her actions.
- Additionally, audit logs also help to maintain the integrity of the database by logging all the actions that are being taken on the database. You can periodically review the logs and ensure that only the desired actions are being carried out. If you find any discrepancy in what you expected and what is being done on the database system, you can take the appropriate actions and restore the database integrity.

5.4.2 Reliability and Integrity

As you understand, database up time and accuracy are probably the most crucial aspects of delivering any application. Hence, database reliability and database integrity require special and careful attention. Let's begin with defining and understanding those terms and then diving deeper into understanding them.

✍ ***Definition :*** *The term reliability refers to the ability of a hardware or a software component to consistently perform according to its specifications.*

With respect to database,

✍ ***Definition :*** *Database reliability ensures that the database server can perform at the desired level consistently for a long time without requiring regular downtime and maintenance activities.*

Similarly, as you learnt earlier,

✍ ***Definition :*** *Integrity of data ensures that the data stored in the database is not altered in any unauthorised way and its structure, value and meaning is preserved as it was desired to be.*

5.4.2(A) Database Reliability and Integrity Requirements

From security perspective, database reliability and integrity requirements can be categorised as following.

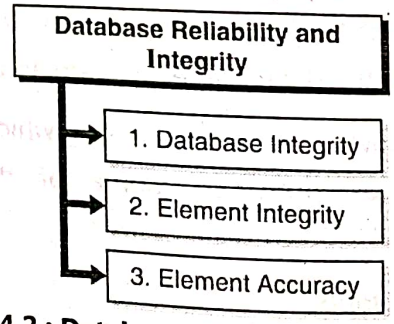


Fig. 5.4.2 : Database reliability and integrity

1. Database Integrity

Database integrity requires that the database maintains both the physical as well as the logical integrity. The OS, on which the database server is running, must ensure that the files related to the database are not accessed by unauthorised entities and also, they are free from disk errors and provided general OS level protection and correction mechanisms. Additionally, the logical structure of database, tables within it, permissions, roles, etc. must all be sufficiently preserved to avoid any integrity errors.

2. Element Integrity

Element integrity requires that the values of various elements in the database tables are changed, updated or modified by only authorised entities. It is not altered outside the application (by directly connecting to the database) or by following random and unapproved processes. Setting up adequate access control mechanisms could ensure element integrity.

3. Element Accuracy

Element accuracy ensures that only correct and desired values are written to elements in the table. For example, if the balance field in the table can take numeric values, only numeric values should be allowed and not characters. Database itself enforces several integrity checks and constraints to ensure that elements are accurate.

(i) **Domain Constraints :** Domain constraints specify a valid set of values for an attribute. For example, age can only be numeric, and it cannot take character values.

(ii) **Entity Integrity Constraints :** The entity integrity constraint states that the primary key value cannot be null for any record.

(iii) **Referential Integrity Constraints :** Referential integrity ensures that all foreign keys have an existing and referenceable primary key.

(iv) **Primary Key Constraints :** Primary key in a table must be unique.

5.4.2(B) Database Integrity Protection Mechanism

Let's discuss some of the database integrity protection mechanisms.

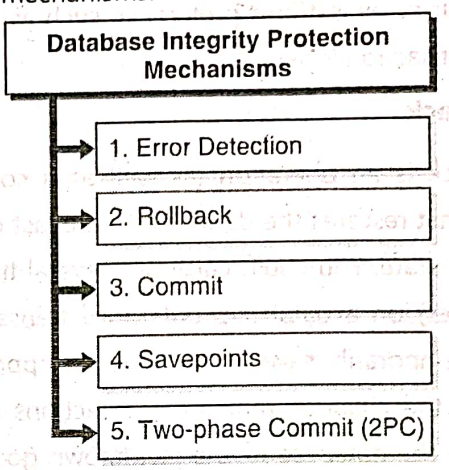


Fig. 5.4.3 : Database integrity protection mechanisms

1. Error Detection

There could be several errors that a database system can encounter.

Some of them could be:

(i) **Application errors :** Application wrongly updated the database

(ii) **Database internal errors :** Database had some data processing errors internally

(iii) **OS errors :** OS crashed during data processing

(iv) **Hardware errors :** Some hardware hosting the database failed during data processing

(v) **Network errors :** Database lost connectivity during data processing

(vi) **Disaster :** A natural disaster such as flood completely wiped off all data on the database server.

– Error detection mechanism such as Cyclic Redundancy Codes (CRC), hash values, parity bits etc. can be computed for crucial data. Such codes can be computed at the field level, table level or for the entire database. Before any read or write operation is carried out, these codes can be checked to ensure

that the element value is accurate and has not been manipulated or altered from the last known good value (based on the codes).

- If and when errors are detected, the last known good value can be restored either from the backup or any other preferred mechanism of correcting data based on your organisation's preference such as processing the transaction logs.

2. Rollback

- Rollback is an operation (as well as a command in SQL) that restores the database to the last committed (good) state. You could carry out several transactions (updates) on a database, but those transactions are only temporarily saved. You have an opportunity to cancel the changes that the transactions made and restore the database to the last known good state. If you are sure about the transaction, you can commit (permanently save) the transaction changes into the database.

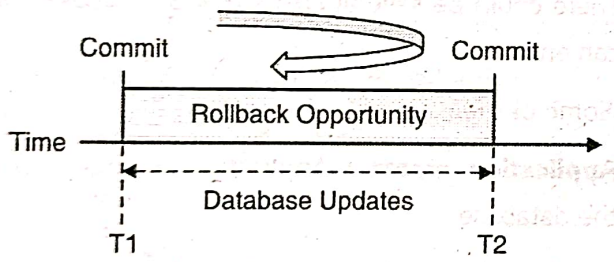


Fig. 5.4.4

- Rollback is immensely helpful if the transaction fails due to any errors or if the user herself cancels the transaction (for example, deciding to not purchase after putting thing in cart and then backing out from the payment page). Rollback helps to maintain the database integrity by cancelling out the undesired changes and restoring the good state.

Let's take an example.

```
INSERT INTO product VALUES(1, 'Biscuit');
INSERT INTO product VALUES(2, 'Tea');
INSERT INTO product VALUES(3, 'Coffee');
INSERT INTO product VALUES(4, 'Chocolate');
INSERT INTO product VALUES(5, 'Cake');

COMMIT;

SELECT * FROM product;

UPDATE product SET name = 'Ketchup' WHERE id = '3';

SELECT * FROM product;

ROLLBACK;

SELECT * FROM product;
```

- The output from the first SELECT statement would be as following.

Id	Name
1	Biscuit
2	Tea
3	Coffee
4	Chocolate
5	Cake

- The database state is committed with these values.
- Now, you make a change to the database entry. The output from the second SELECT statement would be as following.

Id	Name
1	Biscuit
2	Tea
3	Ketchup
4	Chocolate
5	Cake

- If this change is undesired, you can restore the previous state (last committed state) by issuing the ROLLBACK command. The third SELECT statement shows that the last good state is restored.

Id	Name
1	Biscuit
2	Tea
3	Coffee
4	Chocolate
5	Cake

3. Commit

- Commit is an operation (as well as a command in SQL) that permanently saves the changes made to the database. The time at which the changes are committed becomes a reference point for a known good state of the database. Any transactional changes (INSERT, UPDATE, DELETE) made after the commit until the next commit can be rolled back if undesired.

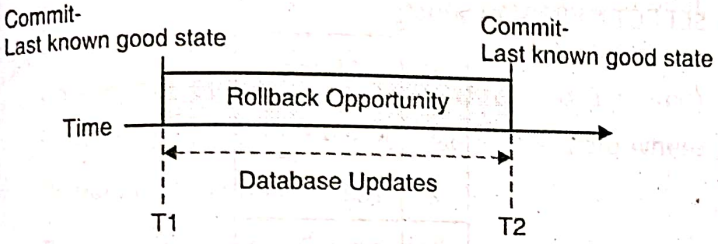


Fig. 5.4.5

- Commit ensures that the database is in good state and preserves the database state until a next good state is reached. This way the database integrity can be assured even if there are errors or undesired changes to the database. Once the changes are committed, the new values are available to all the applications and users.
- Let's take an example.

```
INSERT INTO product VALUES(1, 'Biscuit');
INSERT INTO product VALUES(2, 'Tea');
INSERT INTO product VALUES(3, 'Coffee');
INSERT INTO product VALUES(4, 'Chocolate');
INSERT INTO product VALUES(5, 'Cake');

COMMIT;

SELECT * FROM product;

UPDATE product SET name = 'Ketchup' WHERE id = '3';

SELECT * FROM product;

COMMIT;

SELECT * FROM product;
```

- The output from the first SELECT statement would be as following.

Id	Name
1	Biscuit
2	Tea
3	Coffee
4	Chocolate
5	Cake

- The database state is committed with these values.
- Now, you make a change to the database entry. The output from the second SELECT statement would be as following.

Id	Name
1	Biscuit
2	Tea
3	Ketchup
4	Chocolate
5	Cake

- If the change looks good, you can COMMIT and save the change. The third SELECT statement then shows the changed value as per the last COMMIT.

Id	Name
1	Biscuit
2	Tea
3	Ketchup
4	Chocolate
5	Cake

4. Save points

- Save points let you keep reference placeholders for your changes so that you can go back to a particular state if desired.

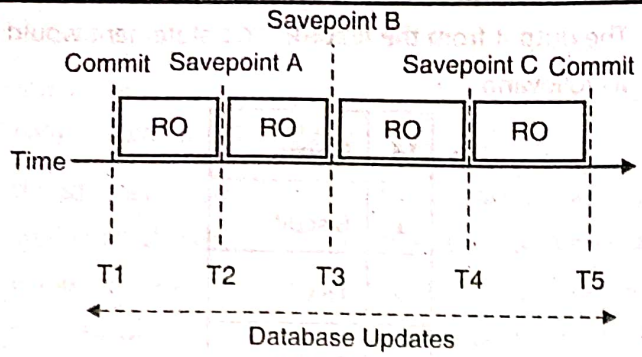


Fig. 5.4.6

– So, instead of one big rollback opportunity between two commits, save points provide several smaller rollback opportunities. This is very useful if you want to selectively rollback the changes instead of all the changes.

Let's see an example.

```
INSERT INTO product VALUES(1, 'Biscuit');
INSERT INTO product VALUES(2, 'Tea');
INSERT INTO product VALUES(3, 'Coffee');
INSERT INTO product VALUES(4, 'Chocolate');
INSERT INTO product VALUES(5, 'Cake');

COMMIT;

SELECT * FROM product;

UPDATE product SET name = 'Ketchup' WHERE id = '3';
SAVEPOINT A;

UPDATE product SET name = 'Juice' WHERE id = '3';
SAVEPOINT B;

UPDATE product SET name = 'Chips' WHERE id = '3';
SAVEPOINT C;
```

– In this example, you have created three save points. Each save point can be considered as a reference placeholder to which you can take the database state to.

For example,

```
ROLLBACK TO SAVEPOINT A;
SELECT * FROM product;
```

Id	Name
1	Biscuit
2	Tea
3	Ketchup
4	Chocolate
5	Cake

```
ROLLBACK TO SAVEPOINT B;
SELECT * FROM product;
```

Id	Name
1	Biscuit
2	Tea
3	Juice
4	Chocolate
5	Cake

```
ROLLBACK TO SAVEPOINT C;
SELECT * FROM product;
```

Id	Name
1	Biscuit
2	Tea
3	Chips
4	Chocolate
5	Cake

– Now, if you `ROLLBACK TO SAVEPOINT B` and then issue `COMMIT`, the database would save that change permanently.

```
ROLLBACK TO SAVEPOINT B;
COMMIT;
SELECT * FROM product;
```

Id	Name
1	Biscuit
2	Tea
3	Juice
4	Chocolate
5	Cake

5. Two-phase Commit (2PC)

- When you carry out a transaction, often multiple databases are required to be updated.

✍ ***Definition :*** *Two-phase Commit (2PC) is a mechanism to ensure that in a transaction either all updates are carried out successfully or none.*

- There is a coordinator that manages the transaction and commit process. The database (workers) follow instructions from the coordinator and either commit or rollback the transaction. The core objective is to ensure that all databases are synchronized (updated) with the transaction. No database is in a state where its data is inaccurate.
- The two phases of 2PC are as following.

(i) **Phase 1 (Vote or Pre-commit phase) :** The coordinator sends a request to all workers asking if each one can agree to commit on the transaction. The workers either respond Yes or No.

(ii) **Phase 2 (Commit or Rollback Phase) :** Based on the response from each worker

(a) If all workers have responded as "Yes" : The coordinator issues a "Commit" instruction to all workers. All workers need to commit the changes and send acknowledgement to the coordinator.

(b) If any worker has responded as "No" : The coordinator issues a "Rollback" instruction to all workers. All workers need to rollback the changes and send acknowledgement to the coordinator.

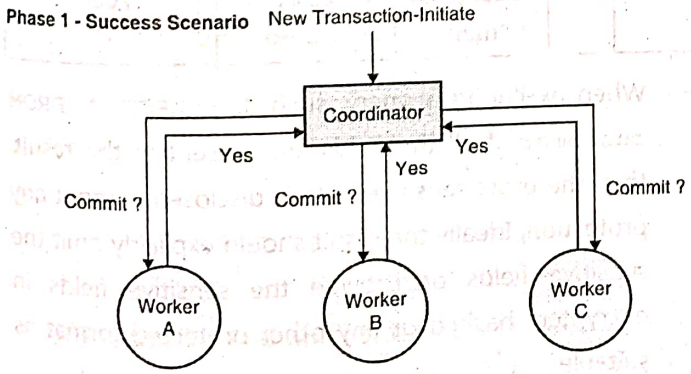


Fig. 5.4.7

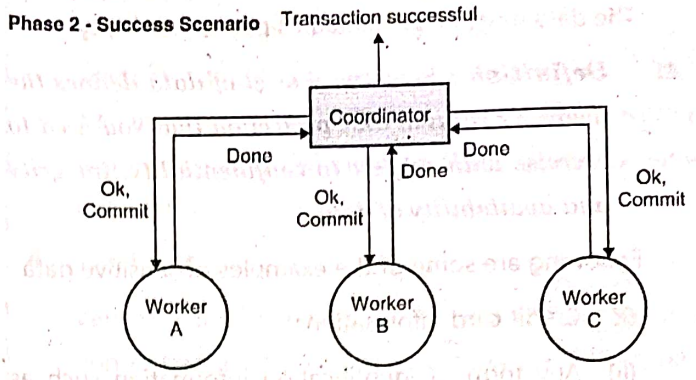


Fig. 5.4.8

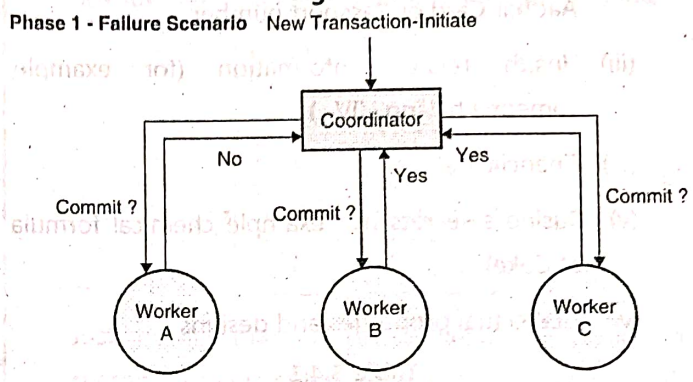


Fig. 5.4.9

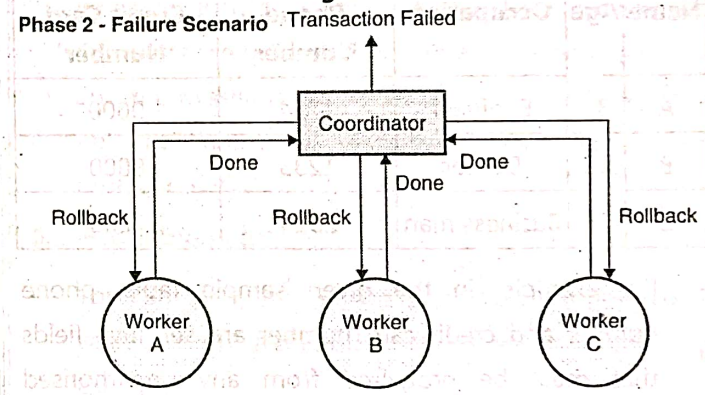


Fig. 5.4.10

- This ensures that a transaction is considered complete and successful only if all the databases are updated and are consistent. If any of the database fails to update or be consistent, the transaction fails and any changes that occurred are rolled back.

5.4.3 Sensitive Data

✍ ***Definition :*** *Sensitive data is the information whose loss, misuse, unauthorised access or modification could adversely affect the national, organisational, societal or personal interest.*

The data must be protected as per its sensitivity.

✍ ***Definition :*** *Sensitivity level of data defines the degree of caution and protection that you need to exercise with respect to confidentiality, integrity and availability of data.*

Following are some of the examples of sensitive data:

(i) Credit card information

(ii) Any form of identification information such as Aadhar Card or Passport number

(iii) Health related information (for example, someone having HIV+)

(iv) Financial details

(v) Business secrets (for example chemical formula of Coke)

(vi) Intellectual properties and designs

Table 5.4.1

Name	Age	Occupation	Phone Number	Credit Card Number
A	32	Engineer	1234	9000
B	33	Doctor	1235	8000
C	34	Business man	2314	7000

- For example, in the given sample table, phone number and credit card number are sensitive fields that must be protected from any unauthorised disclosure. A phone number can be used to identify a person whereas credit card information can be used to commit financial frauds. Both require protection such that only the authorised application can use them as and when desired.
- Databases often store sensitive information along with non-sensitive information. If all information in a database is non-sensitive or all information in the database is sensitive, it is comparatively easier to either open up or lock down the entire database. When the database has mixed type of information (sensitive as well as non-sensitive) it becomes tricky to safeguard the sensitive information from disclosure – either directly or indirectly.
- It is important to understand the types of probable disclosures around sensitive data. Later you would see that how these types of disclosures can be made using various inference attack techniques.

5.4.3(A) Types of Disclosure of Sensitive Data

There could be several types of disclosure of sensitive data as shown in Fig. 5.4.11.

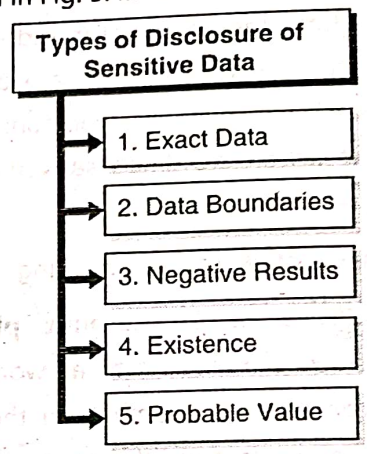


Fig. 5.4.11 : Types of disclosure of sensitive data

1. Exact Data

- Disclosure of exact sensitive data is most damaging. This could be because of errors or because of lack of adequate security controls around the protection of sensitive data. For example, consider the following sample table named `customer`.

Name	Age	Occupation	Phone Number	Credit Card Number
A	32	Engineer	1234	9000
B	33	Doctor	1235	8000
C	34	Business man	2314	7000

- When executing a query, such as `SELECT * FROM customer`, if all the fields are present in the result, then the exact sensitive data is disclosed without any protection. Ideally the result should explicitly omit the sensitive fields or provide the sensitive fields in encrypted, hashed or any other protected format as suitable.

2. Data Boundaries

- In this type of disclosure, the actual data is not disclosed but the range of data could be disclosed. The range itself could give you a fair idea of how the distribution of various sensitive values could be in the actual table. Let's consider the following example.

Name	Income Tax	Income
A	1,00,000	10,00,000
B	2,00,000	20,00,000
C	3,00,000	30,00,000
D	5,00,000	50,00,000

- Consider that the example table consists of records of employees and their respective income tax and income field is sensitive and is not disclosed to everyone. If you know that the minimum income tax paid by an employee is 1,00,000 and the maximum income tax paid by an employee is 5,00,000, you can fairly estimate the income range of employees of a company. Consider a flat tax of 10%, it would mean that the lowest paid employee earns around 10,00,000 whereas the highest paid employee earns around 50,00,000. This salary range disclosure could be sensitive information for a company. The competitor company could structure their employment offers around this information and attract talented employees.

3. Negative Results

Consider the following Table 5.4.2.

Table 5.4.2

Location	Weapon Count
A	0
B	10
C	200
D	50

If you can just find whether the count of weapons in a military location is either 0 or not 0, that itself could be sensitive disclosure. The actual count of weapons does not matter here. A location not having any weapons might be easier to target and attack. Negative result disclosure proves or disproves an assumption and can help to infer (guess) sensitive information.

4. Existence

- Very similar to negative results, just the existence or non-existence of a record or a field could itself be a sensitive information. Consider the following table.

Location	Missiles
A	10
B	10
C	200
D	50

- Just knowing that only the locations *A, B, C* and *D* are present in the missiles table, it could mean that other locations such as *E* and *F* do not have missiles yet. The existence of location information itself is sensitive and the missile count, even though is sensitive, may not be the only sensitive information worth protecting.

5. Probable Value

- You can query the database to know the probability value of an assumption. Consider the following example.

Name	Preference	Political Party
A	Coffee	X
B	Coffee	Y
C	Tea	X
D	Tea	X

- Your goal here could be to determine how many people are with a particular political party. The association of an individual name with a political party is sensitive and protected information but the association of preference and political party is not. Now, you can execute the following queries.

SELECT COUNT(*) FROM people WHERE preference = 'Coffee';

SELECT COUNT(*) FROM people WHERE preference = 'Coffee' AND party = 'X';

SELECT COUNT(*) FROM people WHERE preference = 'Tea;

SELECT COUNT(*) FROM people WHERE preference = 'Tea' AND party = 'X';

- The result of respective queries is 2, 1, 2, 2. So, you can safely assume the following probable values
  - Around 50% of people who prefer coffee are with the party X
  - Majority of people who prefer tea are with the party X
  - There is a 50-50 chance of someone preferring coffee or tea
- Now, assume that there are 1,000 records in the table. Based on the probability values you inferred earlier, the following statements look to be very convincing, isn't it?
  - Around 500 people would prefer coffee
  - Around 200 of coffee drinking people would be with party X
  - Around 500 people would prefer tea
  - Around 400 of tea drinking people would be with part X
  - So, 200 + 400 = 600, means that around 60% of the people are with political party X
- This value of 60% probability of a political party association could be sensitive but could be easily disclosed.

5.4.4 Inference Attacks

Inference refers to the ability to deduce information (guess or logically arrive at a conclusion).

✍ ***Definition :*** *Inference attacks target non-sensitive information to deduce sensitive information.*

The attack analyses several information and combination of facts associated with the sensitive information to make smart guesses about the sensitive information. You have already learnt about the several types of disclosure that are possible for sensitive data in the previous section. Let's now learn about inference attacks.

5.4.4(A) Types of Inference Attacks

At a high level, there are three types of inference attacks that can be done on databases to reveal (or infer) sensitive information.

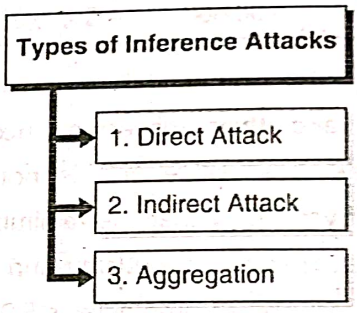


Fig. 5.4.12 : Types of inference attacks

1. Direct Attack

✍ ***Definition :*** *Direct attacks try to reveal the sensitive information by executing direct queries on the database.*

The queries are formed such that it is not-obvious that the attacker is trying to get sensitive information. Let's take an example.

Name	Age	Gender	Role	Promotion
A	32	M	Designer	No
B	33	F	Programmer	Yes
C	34	F	Manager	No
D	31	M	Architect	No
E	35	M	Programmer	Yes
F	34	M	QA	No

SELECT name FROM employee WHERE promotion='Yes';

The database might reject the query because the attacker is trying to access the sensitive field and the employee names, whose promotions are due, must not be revealed.

Let's modify the query little bit.

```
SELECT name FROM employee WHERE (gender = 'M' AND
gender = 'F') OR promotion='Yes' OR Age > 50 OR name = 'X';
```

In this query, the database might assume that the query would result into several records (names – both who are due for promotion and who are not due for promotion) and the list of names would not provide the sensitive information about who is about to get promoted. But that does not actually happen. The query still results into only revealing names who are due for promotion because other conditions in the query evaluate to false.

- A gender cannot be a male and a female at the same time
- There are no employees aged above 50
- Neither there is an employee whose name is X

The sensitive information might be revealed by making the query more complex and assuming that the produced results are not sensitive enough to be protected.

2. Indirect Attack

✍ ***Definition :*** *Indirect inference attacks do not actually get the sensitive data but makes guesses based on facts regarding the data.*

It could be based on statistical approaches that you learnt earlier – sum, count, mean, median, etc. The examples are disclosures such as

- Data boundaries
- Negative results
- Existence and
- Probable values

3. Aggregation

✍ ***Definition :*** *Aggregation is the collection of various facts associated with the data to make the guessed information (inference) more realistic.*

- Note here that unlike other attacks which are purely done on databases, aggregation can be done outside the database. The results from database are correlated with other information that the attacker might have about the data, environment, people or system.
- For example, the employee vacation database might not show the vacation details (from which date to which date the employee was absent from work), but her Facebook account details might have status updates, photos and other comments regarding the vacation. The attacker could then use this information to carry out attacks in the absence of employee.
- Another example could be that a general military clerk may not have information about how many soldiers are getting placed where, but she might have food or shelter arrangement tasks assigned that would indirectly exhibit the number of people who are likely to be stationed at a base.
- Aggregation is increasingly becoming powerful as more and more companies collect your online activities, locations where you go, your purchase choices and other consumer behaviour. The companies then use your data to sell customized services to you or for other purposes and motives.

5.4.4(B) Protection Against Inference Attacks

There are several ways to protect from inference attacks.

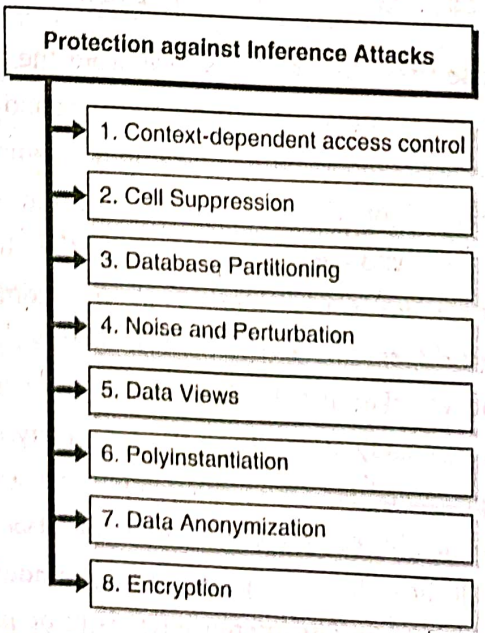


Fig. 5.4.13 : Protection against inference attacks

1. Context-dependent access control

- Databases are content driven, meaning that they respond with the content (records) as queried. No active state of the system is maintained to decide on returning the records based on what has been previously accessed or shared.

✍ ***Definition :*** *Context-dependent access control is a sensitive data protection technique where the previous state of access requests is maintained and the future access decisions are taken based on what the user already knows from the previous access requests.*

- Based on the context (situation), further information access decisions are taken. If the user already knows enough, such that letting her to know more information would lead to inferring or disclosing sensitive data either directly or indirectly, then the further information is not provided to the user.
- Let's take an example.

Name	Age	Gender	Role	Tenure	Promotion
A	32	M	Designer	2 years	No
B	33	F	Programmer	3 years	Yes
C	34	F	Manager	6 months	No
D	31	M	Architect	2 years	No
E	35	M	Programmer	7 years	Yes
F	34	M	QA	1 year	No

- Suppose that the attacker knows from the past data that programmers get a promotion around their 3rd year in the company and 7th year in the company. The attacker might try to first find out the name of employees who are programmers and then try to find out their respective tenures in the company. A context-dependent access control system would not let the attacker run both the queries because the attacker already knows from the first query that who the programmers are. Knowing their respective tenures in the company would let the attacker infer who all are due for promotion. Promotion is a sensitive field and its inference needs to be protected as well.

2. Cell Suppression

✍ ***Definition :*** *Cell Suppression is a sensitive data protection technique where the table cells containing the potentially sensitive information are hidden.*

- The table cells that contain sensitive information or that can make the inference possible are hidden. Let's take an example.

Sales (units sold)	Asia	America	Europe	Total
Product A	10	25	35	70
Product B	20	75	80	175
Product C	30	10	10	50
Total	60	110	125	295

- It is important to understand that which cells should be suppressed such that the overall sales data for each product and each region is not disclosed.
- Consider the below cell suppression pattern.

Sales (units sold)	Asia	America	Europe	Total
Product A	10	25	35	70
Product B	-	-	-	-
Product C	30	10	10	50
Total	60	110	125	295

- Can you still figure out the suppressed values? Yes, by quick arithmetic. For example, if the total sales in Asia is 60 units, then product B must have sold (60 – 30 – 10) 20 units.
- So, it is important to understand which cells should be actually suppressed to avoid revealing the sensitive information. Now, consider the below cell suppression pattern. Is this better? Can you determine sales per product in each region?

Sales (units sold)	Asia	America	Europe	Total
Product A	10	-	35	-
Product B	-	-	-	175
Product C	-	10	-	-
Total	60	110	125	295

3. Database Partitioning

✍ ***Definition :*** *Database Partitioning is a sensitive data protection technique where the table containing the sensitive information is split into multiple tables such that the attacker would have to know all the related tables to infer the sensitive data.*

- The problem with keeping both the insensitive and sensitive data together in a table is that the attacker has multiple ways to infer sensitive information either directly or indirectly. It is hard to protect the sensitive data against manipulated queries in a mixed (insensitive with sensitive data) table. Database partitioning splits the sensitive and other related data into multiple tables to make it hard for the attacker to make inference attack on just one table.
- Let's see an example. Consider the following table.

ID	Name	Age	Gender	Role	Tenure	Promotion
E001	A	32	M	Designer	2 years	No
E002	B	33	F	Programmer	3 years	Yes
E003	C	34	F	Manager	6 months	No
E004	D	31	M	Architect	2 years	No
E005	E	35	M	Programmer	7 years	Yes
E006	F	34	M	QA	1 year	No

- The table can be partitioned as the following.

ID	Name	Age	Gender
E001	A	32	M
E002	B	33	F
E003	C	34	F
E004	D	31	M
E005	E	35	M
E006	F	34	M

Role ID	Role
R001	Designer
R002	Programmer
R003	Manager
R004	Architect
R005	QA

ID	Role ID
E001	R001
E002	R002
E003	R003
E004	R004
E005	R002
E006	R005

ID	Tenure	Promotion
E001	2 years	No
E002	3 years	Yes
E003	6 months	No
E004	2 years	No
E005	7 years	Yes
E006	1 year	No

- The attacker would now require working with four different tables and the ability to carry out more detailed analysis for inference. If the Role ID to Role mapping table is completely blocked access to along with the Employee ID to Tenure and Promotion mapping table, it becomes very hard to infer anything from the other two tables. You have split the insensitive data and sensitive data and can now apply access controls more rigorously to avoid inference attacks.

4. Noise and Perturbation

✍ ***Definition :*** *Noise and Perturbation is a sensitive data protection technique where fake information (records) is inserted into the table to make the inference misleading.*

- The bogus data inserted into the table makes inference non-conclusive or uncertain. It rather confuses the attacker whether the data is accurate enough to be trusted.
- Let's take an example.

Name	Age	Gender	Role	Tenure	Promotion
A	32	M	Designer	2 years	No
B	33	F	Programmer	3 years	Yes
K	29	F	Programmer	8 years	Yes
C	34	F	Manager	6 months	No
D	31	M	Architect	2 years	No
L	44	M	Programmer	4 years	No
E	35	M	Programmer	7 years	Yes
F	34	M	QA	1 year	No

- The highlighted records in the given example are fake records. Programmer *L*, even though has spent 4 years in the company, is not chosen for promotion. Programmer *K*, with the age of just 29, is shown 8 years in the company and ready for promotion. There is no clear inference to make here.

5. Data Views

✍ ***Definition** : Data View is a sensitive data protection technique where the data is logically partitioned as per the access requirements for various groups.*

- Data Views are created on top of existing tables to contain only rows and columns that are specifically allowed for a group or an entity.
- Let's take an example. Consider that the following Table 5.4.3 is an employee salary database

Table 5.4.3 : Employee salary

Manager Name	Employee Name	Salary
A1	A	S1
	B	S2
	C	S3
B1	D	S4
	E	S5
	F	S6
	G	S7
	H	S8
C1	J	S9
	K	S10

- Each manager is only allowed to view the salary of their direct staff and not others in the company. So, each manager has access to the partial records (records of her direct reports only) and not all the records in the entire table. But HR and Payroll is allowed to see all the records because they need to process salary every month for each employee. So, basically, you have 5 groups that have varied access requirements for the same table –
  - **(i) HR :** requiring full access
  - **(ii) Payroll :** requiring full access
  - **(iii) Manager A1 :** requiring access to employees A, B and C
  - **(iv) Manager B1 :** requiring access to employees D, E, F, G and H
  - **(v) Manager C1 :** requiring access to employees J and K
- So, you can create one database view for each manager.

Manager Name	Employee Name	Salary
A1	A	S1
	B	S2
	C	S3
Manager Name	**Employee Name**	**Salary**
B1	D	S4
	E	S5
	F	S6
	G	S7
	H	S8
Manager Name	**Employee Name**	**Salary**
C1	J	S9
	K	S10

- This way, the respective managers can only access the records of their staff and no one else.

6. Polyinstantiation

✍ ***Definition :*** *Polyinstantiation is a sensitive data protection technique where what data to show is explicitly decided based on the entity that is requesting access.*

Simply put, you can restrict access to sensitive data. But a direct restriction may give an attacker a clue that there is something which is sensitive and hence it is restricted. The attacker then might try other ways of disclosing the data. To avoid this, when an attacker queries the data, she is shown manipulated records and not the real ones. This satisfies the attacker to some extent by assuring her that her actions are actually resulting into the sensitive data disclosure.

Polyinstantiation thus allows creating multiple instances of data as per the entity. It could also create instances as per the sensitivity labels that you learnt in the MAC-based access policy. Let's see an example.

Sensitive Label	Name	Age	Gender	Role	Tenure	Promotion
Confidential	A	32	M	Designer	2 years	No
Unclassified	A	32	M	Designer	4 years	Yes
Confidential	B	33	F	Programmer	3 years	Yes
Unclassified	B	33	F	Programmer	1 year	No

So, an entity that has a clearance level of Confidential would see the actual records of the employee.

Sensitive Label	Name	Age	Gender	Role	Tenure	Promotion
Confidential	A	32	M	Designer	2 years	No
Confidential	B	33	F	Programmer	3 years	Yes

An entity that does not match the Confidential level and is below it, would see manipulated records of the same employee.

Sensitive Label	Name	Age	Gender	Role	Tenure	Promotion
Unclassified	A	32	M	Designer	4 years	Yes
Unclassified	B	33	F	Programmer	1 year	No

7. Data Anonymization

✍ ***Definition :*** *Data Anonymization is a sensitive data protection technique where the sensitive information (usually personal information) from the data is removed or randomized.*

- These days a lot of companies use machine learning and artificial intelligence technology to make meaningful inferences to predict the consumer behaviour or future. The technology could be used to predict chances of someone getting diagnosed with cancer or predicting match result of a football game. Large datasets, from the past studies or results, are collected to train the machine learning models so that the models can learn the past patterns and use those patterns for future prediction.
- To ensure that such datasets do not provide personal information such that to expose someone's privacy, the identity information from the data is taken out (removed) or replaced with something that is meaningless or not so obvious. It then becomes difficult to infer or pinpoint individuals behind the data and the data can be safely used for training purposes.
- Let's take an example. The following Table 5.4.4 shows the medical records of a group of people.

Table 5.4.4

Name	Age at which first consumed tobacco	Age at which diagnosed with cancer	Died at the age of
A	22	37	41
B	17	28	32
C	16	47	51
D	34	41	46

- For research purpose, the name of the person is irrelevant. Data anonymization removes this information or replaces it with something else so that individual information is not disclosed.

Age at which first consumed tobacco	Age at which diagnosed with cancer	Died at the age of
22	37	41
17	28	32
16	47	51
34	41	46

Name	Age at which first consumed tobacco	Age at which diagnosed with cancer	Died at the age of
R001	22	37	41
R002	17	28	32
R003	16	47	51
R004	34	41	46

8. Encryption

✍ ***Definition :*** *Encryption is a sensitive data protection technique where the sensitive information is randomized to make it unreadable by unauthorised entities.*

- You must already be aware of encryption from your prior courses. Encryption ensures that the data, that is to be protected, is not stored in plaintext and is rather stored in cipher text (as encrypted). So, even if the encrypted text is revealed, it does not cause much harm because it is useless until decrypted with the right key to provide the real information.
- So, the Table 5.4.4 from previous examples, could possibly encrypt and store the sensitive data as shown in the following Table 5.4.5.

Table 5.4.5

Name	Age	Gender	Role	Tenure	Promotion
A	32	M	Designer	2 years	Weg2e212763t
B	33	F	Programmer	3 years	Rgrngooi22288
C	34	F	Manager	6 months	Fef3ijfi912e222

Comparison between Inference Protection Techniques

Table 5.4.6

Technique	Complexity	Performance	Protection	Used
Context-dependent	Very High	Low	Medium	Less Frequently
Cell Suppression	Very High	Low	Medium	Less Frequently
Database Partitioning	Low	High	High	Frequently
Noise	High	Low	Medium	Less Frequently
Data Views	Low	High	High	Most often
Polyinstantiation	Medium	Medium	High	Less Frequently
Encryption	Medium	High	Very High	Most often

5.4.5 Multilevel Database Security

- Similar to the MAC-based access model that you learnt earlier,

✍ ***Definition :*** *Multilevel databases store data that have different sensitivity requirements.*

- A particular cell in the table might be more sensitive than the entire record.
- The values resulting from the arithmetic operations on the individual values in the table might be more sensitive than the individual values themselves.
- Categorizing the data as either sensitive or insensitive may not be sufficient.
- Multilevel databases allow you to assign specific sensitivity labels to the data. The sensitivity label might be for a table, for a record or for a cell. Based on the clearance level of the entity requesting access, the access decisions are evaluated and taken.

5.4.5(A) Approaches for Designing Multilevel Databases

There are several approaches for designing multilevel databases.

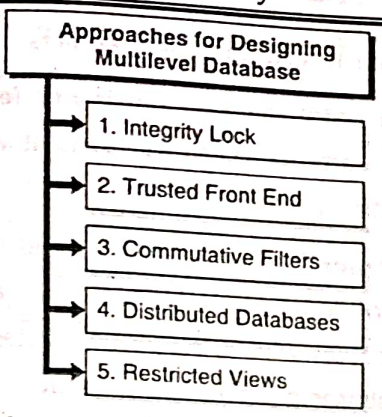


Fig. 5.4.14 : Approaches for designing multilevel database

1. Integrity Lock

✍ ***Definition** : Integrity Lock mechanism for providing multilevel database security assigns the sensitivity label to each data item.*

- Based on the clearance level of the entity requesting access, the access decisions are taken. The sensitivity label is protected from getting changed or disclosed.

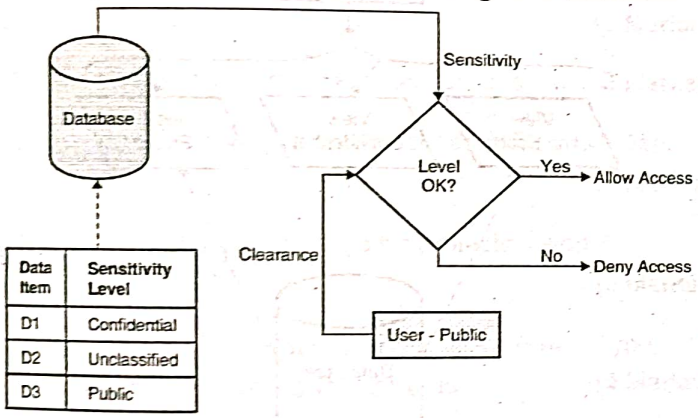


Fig. 5.4.15

- The problem with this approach is that it severely impacts performance.
- Each access requests requires comparing the sensitivity level with clearance level
- Extra space is required to store the sensitivity level itself for each data item
- Sensitivity level itself requires protection from disclosure and alteration

2. Trusted Front End

✍ ***Definition** : Trusted Front End mechanism for providing multilevel database security relies on an external system to validate access decisions and control access to the sensitive data.*

- Databases are best suited for storing and providing information. Implementing security for each data item might add a significant overhead in terms of operations. The databases are not often used as a standalone system. Rather they are used with applications that provide a user interface where the user interacts with the required data.
- For example, you do not directly interact with the Flipkart database for your purchase. Instead you go through the Flipkart portal.

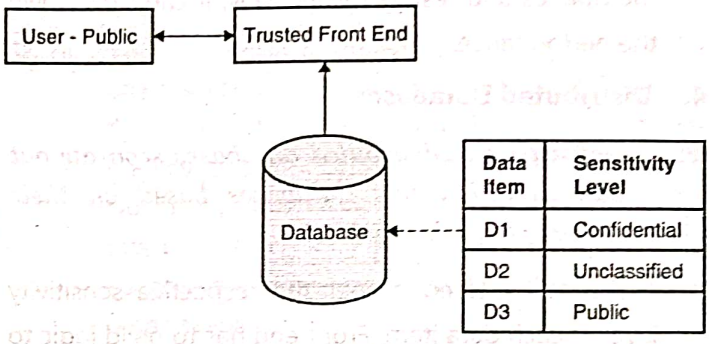


Fig. 5.4.16

- Front end is the application that actually interacts with the database and the items stored therein. The database relies on this front end to make good access decisions based on the user or entity that it is serving. This way, there is no need to carry out access decisions at the database layer. Instead sensitivity-based access control decisions can be taken right from the front-end application. All the logic for evaluating access decisions is built in the front end. The database performance is not hit.

3. Commutative Filters

✍ ***Definition** : Commutative filters examine both incoming requests and outgoing data to enforce appropriate access control.*

- It works very similar to how firewall works at the network layer. It examines the user requests (queries) and reformats them to ensure that they reach to the database securely. Similarly, when the database returns the data as requested by the user, it again examines that all the data that the user would likely see is actually permitted. It does not permit inappropriate requests to go in and unauthorised data to come out.

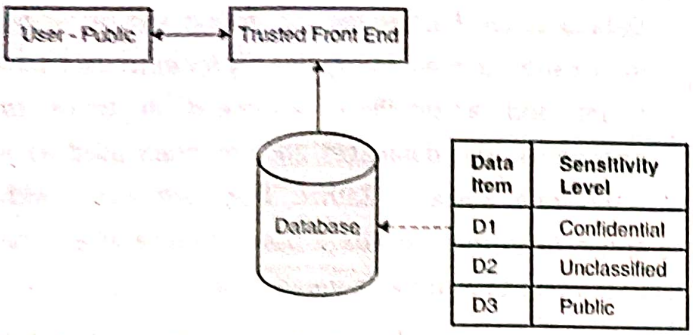


Fig. 5.4.17

- However, because of the pre and post processing of the queries and results respectively, it can slow down the performance.

4. Distributed Databases

Definition : Distributed databases separate out data into multiple databases based on their respective sensitivity levels.

- The data is stored as per the respective sensitivity level of each data item. Front end has to build logic to query the appropriate databases based on the entity requesting the data.
- For example, consider that there are three databases.
  - (i) **Most Sensitive :** Top Secret
  - (ii) **Medium Sensitive :** Confidential
  - (iii) **Insensitive :** Public

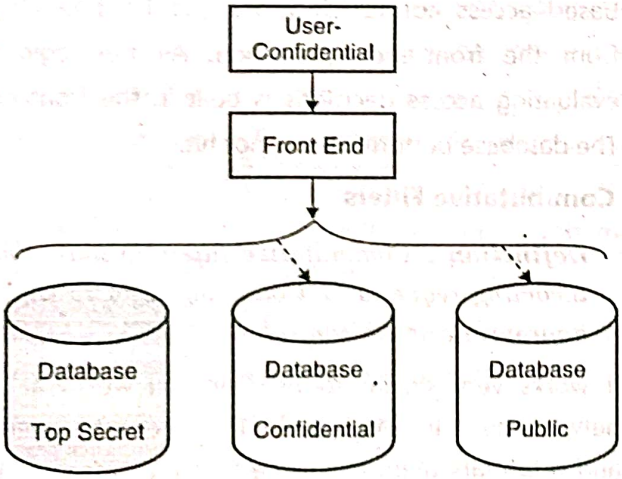


Fig. 5.4.18

- Now, if there is an entity requesting access that has Confidential clearance, the front end would submit query only to the second and the third database. The results from both the databases would be combined and presented to the entity.
- This approach is complex and requires
  - (i) **One database per sensitivity level :** So, if there are 10 sensitivity levels, it would require splitting data into 10 databases.
  - (ii) **Complex Front end :** Front end has to build a lot of logic to re-format and design queries based on who is requesting access and which databases are allowed to be queried as per the clearance level of the entity.

5. Restricted Views

- You learnt earlier how database views can be used to protect against the inference attack. Similarly, you can create views based on sensitivity levels.

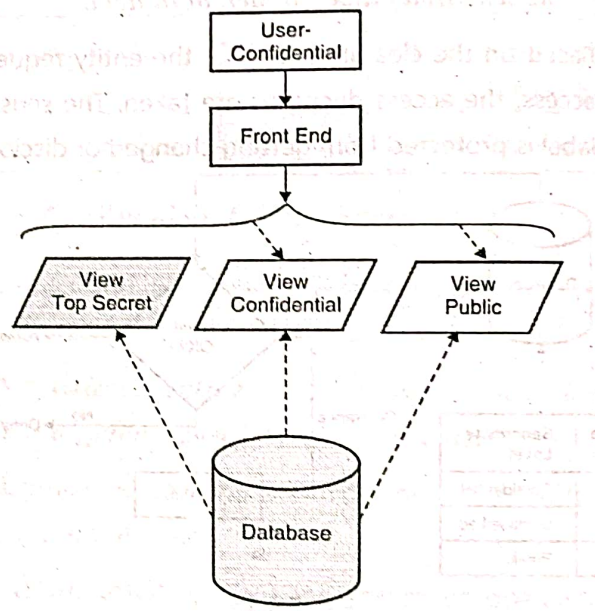


Fig. 5.4.19

- The entity requesting access to the data would be tied to the views that are allowed as per clearance level of the entity.

Comparison between approaches for designing multilevel databases

Sr. No.	Approach	Complexity	Performance
1.	Integrity Lock	Medium	Low
2.	Trusted Front End	Low	High
3.	Commutative Filters	Medium	Medium
4.	Distributed Databases	High	Medium
5.	Restricted Views	Medium	High

Review Questions

Here are a few review questions to help you gauge your understanding of this chapter. Try to attempt these questions and ensure that you can recall the points mentioned in the chapter.

Operating System Security

Q. 1 Write a short note on OS ring architecture. **(4 Marks)**

Q. 2 Draw a block diagram and explain segmentation. **(6 Marks)**

Q. 3 Draw a block diagram and explain pagination. **(6 Marks)**

Q. 4 How can you use Memory Protection Keys? **(6 Marks)**

Q. 5 Compare various memory and address protection techniques. **(6 Marks)**

Q. 6 Compare file protection mechanisms. **(5 Marks)**

Q. 7 Write a short note on Object-based file protection. **(4 Marks)**

Q. 8 How does group-based file protection work? **(6 Marks)**

Q. 9 What is "Something You Know" type of authentication? Give examples. **(4 Marks)**

Q. 10 List the advantages and disadvantages of "Something You Know" based authentication. **(6 Marks)**

Q. 11 What is "Something You Have" type of authentication? Give examples. **(4 Marks)**

Q. 12 List the advantages and disadvantages of "Something You Have" based authentication. **(6 Marks)**

Q. 13 Define the term biometric. Give examples of static and dynamic biometrics. **(4 Marks)**

Q. 14 List the advantages and disadvantages of "Someone You Are" based authentication. **(6 Marks)**

Q. 15 Write a short note on location-based authentication. **(4 Marks)**

Q. 16 Alex is proposing that we should at least use two-factor authentication these days to secure our accounts. John says "Come on Alex, don't you know that we already require a PIN followed by a strong password to authenticate on our application. We are good here". Alex says, no, that is not what I mean. What could Alex mean here? **(4 Marks)**

Q. 17 You should choose simple and easy to remember passwords so that you don't have hard time recalling them. For example, your name or your birth date can be a good password. Simple and easy to remember. Isn't it? **(4 Marks)**

Q. 18 Your supervisor is worried that not everyone in the organisation is using strong passwords for securing their accounts. You suggest her that we should enforce password quality criteria in the organisation. Confused, she asks you, what does password quality criteria mean and what are some of the suggested quality criteria we should adopt in the organisation?

Q. 19 Describe how passwords are typically stored and verified on a system. **(8 Marks)**

Q. 20 Explain any three attacks on Password Based Authentication. **(6 Marks)**

Q. 21 Write a short note on Synchronous tokens. **(4 Marks)**

Q. 22 Write a short note on Asynchronous tokens. **(4 Marks)**

Q. 23 Describe software tokens. **(6 Marks)**

Q. 24 What are the components of a biometric system? **(6 Marks)**

Q. 25 Explain the enrolment process in a biometric system with a block diagram. **(6 Marks)**

Q. 26 Explain the verification process in a biometric system with a block diagram. **(6 Marks)**

Q. 27 Explain FRR, FAR, and CER with respect to a biometric system. How do you choose a biometric system based on these parameters? **(6 Marks)**

Q. 28 A biometric system rejects 2 authorised attempts out of 10. Calculate its FRR. **(4 Marks)**

Q. 29 A biometric system accepts 4 unauthorised attempts out of 10. Calculate its FAR. **(4 Marks)**

Q. 30 Compare any 6 biometric methods of your choice on the basis of **(8 Marks)**

(a) Processing Speed

(b) Accuracy and

(c) Ease of Enrolment

Q. 31 You are implementing a patch management program. What are the general steps in patch management? **(6 Marks)**

Q. 32 Explain hardening and give suitable examples of hardening recommendations. **(6 Marks)**

Q. 33 Dan is saying that you should implement monitoring system for protecting your system What is he referring to? How such a solution works? **(6 Marks)**

Q. 34 What do you understand by full volume encryption? **(4 Marks)**

Q. 35 Write a short note on BitLocker. **(5 Marks)**

Q. 36 Describe Encrypting File System (EFS). **(6 Marks)**

Q. 37 Setting disk quota provides security. Comment. **(4 Marks)**

Q. 38 Compare EFS and BitLocker. **(4 Marks)**

Q. 39 On Linux OS, the options with which the file systems are mounted could be crucial to security. Explain with example. **(8 Marks)**

Q. 40 Draw block diagrams of eCryptfs, dm-crypt and fscrypt (no explanation required). **(6 Marks)**

Q. 41 Write a short note on eCryptfs. **(4 Marks)**

Q. 42 Explain dm-crypt. **(6 Marks)**

Q. 43 Compare eCryptfs, dm-crypt and fscrypt. **(6 Marks)**

Q. 44 Write a short note on fscrypt. **(4 Marks)**

Database Security

Q. 45 Shallie is new to databases. She comes to you to understand what the database security requirements are. What do you suggest? **(6 Marks)**

Q. 46 Element accuracy is crucial for maintaining database integrity. Explain. **(6 Marks)**

Q. 47 Describe various database integrity protection mechanisms. **(8 Marks)**

Q. 48 Draw a block diagram and explain savepoints. **(6 Marks)**

Q. 49 Write SQL queries to create 3 savepoints and explain how they can be used. **(6 Marks)**

Q. 50 Describe two-phase commit. **(8 Marks)**

Q. 51 Draw block diagrams detailing success and failure scenarios in two-phase commit (no explanation required). **(6 Marks)**

Q. 52 Explain the various types of sensitive data disclosures. **(8 Marks)**

Q. 53 With a suitable example, explain direct inference attack. **(6 Marks)**

Q. 54 Write a short note on aggregation. **(4 Marks)**

Q. 55 Compare inference protection techniques. **(6 Marks)**

Q. 56 What is context-dependent access control? **(5 Marks)**

Q. 57 With a suitable example, explain cell suppression. **(6 Marks)**

Q. 58 Write a short note on data views. **(5 Marks)**

Q. 59 Explain Polyinstantiation technique with respect to database security. **(6 Marks)**

Q. 60 Explain data anonymization. **(6 Marks)**

Q. 61 Write a short note on Trusted Front End. **(4 Marks)**

Q. 62 Compare approaches for designing multilevel database security. **(4 Marks)**

□□□