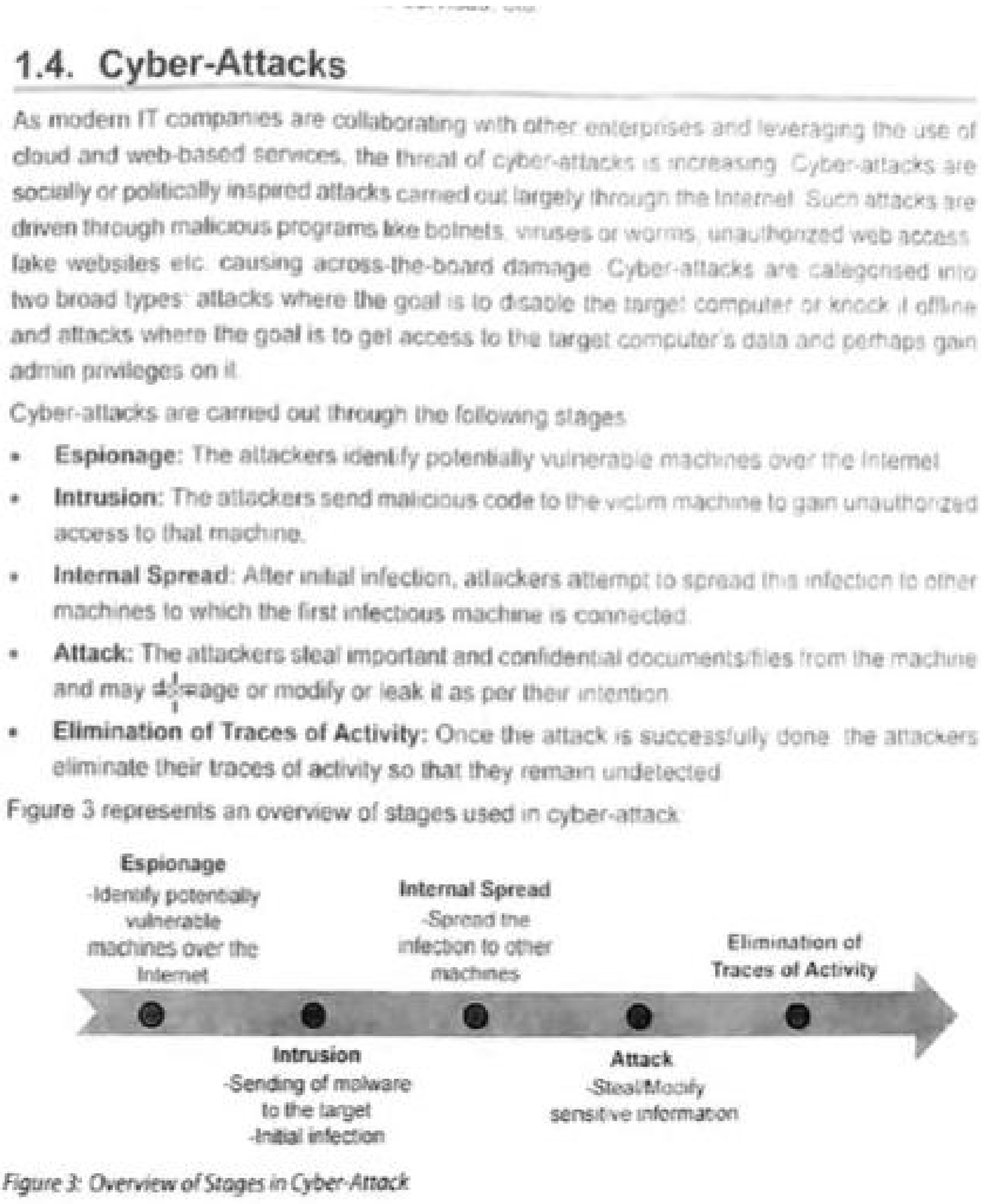
Cyber Attacks



services, etc.

1.4. Cyber-Attacks

As modern IT companies are collaborating with other enterprises and leveraging the use of cloud and web-based services, the threat of cyber-attacks is increasing. Cyber-attacks are socially or politically inspired attacks carried out largely through the Internet. Such attacks are driven through malicious programs like botnets, viruses or worms, unauthorized web access, fake websites etc. causing across-the-board damage. Cyber-attacks are categorised into two broad types: attacks where the goal is to disable the target computer or knock it offline and attacks where the goal is to get access to the target computer's data and perhaps gain admin privileges on it.

Cyber-attacks are carried out through the following stages:

- **Espionage:** The attackers identify potentially vulnerable machines over the Internet
- **Intrusion:** The attackers send malicious code to the victim machine to gain unauthorized access to that machine.
- **Internal Spread:** After initial infection, attackers attempt to spread this infection to other machines to which the first infectious machine is connected.
- **Attack:** The attackers steal important and confidential documents/files from the machine and may damage or modify or leak it as per their intention.
- **Elimination of Traces of Activity:** Once the attack is successfully done, the attackers eliminate their traces of activity so that they remain undetected.

Figure 3 represents an overview of stages used in cyber-attack.

Espionage
-Identify potentially vulnerable machines over the Internet
Intrusion
-Sending of malware to the target
-Initial infection
Internal Spread
-Spread the infection to other machines
Attack
-Steal/Modify sensitive information
Elimination of Traces of Activity

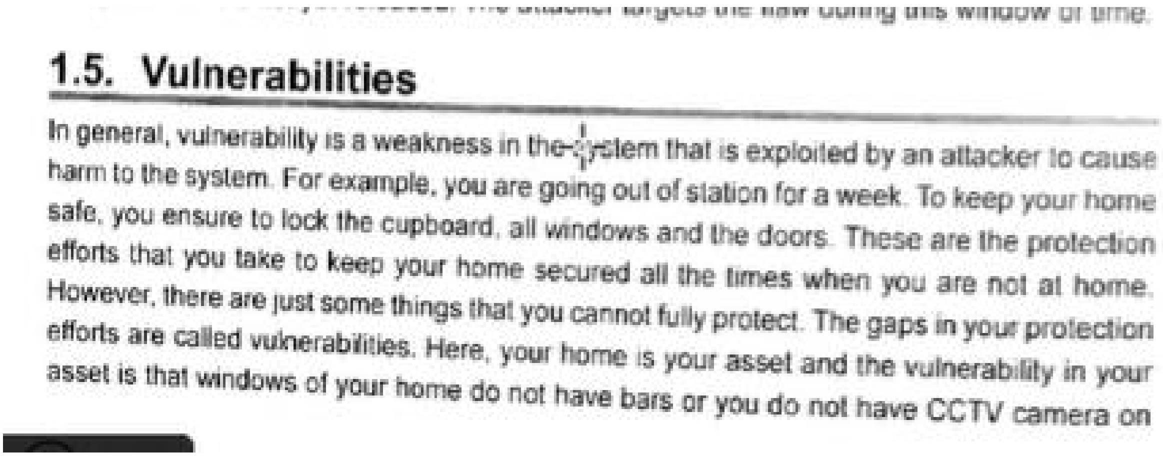
Figure 3: Overview of Stages in Cyber-Attack

Today, the goal of infrastructure security is to protect its valuable assets from cyber-attacks. To know different ways of securing the assets, it is important to understand multiple technical methods deployed by cybercriminals to ruin the infrastructure.

There are always new methods coming up, however, the most common techniques followed by attackers are as follows:

- **Malware:** It is the short form of 'malicious software' which is designed with an intention to gain root access to target machines so that it can be operated remotely. The examples of such malware are viruses, worms and trojan horses.
- **Phishing:** This technique makes use of fake emails or fake websites to steal the user's sensitive information like his/her login credentials. The attackers urge the victim to click on a certain link disguised as an important document. After clicking on this link, the victim is taken to a fake web page where user is asked to enter the sensitive information like bank's username and password.
- **Denial of Service:** It is a brute-force method by which the attacker sends a flood of requests to the victim server in a short amount of time. As a result, the victim server gets overloaded and starts denying the requests coming from legitimate as well as malicious users. To perform this attack in a distributed environment, the attacker creates a botnet over the Internet and uses it to launch the flood of requests. This attack is termed as Distributed Denial of Service (DDoS) attack.
- **Man-in-the-Middle:** The attacker injects himself/herself secretly between the user and the web service. When a user interacts with the web service, the attacker surreptitiously harvests the confidential information.
- **SQL injection:** Many databases are designed to accept commands written in Structured Query Language (SQL). However, in this kind of attack, attacker injects malicious code in SQL statement via web page input. This malicious code helps the attacker to gain confidential information of the user from the database.
- **Zero-day exploit:** This attack happens when the flaw in the software is detected but the patch for it is not yet released. The attacker targets the flaw during this window of time.

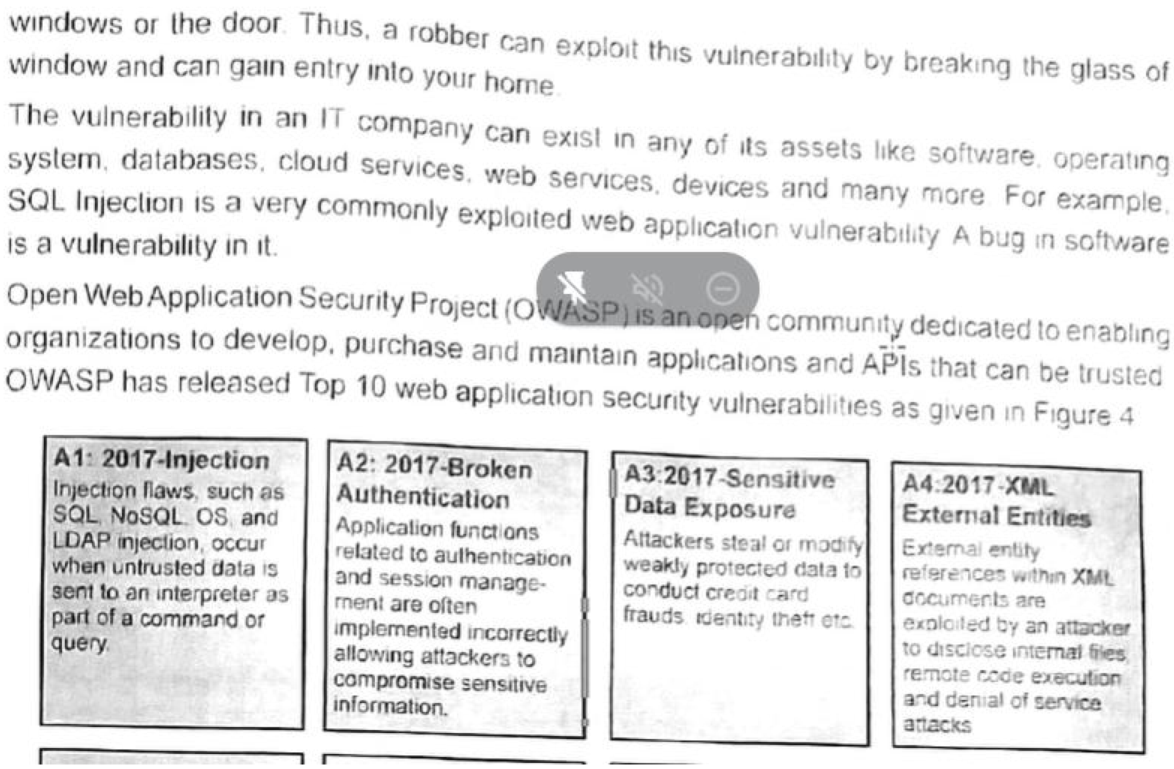
Vulnerabilities



attacker targets the flaw during this window of time.

1.5. Vulnerabilities

In general, vulnerability is a weakness in the system that is exploited by an attacker to cause harm to the system. For example, you are going out of station for a week. To keep your home safe, you ensure to lock the cupboard, all windows and the doors. These are the protection efforts that you take to keep your home secured all the times when you are not at home. However, there are just some things that you cannot fully protect. The gaps in your protection efforts are called vulnerabilities. Here, your home is your asset and the vulnerability in your asset is that windows of your home do not have bars or you do not have CCTV camera on

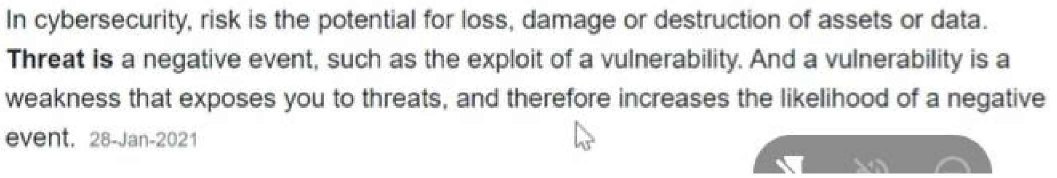


windows or the door. Thus, a robber can exploit this vulnerability by breaking the glass of window and can gain entry into your home.

The vulnerability in an IT company can exist in any of its assets like software, operating system, databases, cloud services, web services, devices and many more. For example, SQL Injection is a very commonly exploited web application vulnerability. A bug in software is a vulnerability in it.

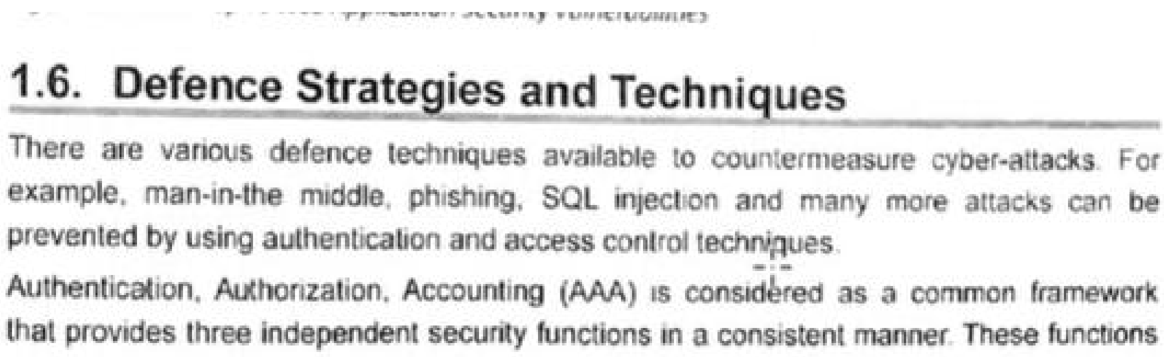
Open Web Application Security Project (OWASP) is an open community dedicated to enabling organizations to develop, purchase and maintain applications and APIs that can be trusted. OWASP has released Top 10 web application security vulnerabilities as given in Figure 4

A1: 2017-Injection	A2: 2017-Broken Authentication	A3:2017-Sensitive Data Exposure	A4:2017-XML External Entities
Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query.	Application functions related to authentication and session management are often implemented incorrectly allowing attackers to compromise sensitive information.	Attackers steal or modify weakly protected data to conduct credit card frauds, identity theft etc.	External entity references within XML documents are exploited by an attacker to disclose internal files, remote code execution and denial of service attacks



In cybersecurity, risk is the potential for loss, damage or destruction of assets or data. **Threat is** a negative event, such as the exploit of a vulnerability. And a vulnerability is a weakness that exposes you to threats, and therefore increases the likelihood of a negative event. 28-Jan-2021

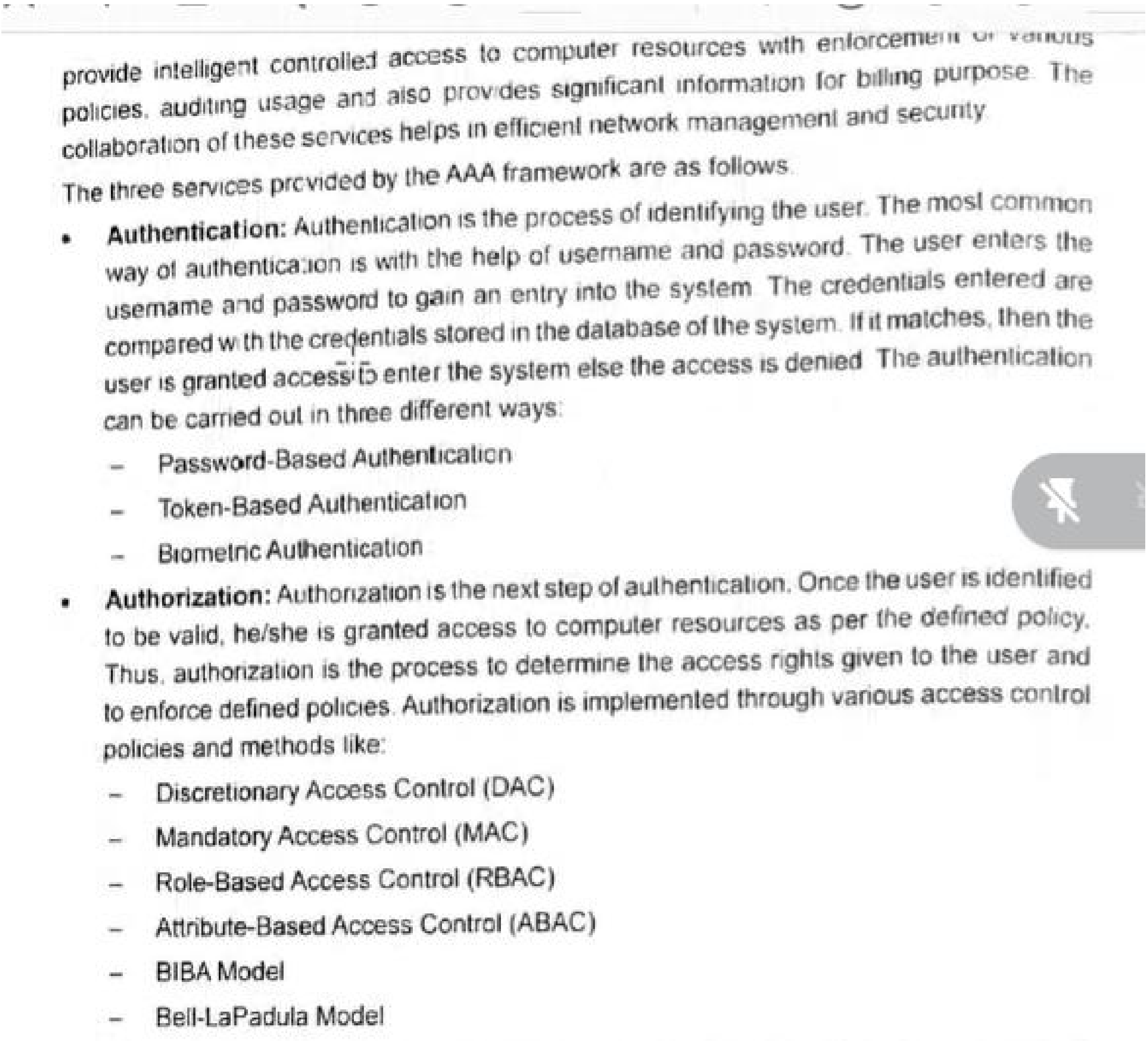
Defense Strategies and Techniques



1.6. Defence Strategies and Techniques

There are various defence techniques available to countermeasure cyber-attacks. For example, man-in-the middle, phishing, SQL injection and many more attacks can be prevented by using authentication and access control techniques.

Authentication, Authorization, Accounting (AAA) is considered as a common framework that provides three independent security functions in a consistent manner. These functions



provide intelligent controlled access to computer resources with enforcement of various policies, auditing usage and also provides significant information for billing purpose. The collaboration of these services helps in efficient network management and security.

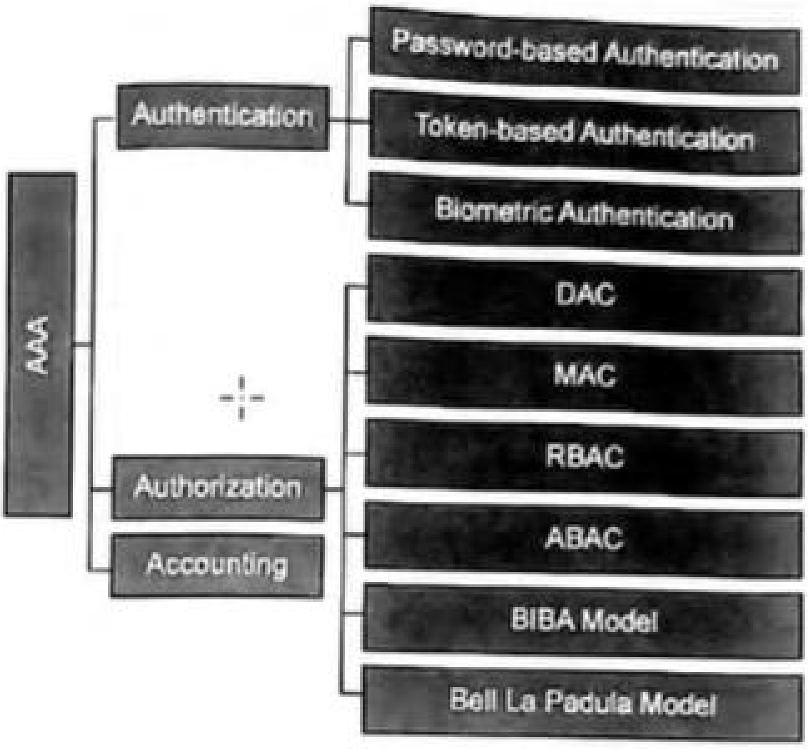
The three services provided by the AAA framework are as follows.

- **Authentication:** Authentication is the process of identifying the user. The most common way of authentication is with the help of username and password. The user enters the username and password to gain an entry into the system. The credentials entered are compared with the credentials stored in the database of the system. If it matches, then the user is granted access to enter the system else the access is denied. The authentication can be carried out in three different ways:
  - Password-Based Authentication
  - Token-Based Authentication
  - Biometric Authentication
- **Authorization:** Authorization is the next step of authentication. Once the user is identified to be valid, he/she is granted access to computer resources as per the defined policy. Thus, authorization is the process to determine the access rights given to the user and to enforce defined policies. Authorization is implemented through various access control policies and methods like:
  - Discretionary Access Control (DAC)
  - Mandatory Access Control (MAC)
  - Role-Based Access Control (RBAC)
  - Attribute-Based Access Control (ABAC)
  - BIBA Model
  - Bell-LaPadula Model

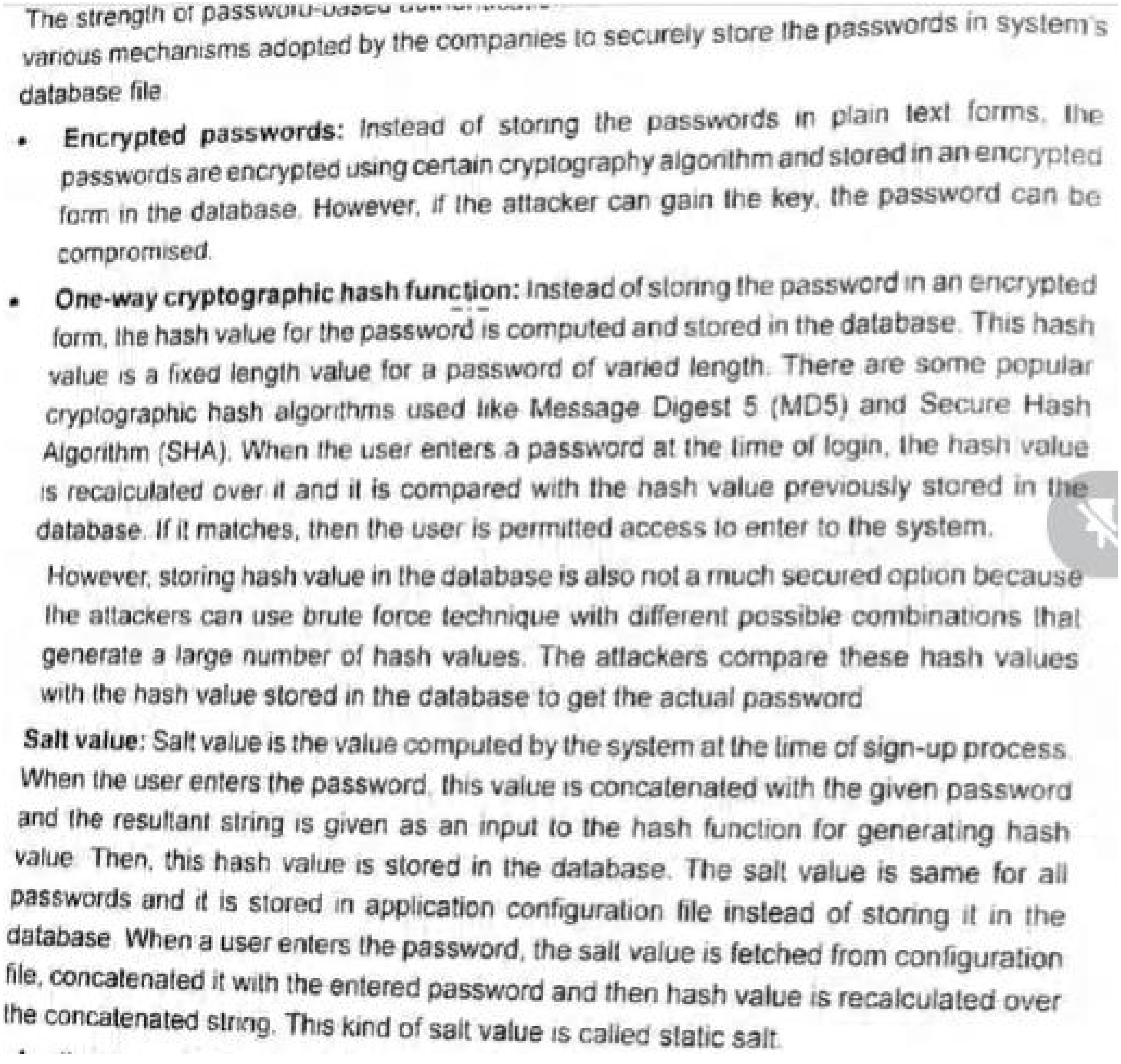
- Bell-LaPadula Model

- **Accounting:** Accounting is the process that records the amount of resources accessed by the user within a specified period. These resources can be amount of system time, amount of data or memory accessed by the user, etc. Accounting is carried out with the help of gathering session statistics and user information. The logging of this information helps in billing, resource usage analysis, provisioning resources in future, trend analysis etc.

methods is described in Figure 5



Authentication

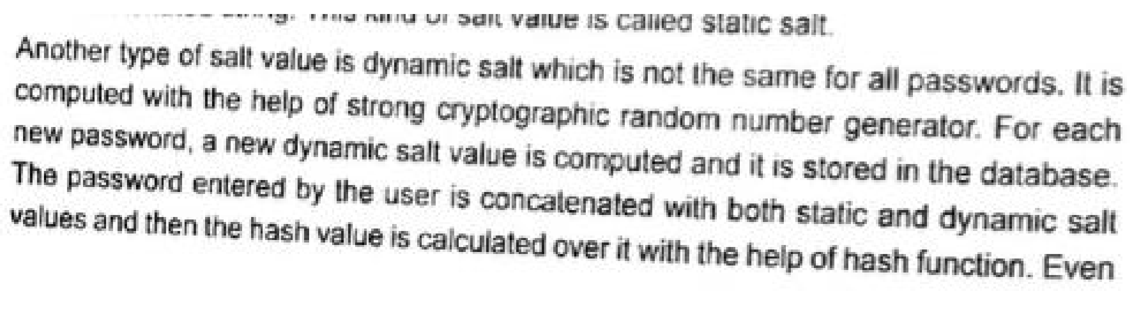


The strength of password-based various mechanisms adopted by the companies to securely store the passwords in system's database file

- **Encrypted passwords:** Instead of storing the passwords in plain text forms, the passwords are encrypted using certain cryptography algorithm and stored in an encrypted form in the database. However, if the attacker can gain the key, the password can be compromised.
- **One-way cryptographic hash function:** Instead of storing the password in an encrypted form, the hash value for the password is computed and stored in the database. This hash value is a fixed length value for a password of varied length. There are some popular cryptographic hash algorithms used like Message Digest 5 (MD5) and Secure Hash Algorithm (SHA). When the user enters a password at the time of login, the hash value is recalculated over it and it is compared with the hash value previously stored in the database. If it matches, then the user is permitted access to enter to the system.

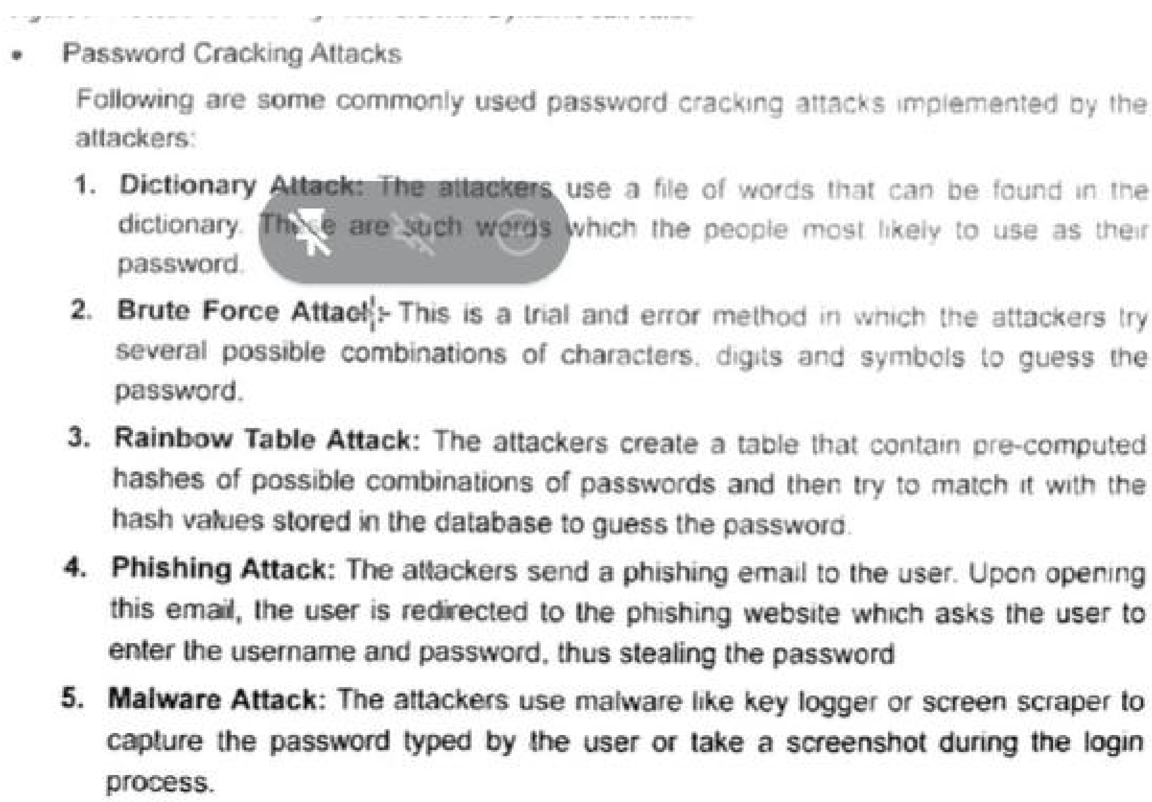
  However, storing hash value in the database is also not a much secured option because the attackers can use brute force technique with different possible combinations that generate a large number of hash values. The attackers compare these hash values with the hash value stored in the database to get the actual password

Salt value: Salt value is the value computed by the system at the time of sign-up process. When the user enters the password, this value is concatenated with the given password and the resultant string is given as an input to the hash function for generating hash value. Then, this hash value is stored in the database. The salt value is same for all passwords and it is stored in application configuration file instead of storing it in the database. When a user enters the password, the salt value is fetched from configuration file, concatenated it with the entered password and then hash value is recalculated over the concatenated string. This kind of salt value is called static salt.



This kind of salt value is called static salt.

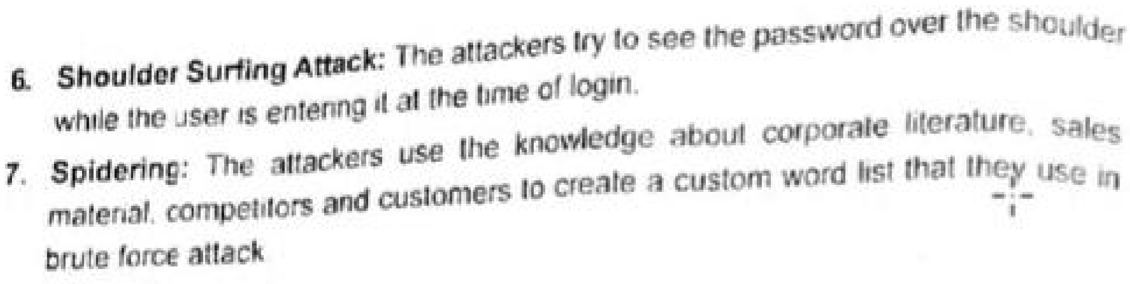
Another type of salt value is dynamic salt which is not the same for all passwords. It is computed with the help of strong cryptographic random number generator. For each new password, a new dynamic salt value is computed and it is stored in the database. The password entered by the user is concatenated with both static and dynamic salt values and then the hash value is calculated over it with the help of hash function. Even



- Password Cracking Attacks

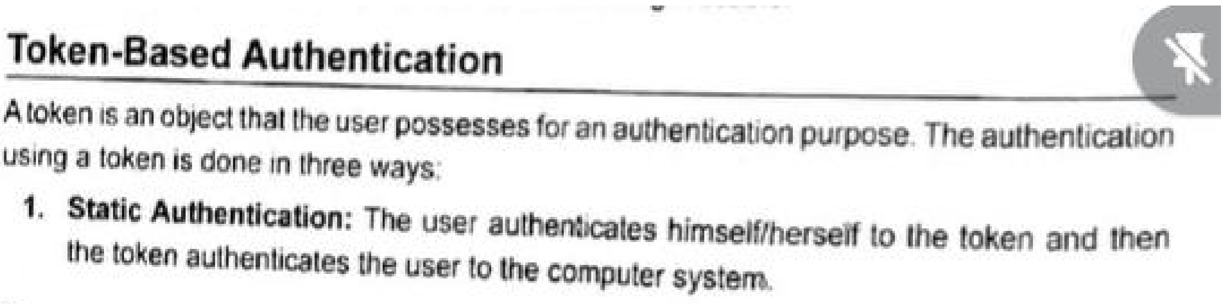
  Following are some commonly used password cracking attacks implemented by the attackers:

  1. **Dictionary Attack:** The attackers use a file of words that can be found in the dictionary. These are such words which the people most likely to use as their password.
  2. **Brute Force Attack:** This is a trial and error method in which the attackers try several possible combinations of characters, digits and symbols to guess the password.
  3. **Rainbow Table Attack:** The attackers create a table that contain pre-computed hashes of possible combinations of passwords and then try to match it with the hash values stored in the database to guess the password.
  4. **Phishing Attack:** The attackers send a phishing email to the user. Upon opening this email, the user is redirected to the phishing website which asks the user to enter the username and password, thus stealing the password
  5. **Malware Attack:** The attackers use malware like key logger or screen scraper to capture the password typed by the user or take a screenshot during the login process.



6. **Shoulder Surfing Attack:** The attackers try to see the password over the shoulder while the user is entering it at the time of login.
7. **Spidering:** The attackers use the knowledge about corporate literature, sales material, competitors and customers to create a custom word list that they use in brute force attack

Token Based Authentication



Token-Based Authentication

A token is an object that the user possesses for an authentication purpose. The authentication using a token is done in three ways:

1. **Static Authentication:** The user authenticates himself/herself to the token and then the token authenticates the user to the computer system.

2. **Dynamic Authentication:** The token generates the password dynamically, like one password per minute and then it is either manually entered or electronically entered into computer system. To implement this method, it is necessary to have synchronization among the token and the computer system so that the computer system can understand the current password.
3. **Challenge-Response:** The computer system generates a cryptographic challenge to which smart token generates a response. The cryptographic challenge can be in the form of certain random string which is encrypted by the token using token's private key and sent it to the system. As there is least possibility to forge token's private key, the authentication is proved.

- **Memory Cards:** Memory cards store the data but do not process it. These cards have magnetic tape at the back. An example of such a card is the card that is used at the hotel to gain entry into the room. The memory card is combined with PIN for computer user authentication and for use in systems like Automatic Teller Machine (ATM).
- **Smart Cards:** Smart cards are able to process the data. Most of the smart tokens are in the form of a smart card and are typically the same size as credit card or driving license. The smart card has a chip implanted into it which contains microprocessor with three types of memory; Random Access Memory (RAM), Read-Only Memory (ROM) and Electrically Erasable Programmable Read-Only Memory (EEPROM). RAM is used to hold temporary data which is generated during application's execution. ROM stores permanent data such as card number and card holder's name. EEPROM holds the data that changes with time. For example, in telephone card, talk-time remaining is stored in EEPROM. Figure 7 represents layout of a smart card.

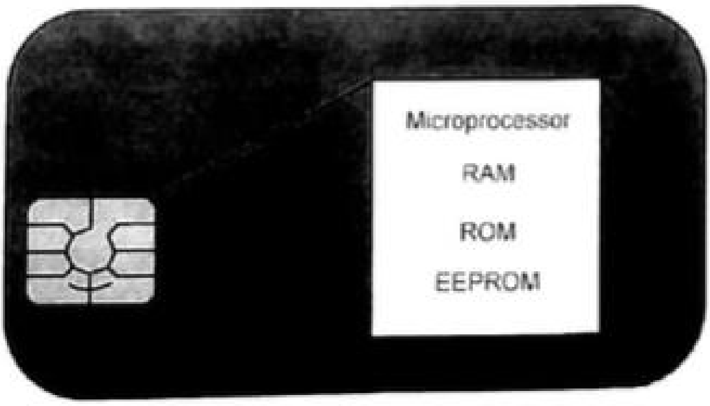


Figure 7: Smart Card Layout

Types of Smart Card

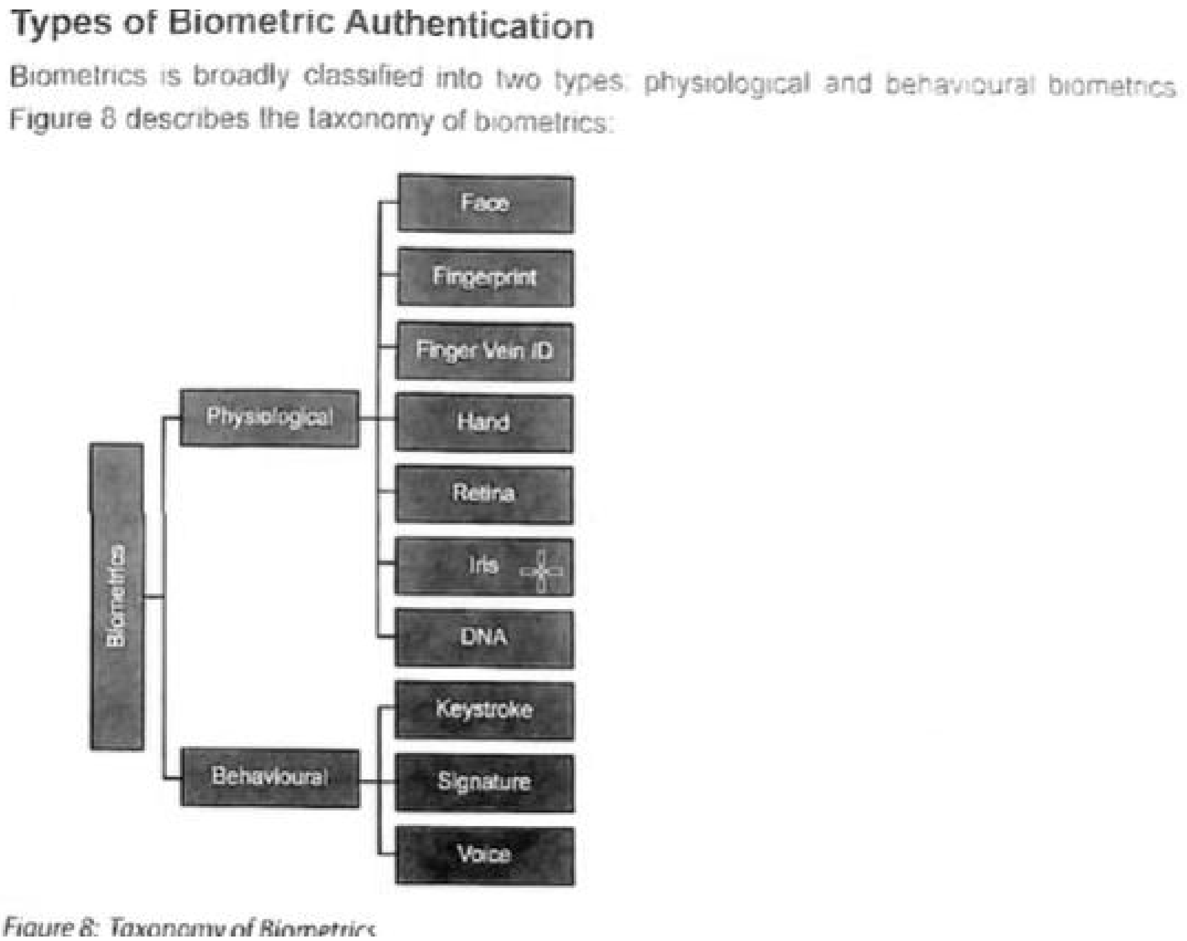
Smart cards are divided into following types depending on card reading and writing methodology, capabilities of chip and type of chip embedded into the card.

1. **Contact Smart Cards:** These are the most common type of smart card. Contact smart cards are inserted into a smart card reader that has a direct connection to a conductive contact p ate on the surface of the card. Commands, data and card status are transmitted over these physical contact points.
2. **Contactless Smart Cards:** Instead of direct connection, these cards should be in close proximity to a card reader to be read. The card and the reader are both armed with antennae and communicate using radio frequencies over the contactless link.
3. **Dual-Interface Cards:** These cards are a combination of contactless and contact interfaces.
4. **Hybrid Smart Cards:** These cards encompass more than one smart card technology For example, a hybrid smart card might have two chips; one chip with embedded microprocessor that is accessed through a contact reader and another chip with RFID used for proximity connection. The two different chips serve different applications like proximity chip is used for physical access to restricted areas while the contact smart card chip is used for single sign-on authentication.

5. **Memory Smart Cards:** These cards contain memory chips only and can only store, read and write data to the chip; the data on memory smart cards can be overwritten or modified, but the card itself is not programmable so data cannot be processed or modified programmatically. Memory smart cards can be read-only and used to store data such as PIN, password or public key; they can also be read-write and used to write or update user data. Memory smart cards can be configured to be rechargeable or

5. **Memory Smart Cards:** These cards contain memory chips only and can only store, read and write data to the chip; the data on memory smart cards can be overwritten or modified, but the card itself is not programmable so data cannot be processed or modified programmatically. Memory smart cards can be read-only and used to store data such as PIN, password or public key; they can also be read-write and used to write or update user data. Memory smart cards can be configured to be rechargeable or disposable, in which case they contain data that can only be used once or for a limited time before being updated or discarded.
6. **Microprocessor Smart Cards:** These cards have a microprocessor embedded onto the chip in addition to memory blocks. This type of card can be used for more than one function and is usually designed to enable adding, deleting and otherwise manipulating data in memory.

Types of biometric



Types of Biometric Authentication

Biometrics is broadly classified into two types: physiological and behavioural biometrics. Figure 8 describes the taxonomy of biometrics:

Figure 8: Taxonomy of Biometrics

2. **Fingerprint Scanning:** Finger scanning is the most widely used authentication technique and it is the digital version of the ink-and-paper fingerprinting process. It works with details in the pattern of raised areas and branches in a human finger image. There are three types of fingerprint scanner: optical, capacitive and ultrasound. An **optical scanner** is the basic scanner that takes a photo of the finger, identifies the print pattern and then compiles it into an identification code. A **capacitive scanner** works by measuring electrical signals sent from the finger to the scanner. Print ridges directly touch the scanner, sending electrical current, while the valleys between print ridges create air gaps. A capacitive scanner basically maps out these contact points and air gaps, resulting in an absolutely unique pattern. These are ones used in smartphones and laptops. **Ultrasonic scanners** will make their appearance in the newest generation of smartphones. Basically, these will emit ultrasounds that will reflect back into the scanner. Similar to a capacitive one, it forms a map of the finger unique to the individual.

   Google and Apple phones store your fingerprint on the device itself and do not store a copy on their own servers. Apple's TouchID does not store the actual image of the fingerprint, but a mathematical representation of it. Android phones store the fingerprint data in a secure part of the main processor called Trusted Execution Environment (TEE). The TEE is isolated from other parts of the processor and does not directly interact with installed apps.

3. **Finger Vein ID:** It is based on the unique vascular pattern in an individual's finger.

4. **Hand Geometry:** Hand geometry identifies users by the shape of their hand. The hand is measured along many dimensions and then the measurements are compared with the measurements stored in the database.

5. **Retina Scan:** It produces an image of the blood vessel pattern in the light-sensitive surface lining the individual's inner eye.

5. **Retina Scan:** It produces an image of the blood vessel pattern in the light-sensitive surface lining the individual's inner eye.

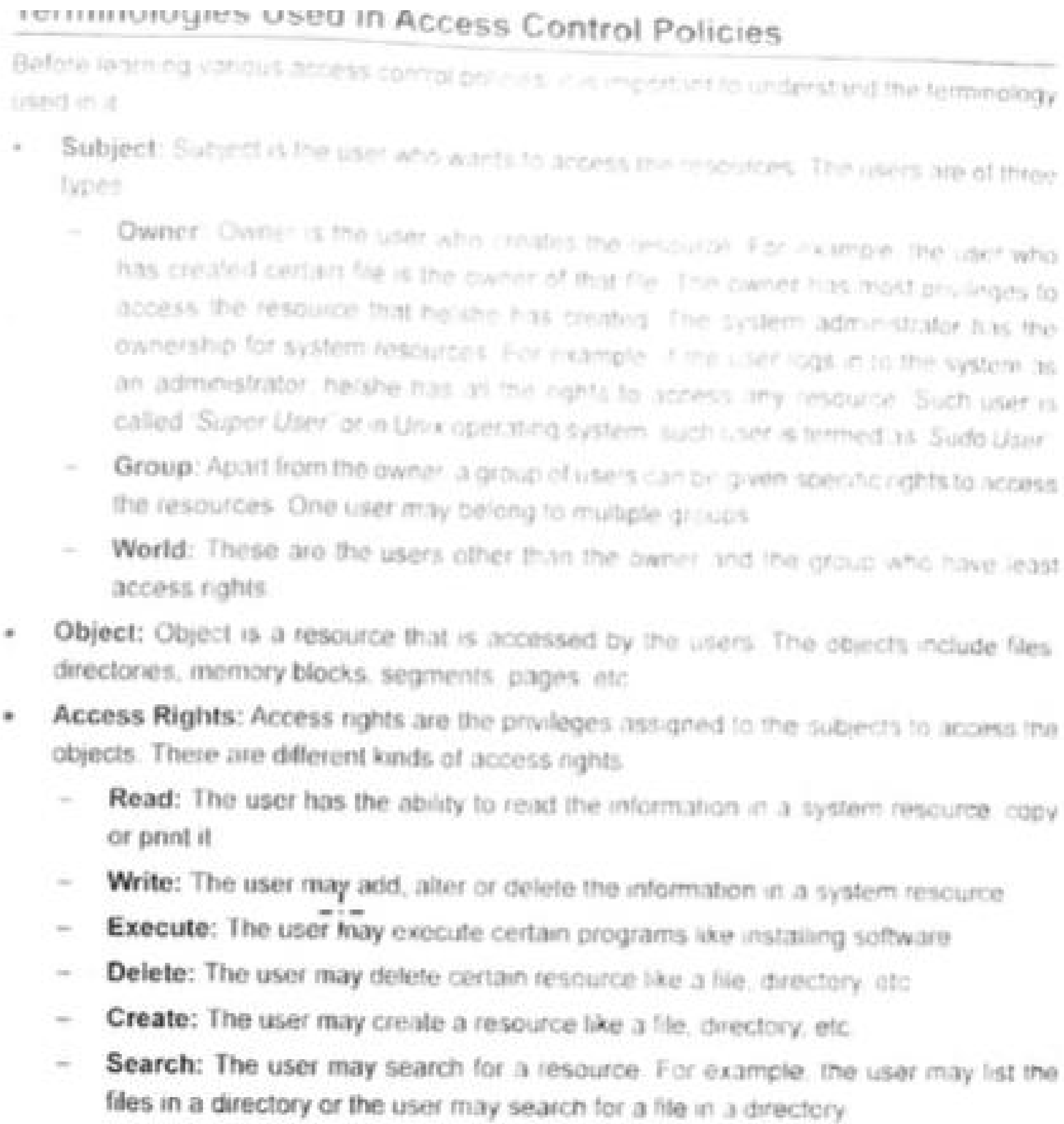
6. **Iris Recognition:** This is used to identify individuals based on unique patterns within the ring-shaped region surrounding the pupil of the eye. Since the eye is an externally visible organ, the iris can be matched using a photograph. However, this requires high proximity and a large amount of storage.

7. **DNA:**DNA authentication is based on the fact that 0.10% of a person's entire genome is unique to them. The chance of two individuals sharing the same DNA profile is less than one in a hundred billion. This type of authentication requires physical sample. Hence, the person needs to be present physically with the testing unit. It requires expensive technology as compared to other physiological techniques.

Behavioural Biometrics

1. **Keystroke Recognition:** Keystroke authentication employs keystroke dynamics by analyzing dynamic typing patterns which cannot be shared or imitated. Based on individual's typing rhythm, the sign-in attempt is verified. It can be easily integrated with any web application with just a few lines of code. Thus, keystroke recognition focuses on how you type instead of what you type.
2. **Signature Recognition:** It is the process to determine to whom a particular signature belongs to. There are two modes in which signature recognition is done: static signature recognition (offline) and dynamic signature recognition (online). In the static mode, the person does a signature on a paper, then scans that signature and the biometric system recognizes it through its shape. In the dynamic mode, the person signs on a digitizing tablet that gains the signature in real time.
3. **Voice Recognition:** This type of authentication does not require proximity. These systems rely on characteristics created by the shape of the speaker's mouth and throat, rather than on variable conditions. People often say the same words in different ways. However, this change does not alter the underlying physical characteristics of the voice. Nevertheless, if the user catches a cold, it affects his/her larynx and results in altering these physical characteristics.

Process of Biometric Authentication



Terminologies Used in Access Control Policies

Before learning various access control policies, it is important to understand the terminology used in it.

- **Subject**: Subject is the user who wants to access the resources. The users are of three types.
  - **Owner**: Owner is the user who creates the resource. For example, the user who has created certain file is the owner of that file. The owner has most privileges to access the resource that he/she has created. The system administrator has the ownership for system resources. For example, if the user logs in to the system as an administrator, he/she has all the rights to access any resource. Such user is called '*Super User*' or in Unix operating system, such user is termed as '*Sudo User*'.
  - **Group**: Apart from the owner, a group of users can be given specific rights to access the resources. One user may belong to multiple groups.
  - **World**: These are the users other than the owner and the group who have least access rights.
- **Object:** Object is a resource that is accessed by the users. The objects include files, directories, memory blocks, segments, pages, etc.
- **Access Rights:** Access rights are the privileges assigned to the subjects to access the objects. There are different kinds of access rights.
  - **Read:** The user has the ability to read the information in a system resource, copy or print it.
  - **Write:** The user may add, alter or delete the information in a system resource.
  - **Execute:** The user may execute certain programs like installing software.
  - **Delete:** The user may delete certain resource like a file, directory, etc.
  - **Create:** The user may create a resource like a file, directory, etc.
  - **Search:** The user may search for a resource. For example, the user may list the files in a directory or the user may search for a file in a directory.

Mandatory Access Control MAC

Figure 10: Access Control

Mandatory Access Control (MAC)

Mandatory Access Control (MAC) is the strictest access control policy. The access control policies are defined by system administrator. The access to resource objects is strictly controlled by the operating system based on system administrator configured settings. The users cannot change the access control of a resource in MAC policy. Even, the owner of the resource (e.g. file) cannot change access policies set by the operating system. This is analogous to the situation wherein the law allows a court to access driving records without the owners' permission. This is a mandatory control, because the owner of the record has no control over the court accessing the information.

Initially, the data is categorized according to the level of confidentiality that needs to be maintained to protect the data. The organization of data in this way is called **multilevel security**. Figure 11 represents confidentiality levels organized in the pyramid form:

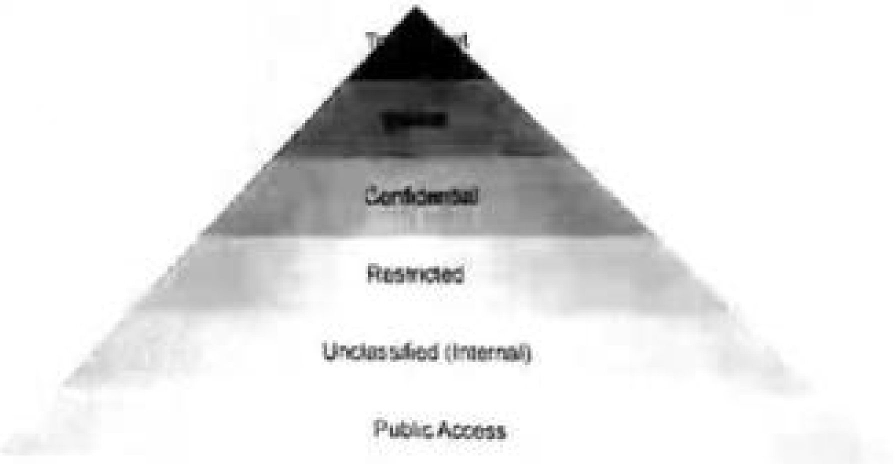


Figure 11: Multilevel Security of Data

The choice of a level is often based on an impact assessment. The government often has its own set of rules which includes rules on determining the level for an information asset and rules on how to protect information classified at each level.

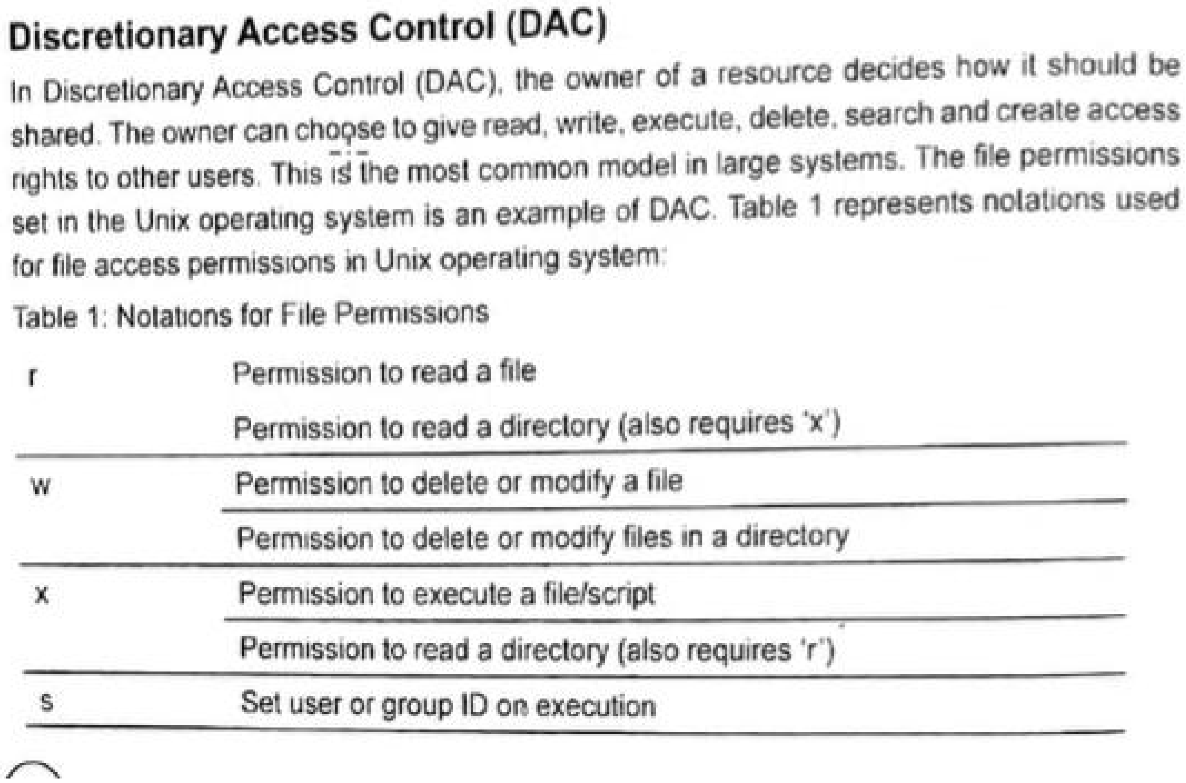
In MAC, according to the

Figure 13: Access Permissions in Windows 10 System

MAC is by far the most secure access control environment but does not come without a price. Following are the drawbacks of MAC:

- MAC requires a considerable amount of planning before it can be effectively implemented.
- Once implemented, it also imposes a high system management overhead due to the need to constantly update object and account labels to accommodate new data, new users and changes in the categorization and classification of existing users.

Discretionary Access Control DAC



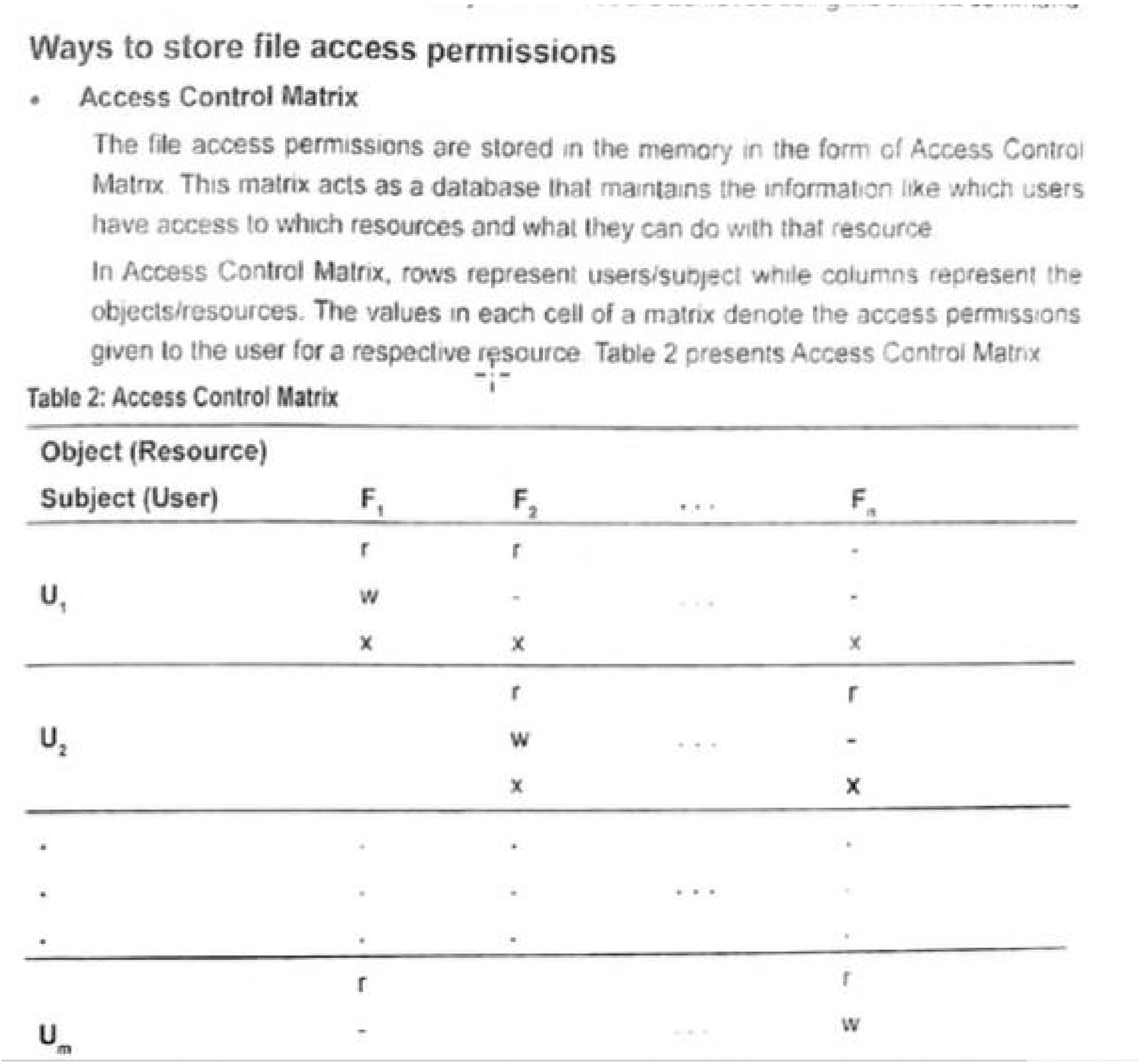
Discretionary Access Control (DAC)

In Discretionary Access Control (DAC), the owner of a resource decides how it should be shared. The owner can choose to give read, write, execute, delete, search and create access rights to other users. This is the most common model in large systems. The file permissions set in the Unix operating system is an example of DAC. Table 1 represents notations used for file access permissions in Unix operating system:

Table 1: Notations for File Permissions

r	Permission to read a file
	Permission to read a directory (also requires 'x')
w	Permission to delete or modify a file
	Permission to delete or modify files in a directory
x	Permission to execute a file/script
	Permission to read a directory (also requires 'r')
s	Set user or group ID on execution

Ways to store file access permissions



Ways to store file access permissions

- **Access Control Matrix**

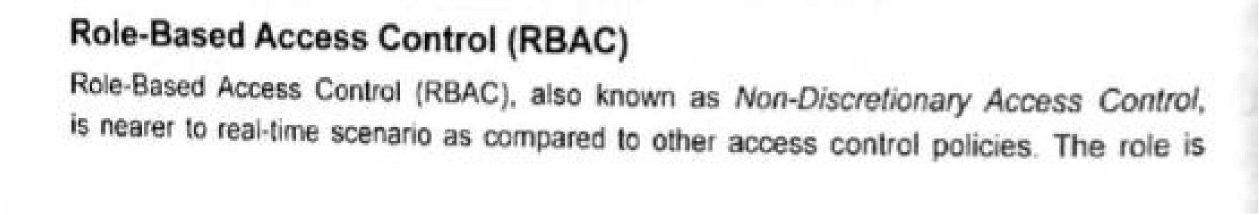
  The file access permissions are stored in the memory in the form of Access Control Matrix. This matrix acts as a database that maintains the information like which users have access to which resources and what they can do with that resource.

  In Access Control Matrix, rows represent users/subject while columns represent the objects/resources. The values in each cell of a matrix denote the access permissions given to the user for a respective resource. Table 2 presents Access Control Matrix

Table 2: Access Control Matrix

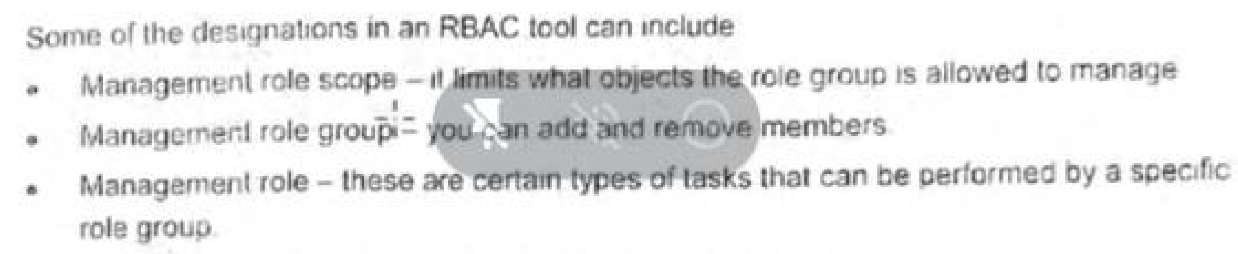
Object (Resource)				
Subject (User)	**F_1**	**F_2**	. . .	**F_n**
U_1	r w x	r - x	. . .	- - x
U_2		r w x	. . .	r - x
. . .	. . .	. . .	. . .	. . .
U_m	r -		. . .	r w

Role Based Access Control RBAC



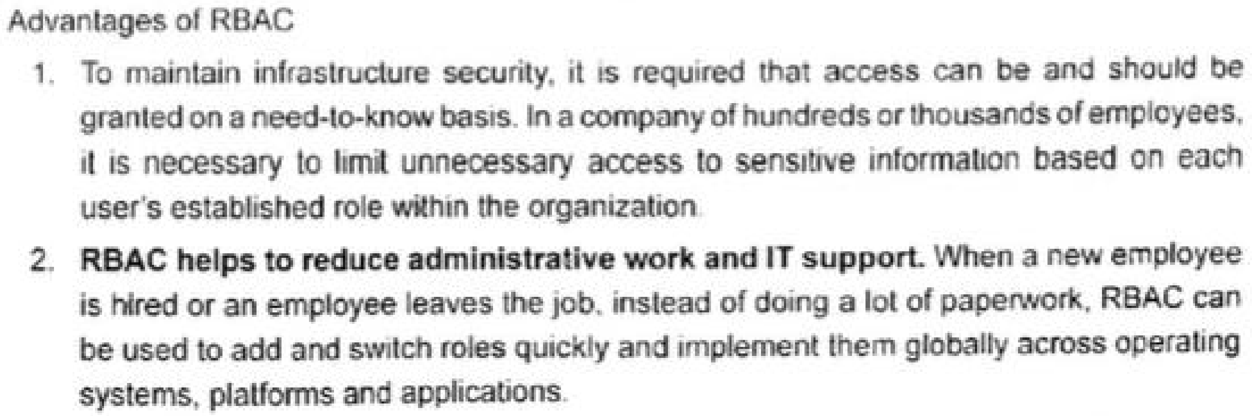
Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC), also known as *Non-Discretionary Access Control*, is nearer to real-time scenario as compared to other access control policies. The role is



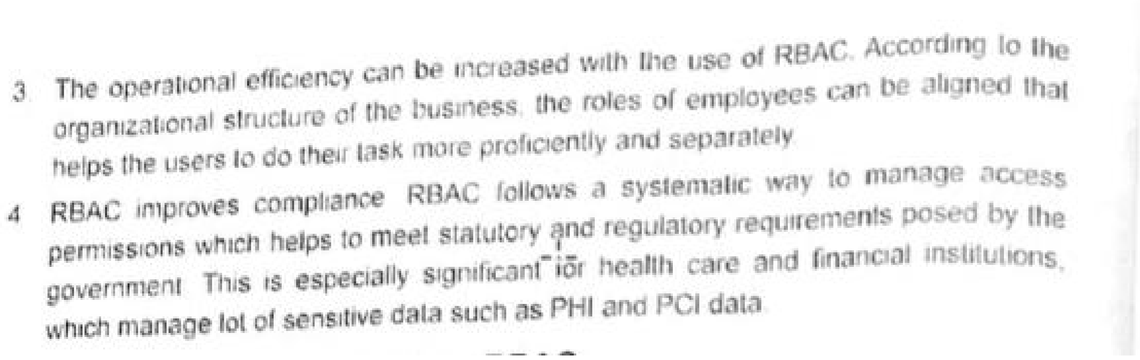
Some of the designations in an RBAC tool can include

- Management role scope – it limits what objects the role group is allowed to manage
- Management role group – you can add and remove members
- Management role – these are certain types of tasks that can be performed by a specific role group.



Advantages of RBAC

1. To maintain infrastructure security, it is required that access can be and should be granted on a need-to-know basis. In a company of hundreds or thousands of employees, it is necessary to limit unnecessary access to sensitive information based on each user's established role within the organization.
2. **RBAC helps to reduce administrative work and IT support.** When a new employee is hired or an employee leaves the job, instead of doing a lot of paperwork, RBAC can be used to add and switch roles quickly and implement them globally across operating systems, platforms and applications.



3. The operational efficiency can be increased with the use of RBAC. According to the organizational structure of the business, the roles of employees can be aligned that helps the users to do their task more proficiently and separately.
4. RBAC improves compliance RBAC follows a systematic way to manage access permissions which helps to meet statutory and regulatory requirements posed by the government This is especially significant for health care and financial institutions, which manage lot of sensitive data such as PHI and PCI data.

Attribute Based Access Control ABAC

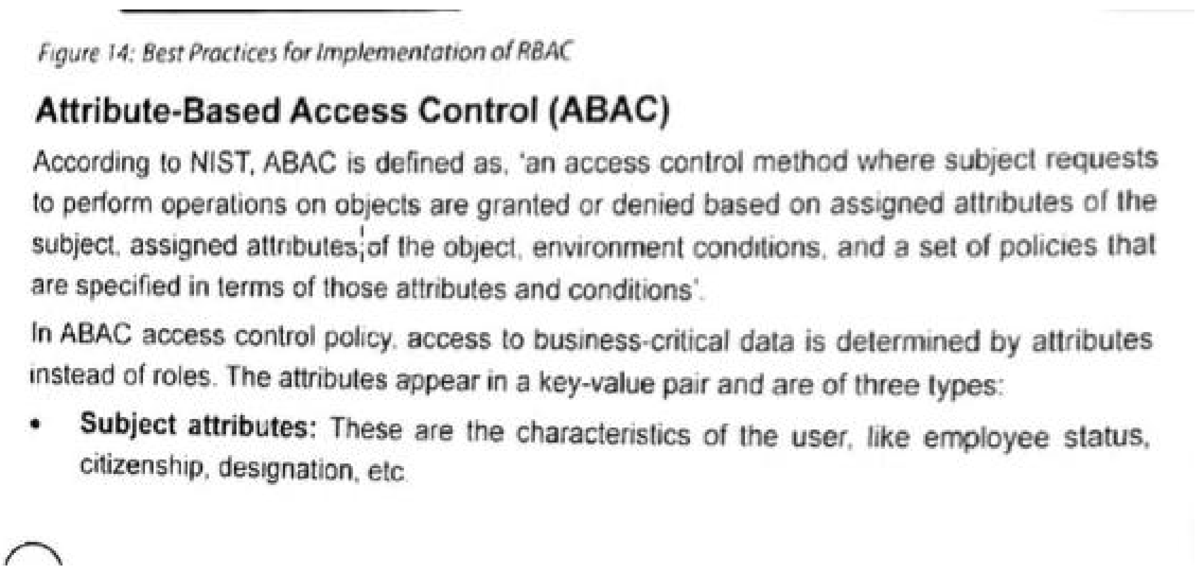


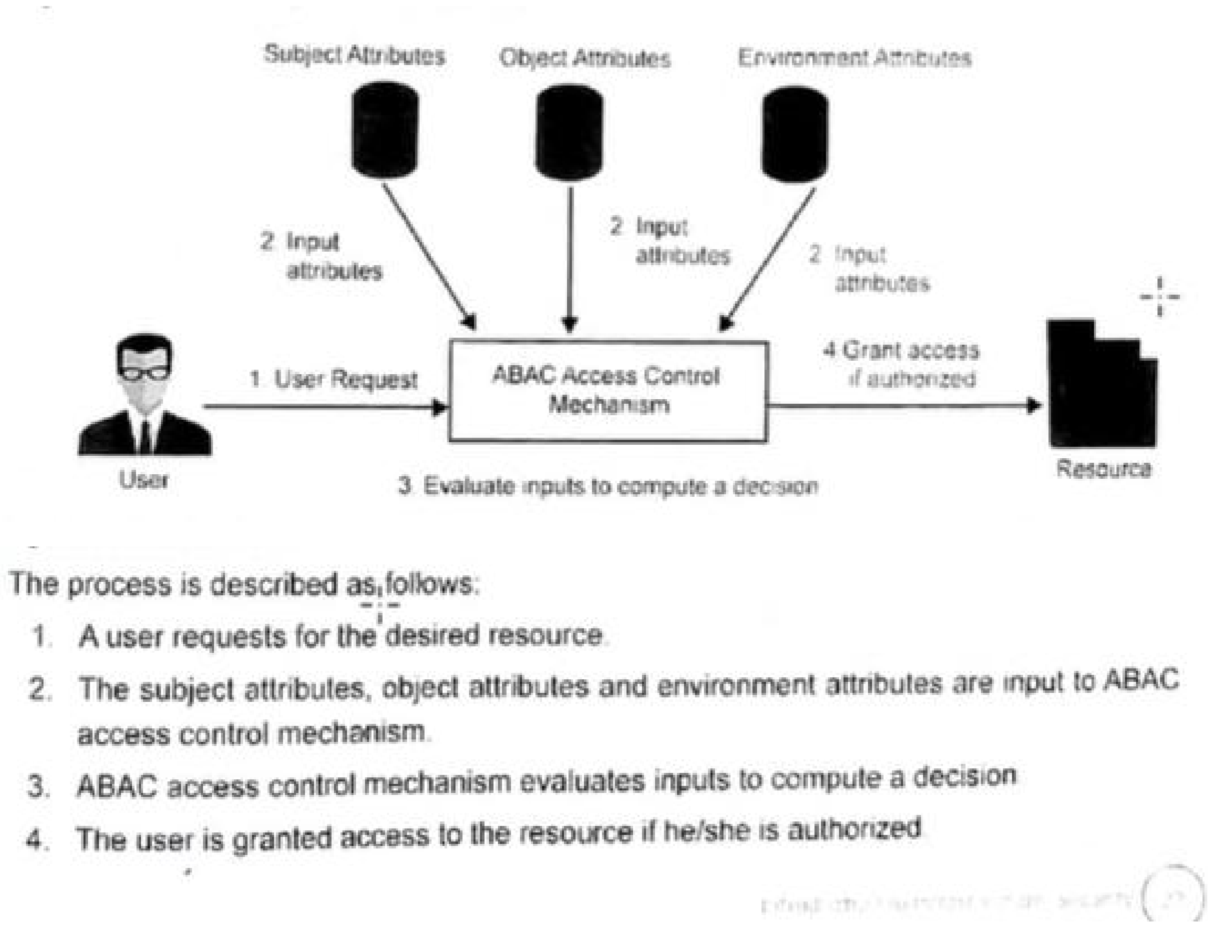
Figure 14: Best Practices for Implementation of RBAC

Attribute-Based Access Control (ABAC)

According to NIST, ABAC is defined as, 'an access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions'.

In ABAC access control policy, access to business-critical data is determined by attributes instead of roles. The attributes appear in a key-value pair and are of three types:

- **Subject attributes:** These are the characteristics of the user, like employee status, citizenship, designation, etc.

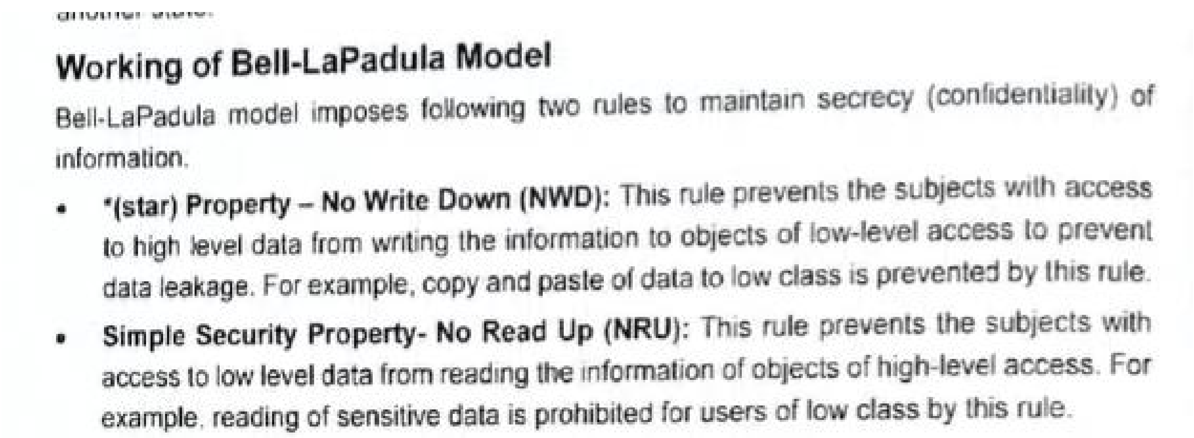


The process is described as follows:

1. A user requests for the desired resource.
2. The subject attributes, object attributes and environment attributes are input to ABAC access control mechanism.
3. ABAC access control mechanism evaluates inputs to compute a decision.
4. The user is granted access to the resource if he/she is authorized.

Bell LaPadula Model

Works on confidentiality



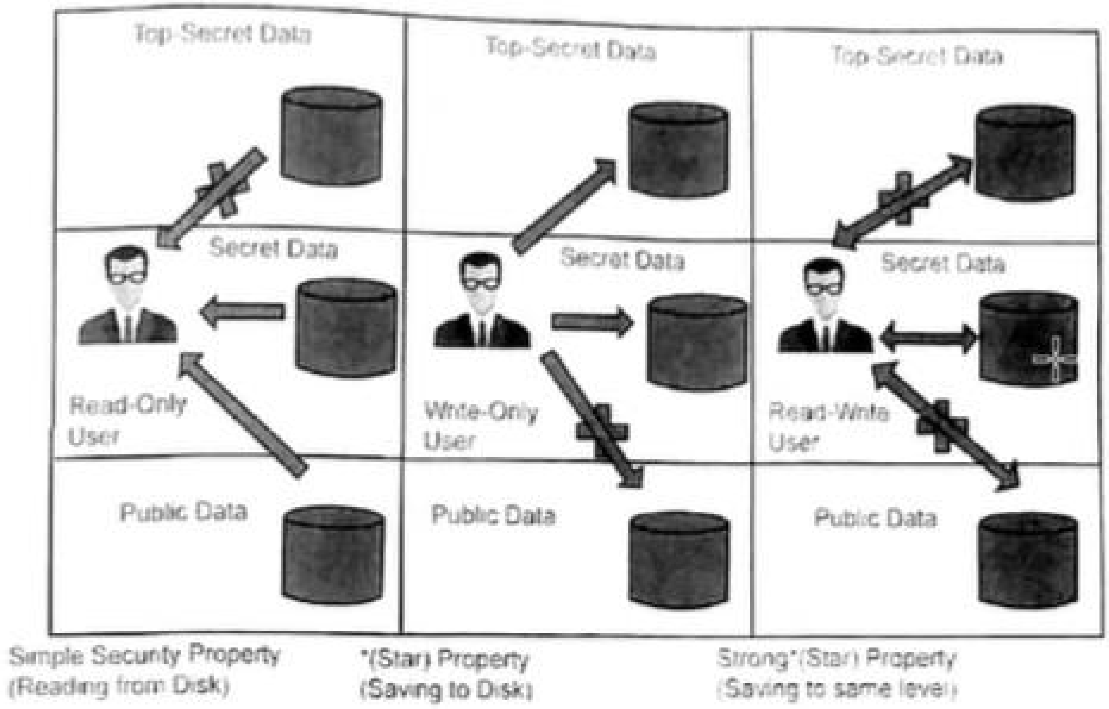
another state.

Working of Bell-LaPadula Model

Bell-LaPadula model imposes following two rules to maintain secrecy (confidentiality) of information.

- ***(star) Property – No Write Down (NWD):** This rule prevents the subjects with access to high level data from writing the information to objects of low-level access to prevent data leakage. For example, copy and paste of data to low class is prevented by this rule.
- **Simple Security Property- No Read Up (NRU):** This rule prevents the subjects with access to low level data from reading the information of objects of high-level access. For example, reading of sensitive data is prohibited for users of low class by this rule.

... of the Bell-LaPadula model in the form of a lattice:



- Defence against Trojan Horse Attacks with Bell-LaPadula Model

  Bell-LaPadula model resists trojan horse attacks that other access control policies cannot do. A trojan horse is a program that pretends to be useful but has hidden nefarious activity. For example, the attacker designs an interesting game or some useful utility and offers it to the victim to be used in his/her spare time. When the victim starts using it, a trojan horse program hidden inside it gets triggered. As this program is being used by the victim user, it automatically gets the same access rights which the victim user has. A trojan horse takes a benefit of this and creates a file in the background to which the attacker has access. Then, the trojan horse program writes the confidential information to this file which an attacker can easily gain. DAC is inoperable in this scenario because all files being accessed by that utility/game are the files that the victim user is authorized to access.

In comparison, the same trojan horse program fails when it is run in an environment where Bell-LaPadula model is enforced. The *(Star) Property (No Write Down) in particular, raises an alarm when trojan horse program attempts to write confidential information *from a file* of high level of security to the file of low level of security; thus preventing from data leakage to happen.

Applications of Bell-LaPadula Model

- The Bell-LaPadula Model was put to practical use in the development of Multics, a multi-user operating system in which computing processes were interpreted as subjects, and the likes of memory segments and input/output devices were defined as objects.
- While secure and operating in accordance with BLM principles, Multics proved too cumbersome and problematic for some project members, who took off on a tangent and designed the simpler and more commercially viable Unix.

Limitations of Bell-LaPadula Model

In spite of providing effective access control mechanism, this model suffers from certain drawbacks:

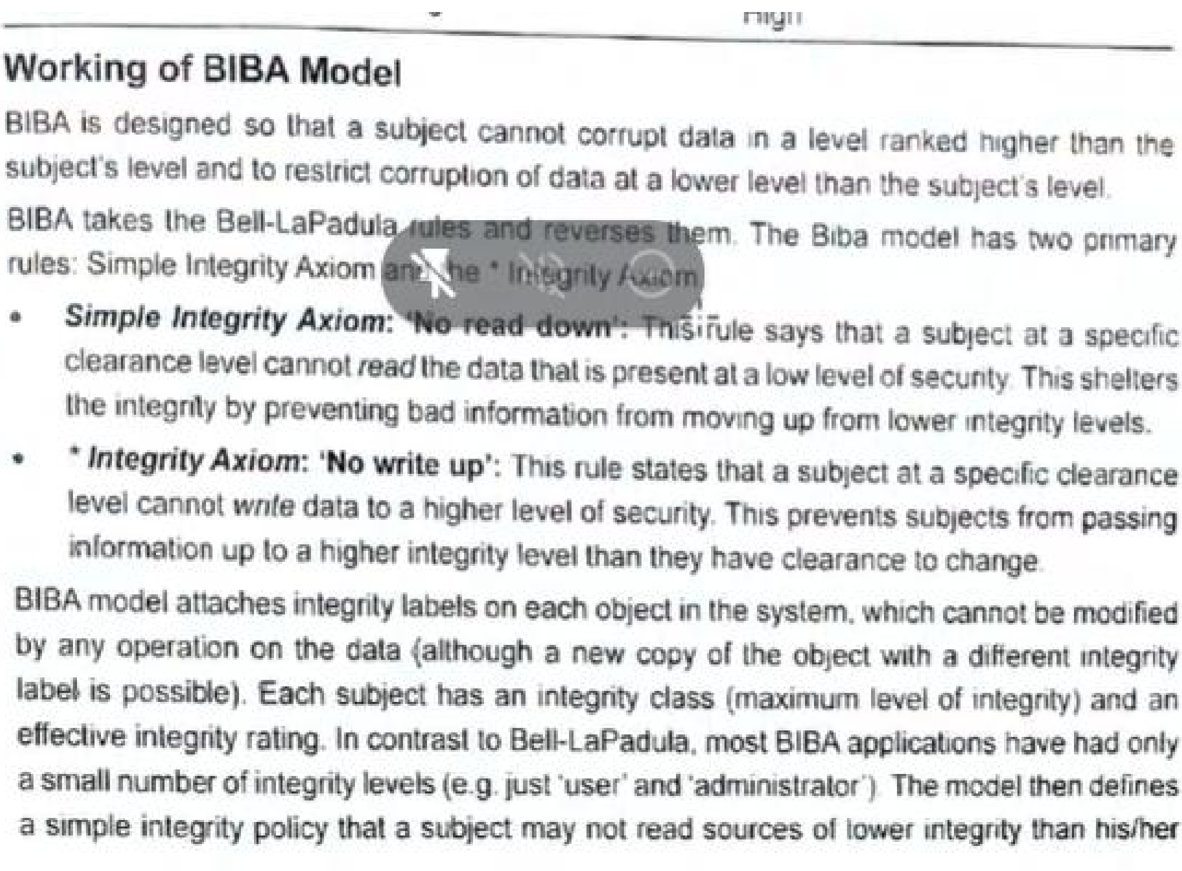
- This model emphasizes only confidentiality and does not address data integrity.
- The model is primarily intended for systems having largely static security levels. There are no inherent policies for changing access rights.
- There exist covert channels by which a subject at a lower clearance may perceive the existence of high-level objects through the simple act of the subject's being denied access to them.

Table 6: Integrity Levels, Weights and Impact of Loss

Level of Integrity Required	Weight	Impact of Loss
Not Applicable	0	Low
Approximate	3	Moderate
Exact	6	High

Working of BIBA Model

Biba Model



High

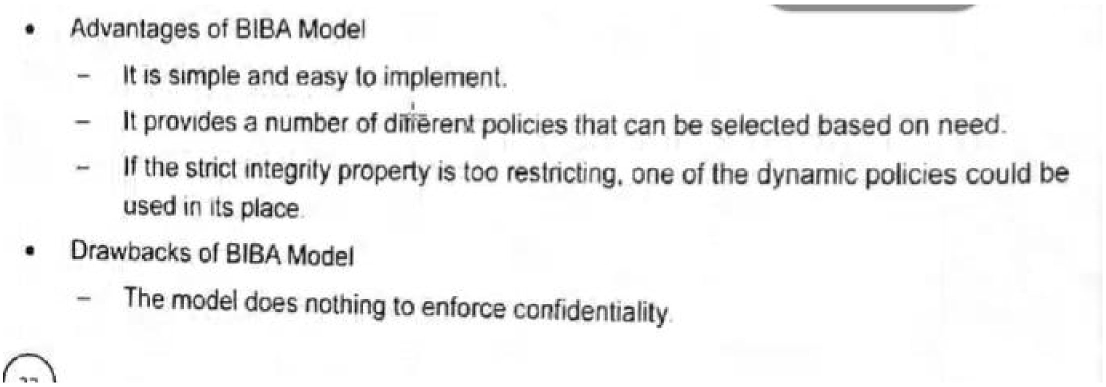
Working of BIBA Model

BIBA is designed so that a subject cannot corrupt data in a level ranked higher than the subject's level and to restrict corruption of data at a lower level than the subject's level.

BIBA takes the Bell-LaPadula rules and reverses them. The Biba model has two primary rules: Simple Integrity Axiom and the * Integrity Axiom.

- ***Simple Integrity Axiom*: 'No read down'**: This rule says that a subject at a specific clearance level cannot *read* the data that is present at a low level of security. This shelters the integrity by preventing bad information from moving up from lower integrity levels.
- **** Integrity Axiom*: 'No write up'**: This rule states that a subject at a specific clearance level cannot *write* data to a higher level of security. This prevents subjects from passing information up to a higher integrity level than they have clearance to change.

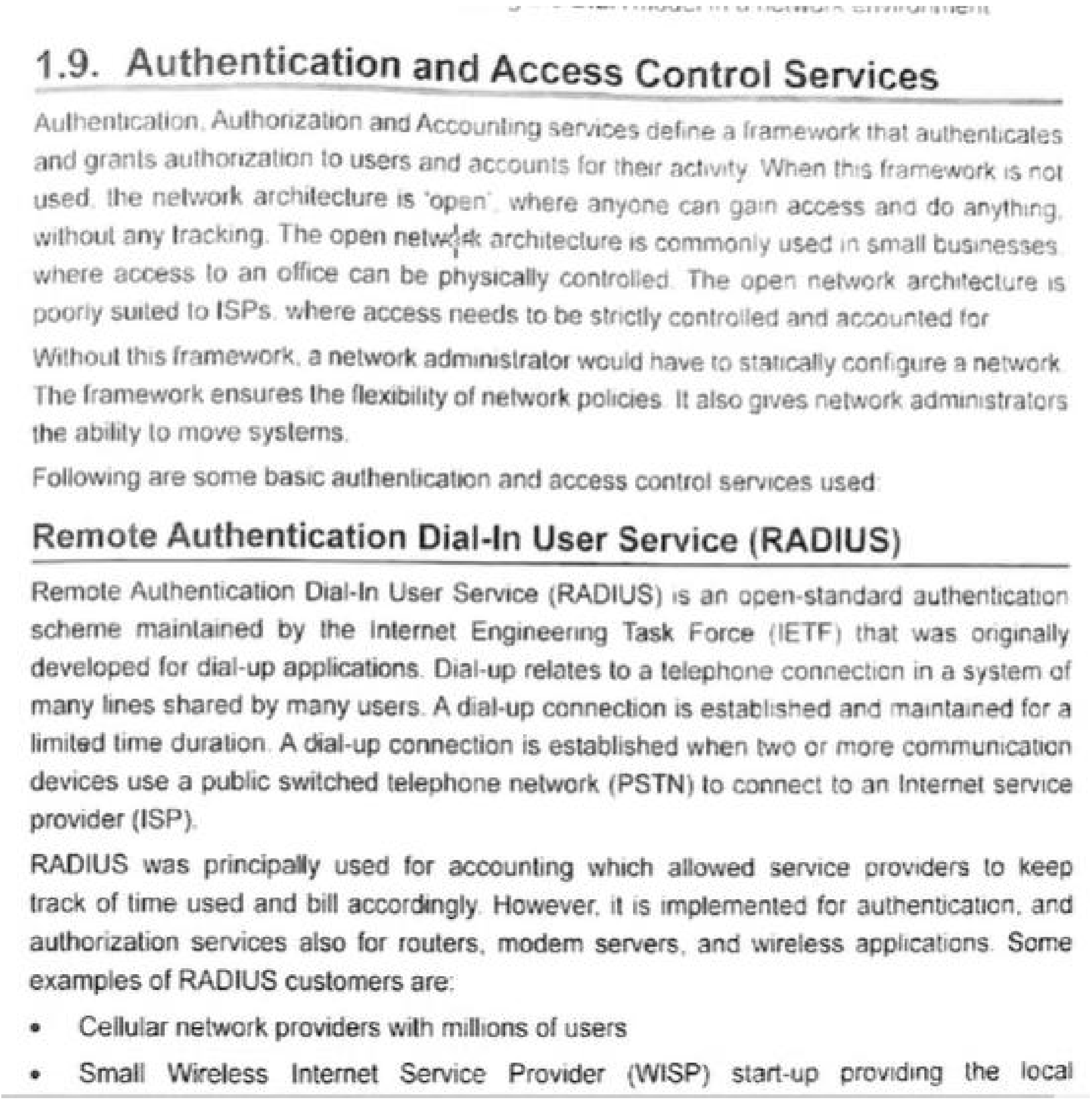
BIBA model attaches integrity labels on each object in the system, which cannot be modified by any operation on the data (although a new copy of the object with a different integrity label is possible). Each subject has an integrity class (maximum level of integrity) and an effective integrity rating. In contrast to Bell-LaPadula, most BIBA applications have had only a small number of integrity levels (e.g. just 'user' and 'administrator'). The model then defines a simple integrity policy that a subject may not read sources of lower integrity than his/her



- Advantages of BIBA Model
  - It is simple and easy to implement.
  - It provides a number of different policies that can be selected based on need.
  - If the strict integrity property is too restricting, one of the dynamic policies could be used in its place.
- Drawbacks of BIBA Model
  - The model does nothing to enforce confidentiality.

Authentication AND Access Control Services

Remote Authentication Dail in User Service (RADIUS)



1.9. Authentication and Access Control Services

Authentication, Authorization and Accounting services define a framework that authenticates and grants authorization to users and accounts for their activity. When this framework is not used, the network architecture is 'open', where anyone can gain access and do anything, without any tracking. The open network architecture is commonly used in small businesses where access to an office can be physically controlled. The open network architecture is poorly suited to ISPs, where access needs to be strictly controlled and accounted for.

Without this framework, a network administrator would have to statically configure a network. The framework ensures the flexibility of network policies. It also gives network administrators the ability to move systems.

Following are some basic authentication and access control services used:

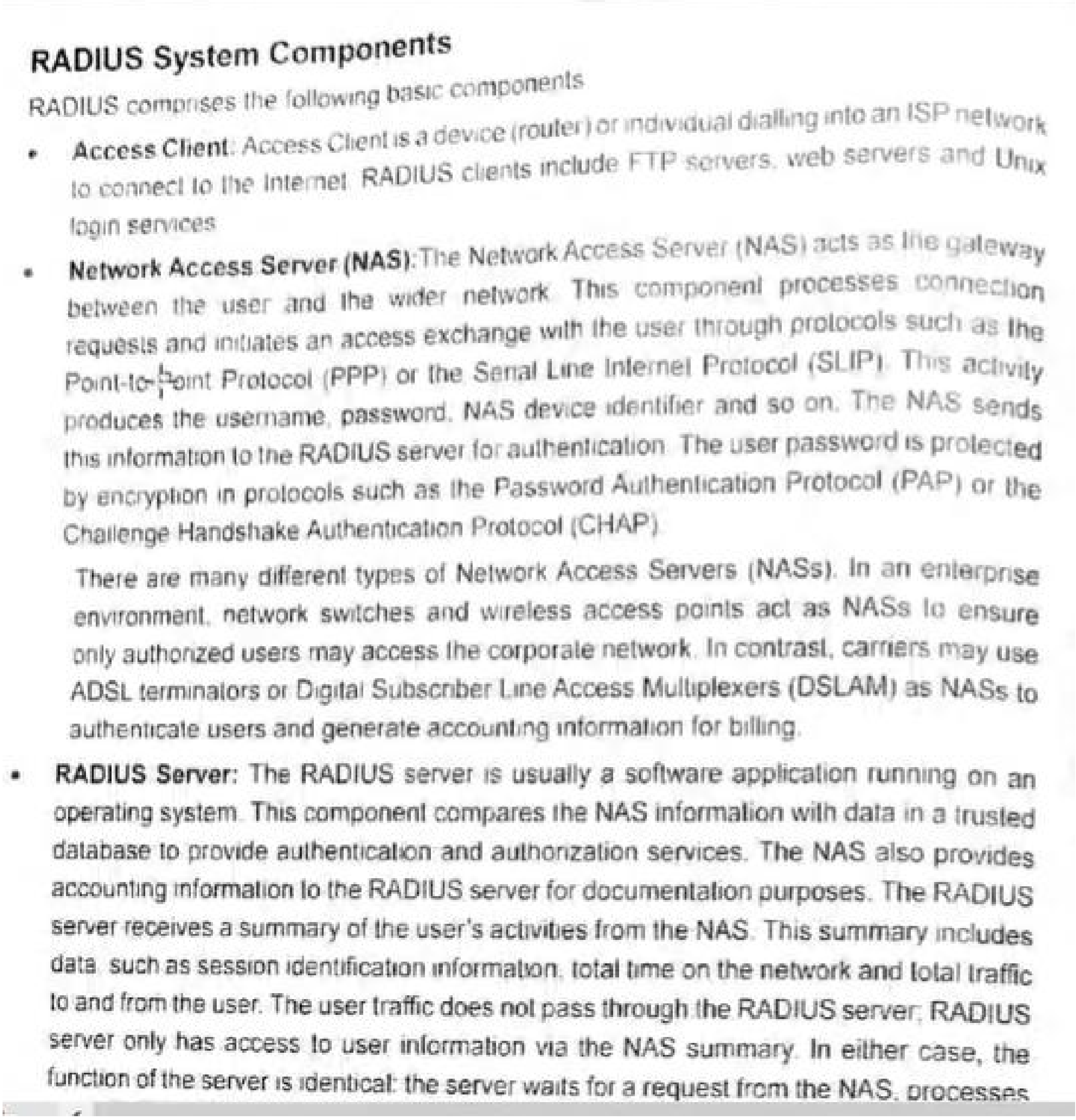
Remote Authentication Dial-In User Service (RADIUS)

Remote Authentication Dial-In User Service (RADIUS) is an open-standard authentication scheme maintained by the Internet Engineering Task Force (IETF) that was originally developed for dial-up applications. Dial-up relates to a telephone connection in a system of many lines shared by many users. A dial-up connection is established and maintained for a limited time duration. A dial-up connection is established when two or more communication devices use a public switched telephone network (PSTN) to connect to an Internet service provider (ISP).

RADIUS was principally used for accounting which allowed service providers to keep track of time used and bill accordingly. However, it is implemented for authentication, and authorization services also for routers, modem servers, and wireless applications. Some examples of RADIUS customers are:

- Cellular network providers with millions of users
- Small Wireless Internet Service Provider (WISP) start-up providing the local

- Small Wireless Internet Service Provider (WISP) start-up providing the local neighbourhood with Internet connectivity.
- Enterprise networks implementing Network Access Control (NAC) using 802.1x to secure access to their network.
- Universities which give WiFi access to their students and staff.



RADIUS System Components

RADIUS comprises the following basic components

- **Access Client**: Access Client is a device (router) or individual dialling into an ISP network to connect to the Internet. RADIUS clients include FTP servers, web servers and Unix login services
- **Network Access Server (NAS)**:The Network Access Server (NAS) acts as the gateway between the user and the wider network. This component processes connection requests and initiates an access exchange with the user through protocols such as the Point-to-Point Protocol (PPP) or the Serial Line Internet Protocol (SLIP). This activity produces the username, password, NAS device identifier and so on. The NAS sends this information to the RADIUS server for authentication. The user password is protected by encryption in protocols such as the Password Authentication Protocol (PAP) or the Challenge Handshake Authentication Protocol (CHAP).

  There are many different types of Network Access Servers (NASs). In an enterprise environment, network switches and wireless access points act as NASs to ensure only authorized users may access the corporate network. In contrast, carriers may use ADSL terminators or Digital Subscriber Line Access Multiplexers (DSLAM) as NASs to authenticate users and generate accounting information for billing.
- **RADIUS Server:** The RADIUS server is usually a software application running on an operating system. This component compares the NAS information with data in a trusted database to provide authentication and authorization services. The NAS also provides accounting information to the RADIUS server for documentation purposes. The RADIUS server receives a summary of the user's activities from the NAS. This summary includes data such as session identification information, total time on the network and total traffic to and from the user. The user traffic does not pass through the RADIUS server; RADIUS server only has access to user information via the NAS summary. In either case, the function of the server is identical: the server waits for a request from the NAS, processes

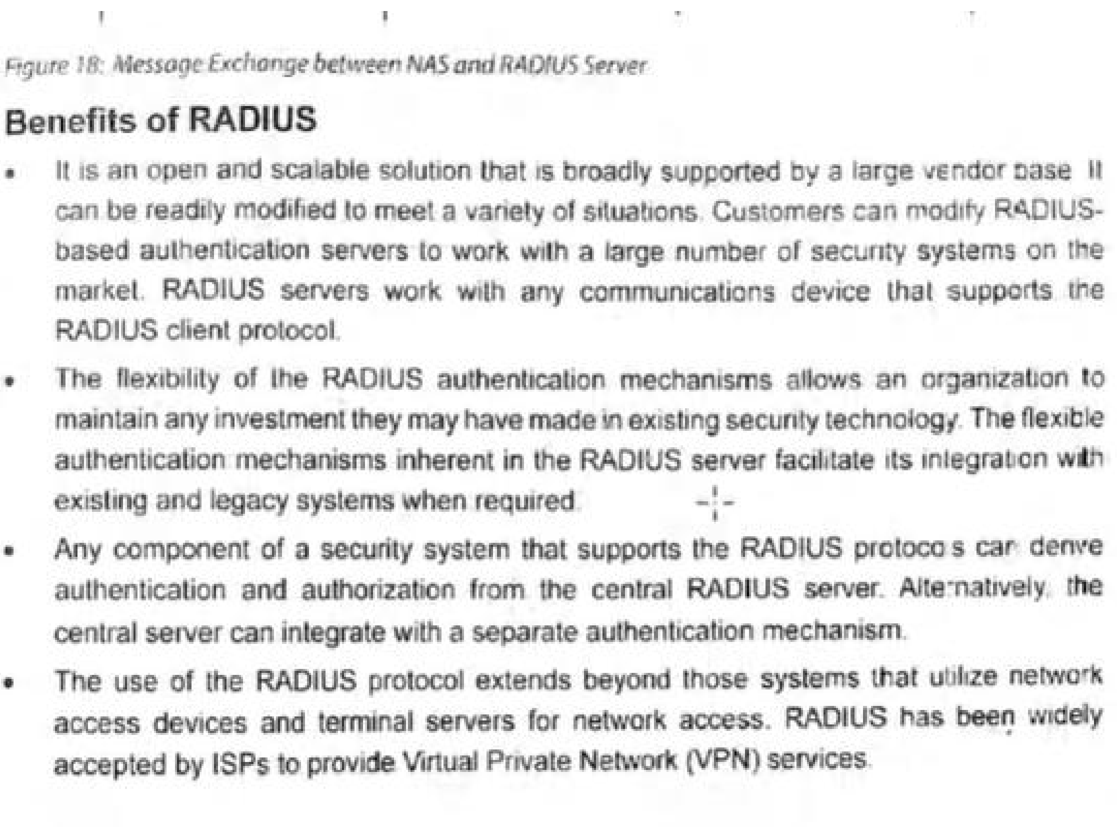
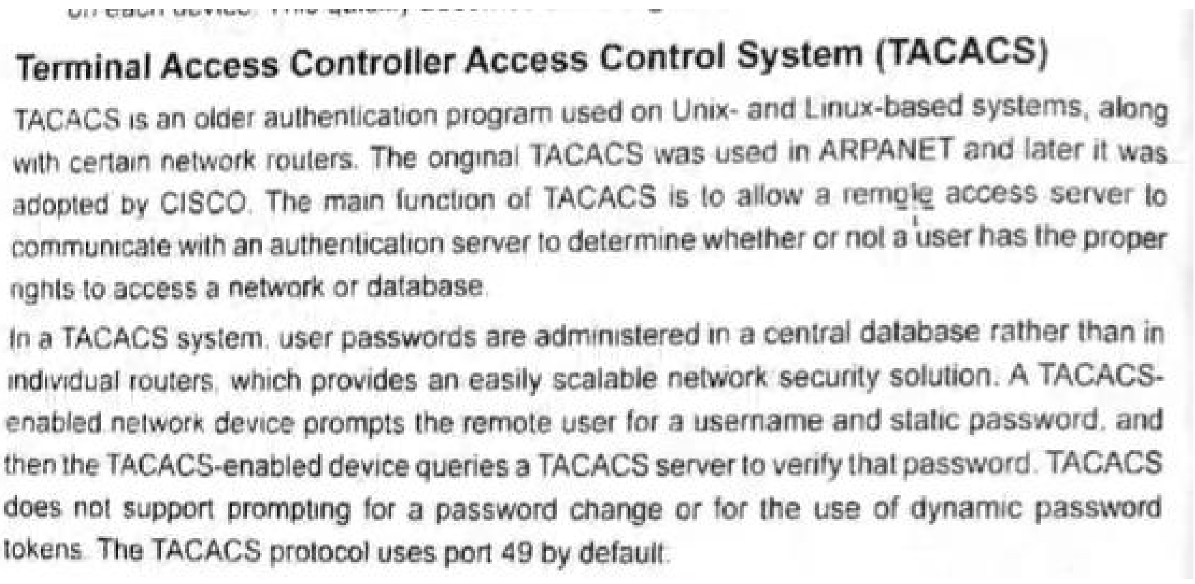


Figure 18: Message Exchange between NAS and RADIUS Server

Benefits of RADIUS

- It is an open and scalable solution that is broadly supported by a large vendor base. It can be readily modified to meet a variety of situations. Customers can modify RADIUS-based authentication servers to work with a large number of security systems on the market. RADIUS servers work with any communications device that supports the RADIUS client protocol.
- The flexibility of the RADIUS authentication mechanisms allows an organization to maintain any investment they may have made in existing security technology. The flexible authentication mechanisms inherent in the RADIUS server facilitate its integration with existing and legacy systems when required.
- Any component of a security system that supports the RADIUS protocols can derive authentication and authorization from the central RADIUS server. Alternatively, the central server can integrate with a separate authentication mechanism.
- The use of the RADIUS protocol extends beyond those systems that utilize network access devices and terminal servers for network access. RADIUS has been widely accepted by ISPs to provide Virtual Private Network (VPN) services.

Terminal Access Controller Access Control System (TACACS)



Terminal Access Controller Access Control System (TACACS)

TACACS is an older authentication program used on Unix- and Linux-based systems, along with certain network routers. The original TACACS was used in ARPANET and later it was adopted by CISCO. The main function of TACACS is to allow a remote access server to communicate with an authentication server to determine whether or not a user has the proper rights to access a network or database.

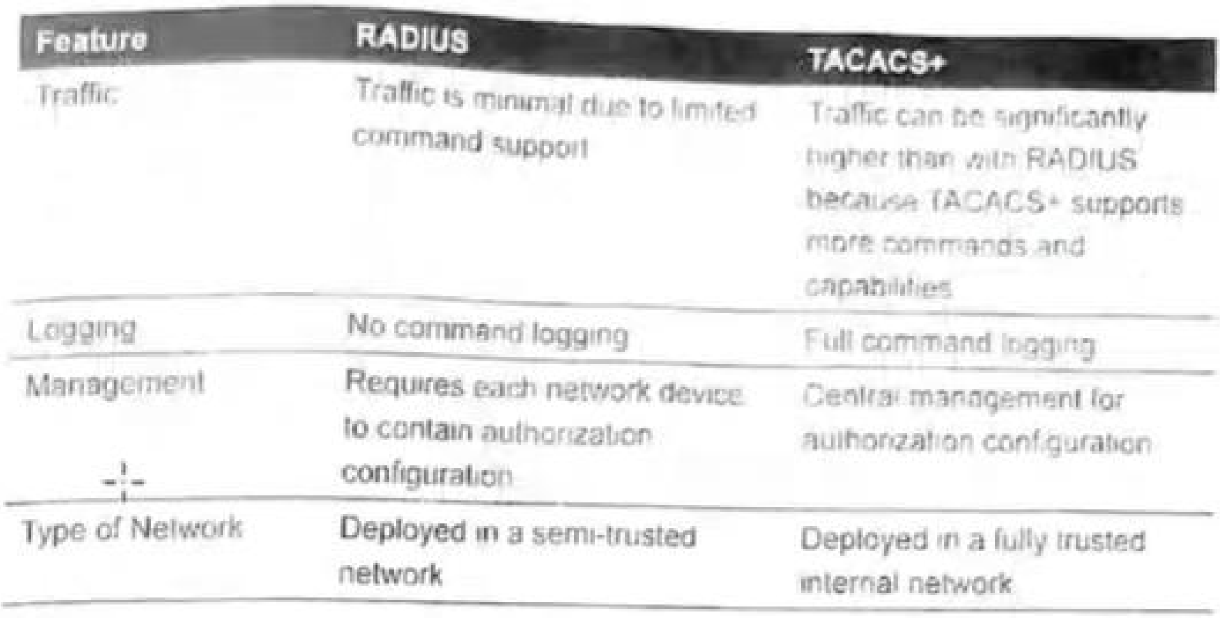
In a TACACS system, user passwords are administered in a central database rather than in individual routers, which provides an easily scalable network security solution. A TACACS-enabled network device prompts the remote user for a username and static password, and then the TACACS-enabled device queries a TACACS server to verify that password. TACACS does not support prompting for a password change or for the use of dynamic password tokens. The TACACS protocol uses port 49 by default.

Terminal Access Controller Access Control System Plus (TACACS+)

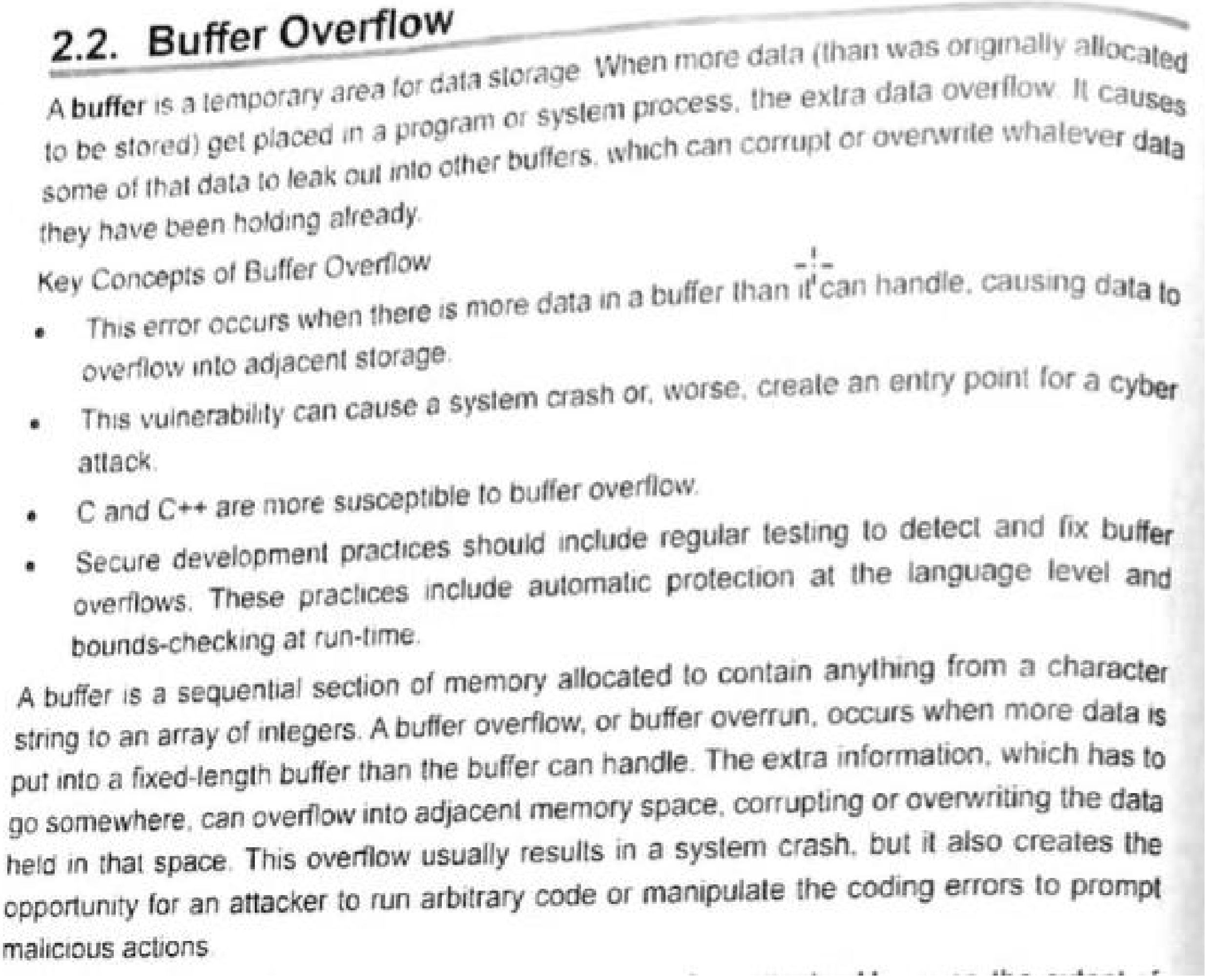
TACACS+ is a authentication protocol, developed by Cisco, which superseded TACACS and it provides access control for routers, network access servers, and many other networked computing devices through one or more centralized servers. It provides separate Authentication, Authorization and Accounting services for server access.

Table 5: Comparison between RADIUS and TACACS+

Feature	RADIUS	TACACS+
Transport Protocol	UDP	TCP
Port	1645/1646 or 1812/1813	49
Communication & Encryption	Less secure – only encrypts password	More secure-encrypts the whole packet including username, password and attributes
Can Authenticate Network Devices	No	Yes
Primary Use	Network Access	Device Administration
Designed for	AAA users/clients	AAA Administrator
Framework	Combines authentication and authorization	Separates authentication from authorization and accounting
Protocol	Limited support for certain protocols	Full multiprotocol support
Vendor	Vendor implementations always differ Interoperability can be an issue.	Specific to Cisco equipment



Feature	RADIUS	TACACS+
Traffic	Traffic is minimal due to limited command support	Traffic can be significantly higher than with RADIUS because TACACS+ supports more commands and capabilities
Logging	No command logging	Full command logging
Management	Requires each network device to contain authorization configuration	Central management for authorization configuration
Type of Network	Deployed in a semi-trusted network	Deployed in a fully trusted internal network



2.2. Buffer Overflow

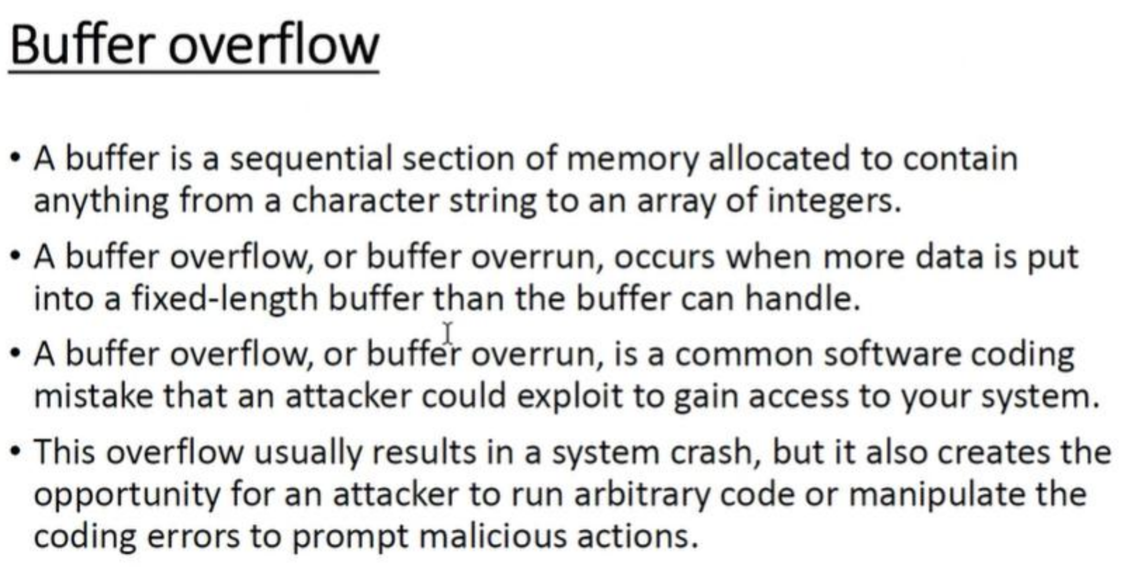
A **buffer** is a temporary area for data storage. When more data (than was originally allocated to be stored) get placed in a program or system process, the extra data overflow. It causes some of that data to leak out into other buffers, which can corrupt or overwrite whatever data they have been holding already.

Key Concepts of Buffer Overflow

- This error occurs when there is more data in a buffer than it can handle, causing data to overflow into adjacent storage.
- This vulnerability can cause a system crash or, worse, create an entry point for a cyber attack.
- C and C++ are more susceptible to buffer overflow.
- Secure development practices should include regular testing to detect and fix buffer overflows. These practices include automatic protection at the language level and bounds-checking at run-time.

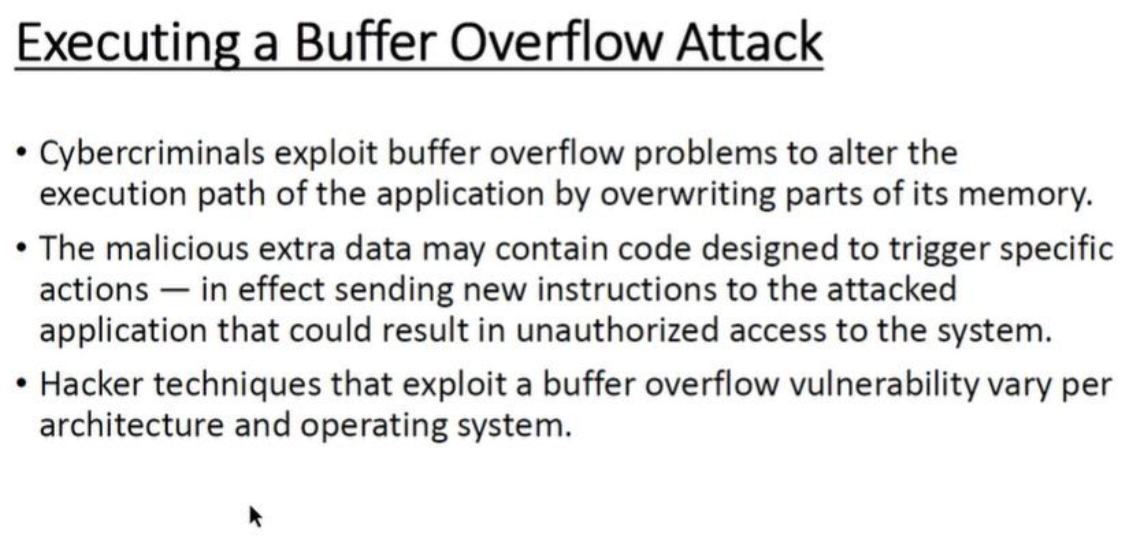
A buffer is a sequential section of memory allocated to contain anything from a character string to an array of integers. A buffer overflow, or buffer overrun, occurs when more data is put into a fixed-length buffer than the buffer can handle. The extra information, which has to go somewhere, can overflow into adjacent memory space, corrupting or overwriting the data held in that space. This overflow usually results in a system crash, but it also creates the opportunity for an attacker to run arbitrary code or manipulate the coding errors to prompt malicious actions.

Buffer overflow



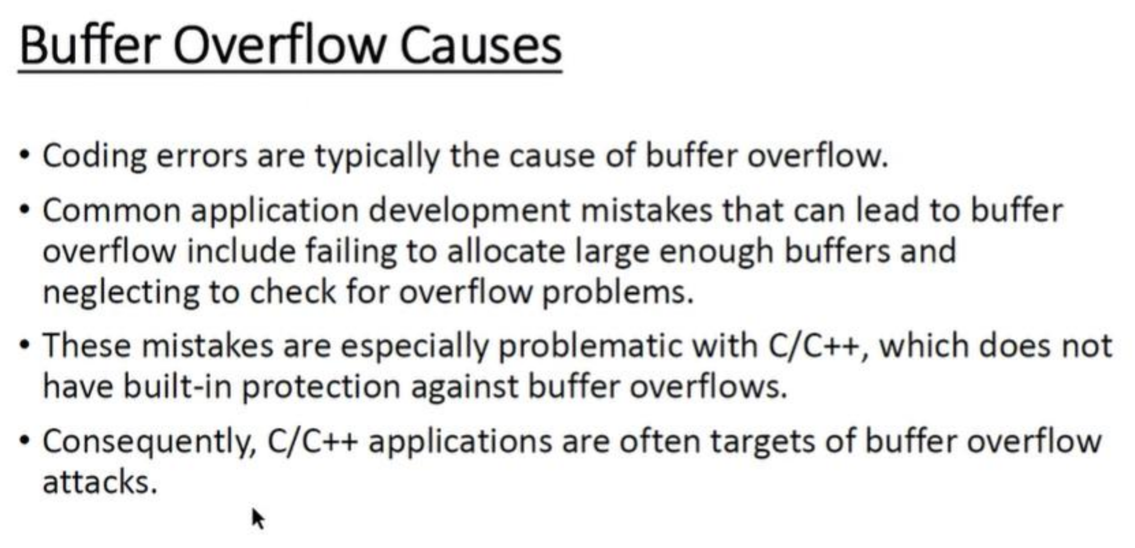
Buffer overflow

- A buffer is a sequential section of memory allocated to contain anything from a character string to an array of integers.
- A buffer overflow, or buffer overrun, occurs when more data is put into a fixed-length buffer than the buffer can handle.
- A buffer overflow, or buffer overrun, is a common software coding mistake that an attacker could exploit to gain access to your system.
- This overflow usually results in a system crash, but it also creates the opportunity for an attacker to run arbitrary code or manipulate the coding errors to prompt malicious actions.



Executing a Buffer Overflow Attack

- Cybercriminals exploit buffer overflow problems to alter the execution path of the application by overwriting parts of its memory.
- The malicious extra data may contain code designed to trigger specific actions — in effect sending new instructions to the attacked application that could result in unauthorized access to the system.
- Hacker techniques that exploit a buffer overflow vulnerability vary per architecture and operating system.



Buffer Overflow Causes

- Coding errors are typically the cause of buffer overflow.
- Common application development mistakes that can lead to buffer overflow include failing to allocate large enough buffers and neglecting to check for overflow problems.
- These mistakes are especially problematic with C/C++, which does not have built-in protection against buffer overflows.
- Consequently, C/C++ applications are often targets of buffer overflow attacks.

Types of buffer overflows:

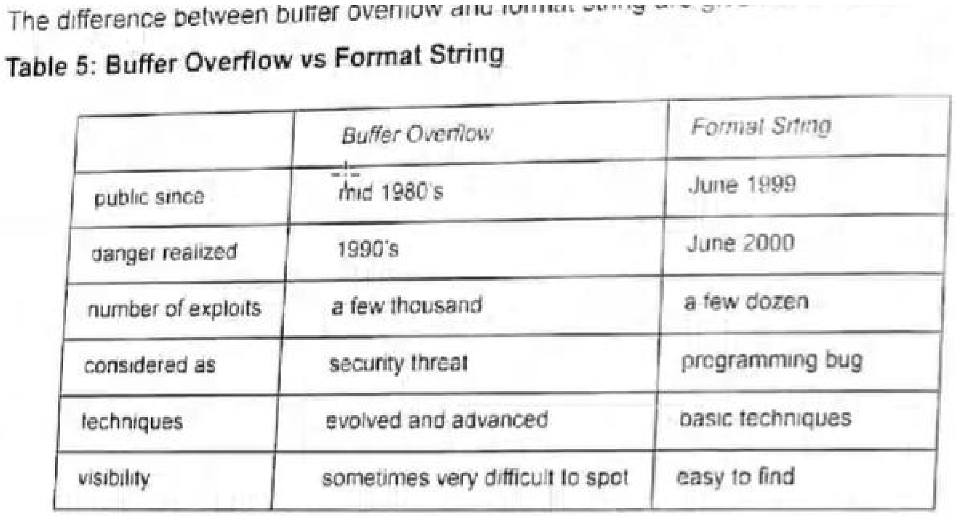
Stack-based and Heap-based.

- **Heap-based:**
  - Are difficult to execute and the least common of the two
  - attack an application by flooding the memory space reserved for a program.

- **Stack-based buffer overflows:**
  - Are more common among attackers
  - exploit applications and programs by using what is known as a stack: memory space used to store user input.

Avoid buffer overflow attacks

- To avoid buffer overflow attacks, the general advice that is given to programmers is to follow good programming practices.
  - Make sure that the memory auditing is done properly in the program using utilities like valgrind memcheck.
  - Use fgets() instead of gets().
  - Use strncmp() instead of strcmp(), strncpy() instead of strcpy()

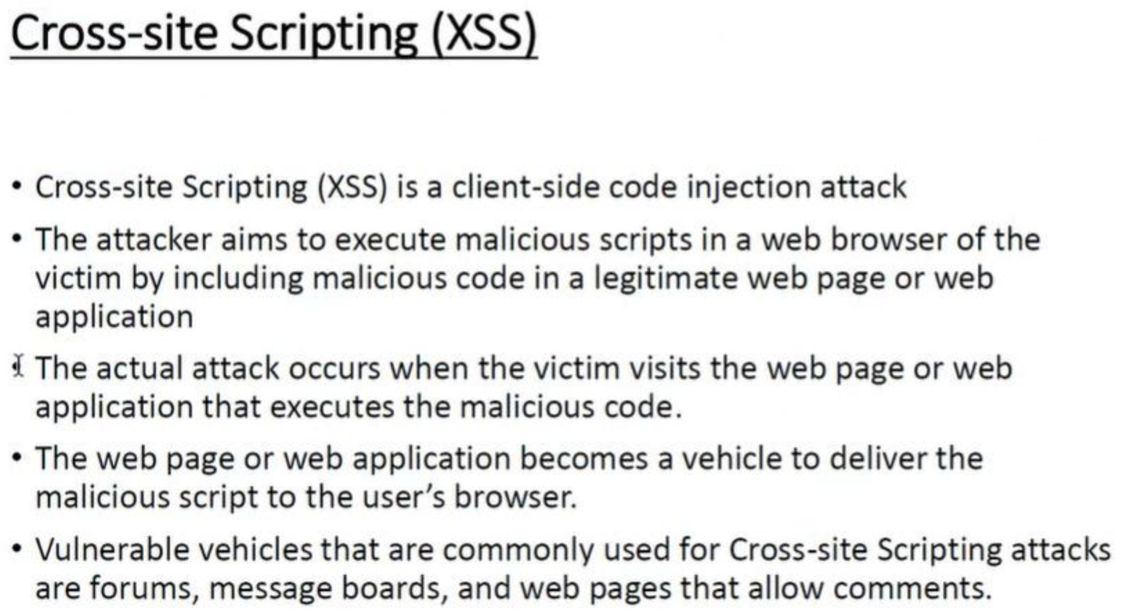


The difference between buffer overflow and format string are g

Table 5: Buffer Overflow vs Format String

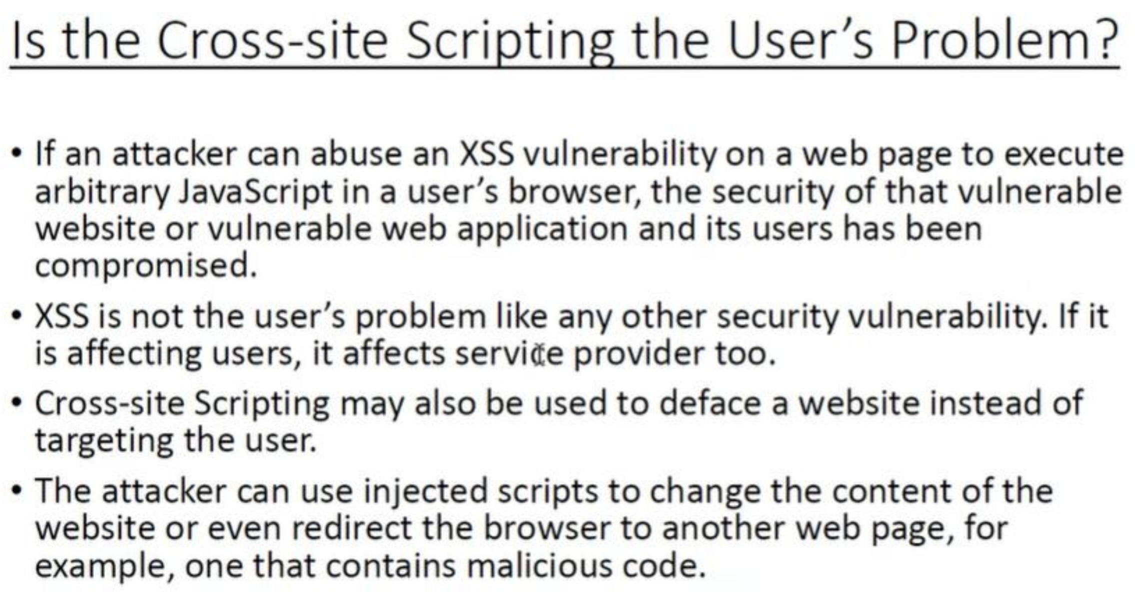
	Buffer Overflow	Format String
public since	mid 1980's	June 1999
danger realized	1990's	June 2000
number of exploits	a few thousand	a few dozen
considered as	security threat	programming bug
techniques	evolved and advanced	basic techniques
visibility	sometimes very difficult to spot	easy to find

Cross site Scripting



Cross-site Scripting (XSS)

- Cross-site Scripting (XSS) is a client-side code injection attack
- The attacker aims to execute malicious scripts in a web browser of the victim by including malicious code in a legitimate web page or web application
- The actual attack occurs when the victim visits the web page or web application that executes the malicious code.
- The web page or web application becomes a vehicle to deliver the malicious script to the user's browser.
- Vulnerable vehicles that are commonly used for Cross-site Scripting attacks are forums, message boards, and web pages that allow comments.



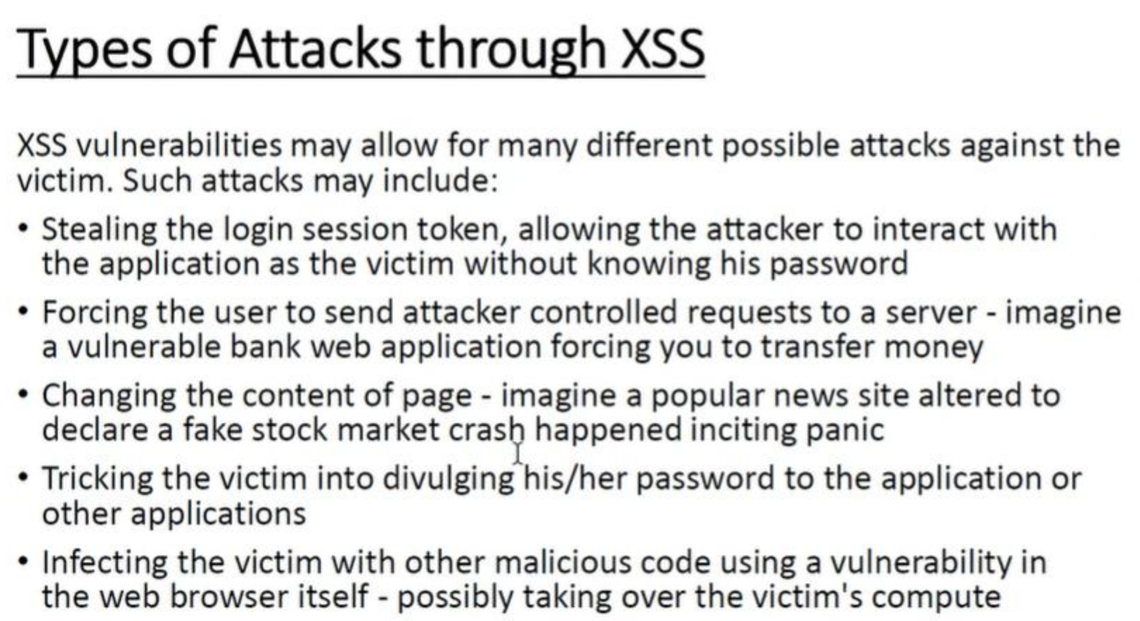
Is the Cross-site Scripting the User's Problem?

- If an attacker can abuse an XSS vulnerability on a web page to execute arbitrary JavaScript in a user's browser, the security of that vulnerable website or vulnerable web application and its users has been compromised.
- XSS is not the user's problem like any other security vulnerability. If it is affecting users, it affects service provider too.
- Cross-site Scripting may also be used to deface a website instead of targeting the user.
- The attacker can use injected scripts to change the content of the website or even redirect the browser to another web page, for example, one that contains malicious code.

How Cross-site Scripting Works

There are two stages to a typical XSS attack:

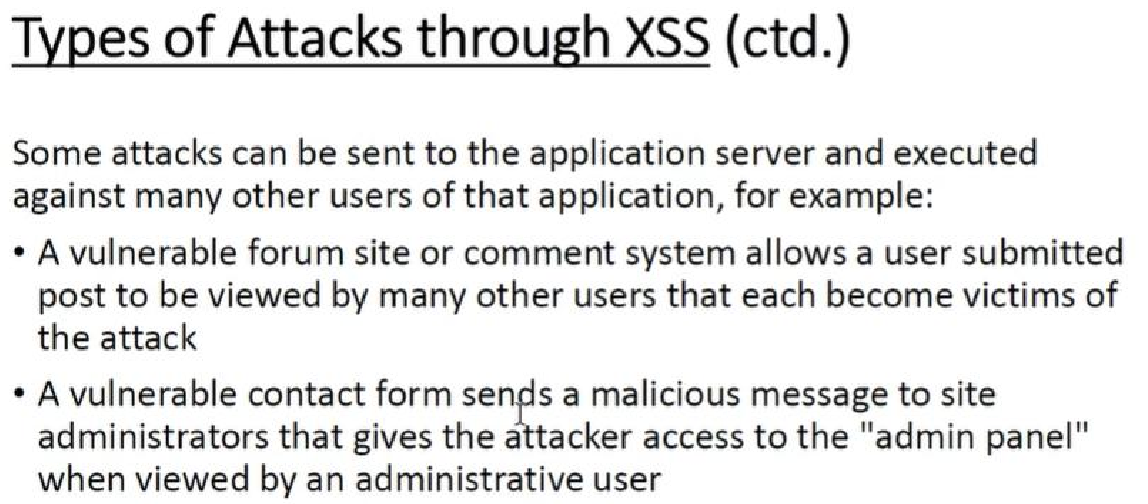
- To run malicious JavaScript code in a victim's browser, an attacker must first find a way to inject malicious code (payload) into a web page that the victim visits.
- After that, the victim must visit the web page with the malicious code. If the attack is directed at particular victims, the attacker can use social engineering and/or phishing to send a malicious URL to the victim.



Types of Attacks through XSS

XSS vulnerabilities may allow for many different possible attacks against the victim. Such attacks may include:

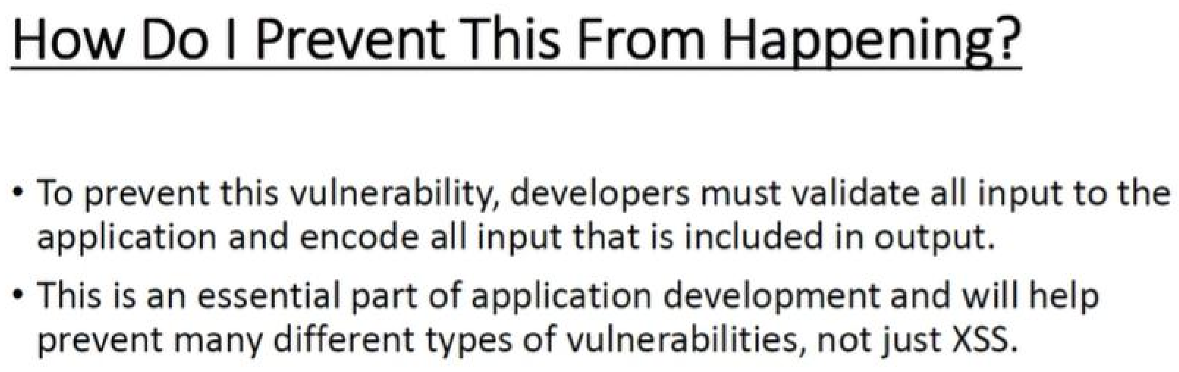
- Stealing the login session token, allowing the attacker to interact with the application as the victim without knowing his password
- Forcing the user to send attacker controlled requests to a server - imagine a vulnerable bank web application forcing you to transfer money
- Changing the content of page - imagine a popular news site altered to declare a fake stock market crash happened inciting panic
- Tricking the victim into divulging his/her password to the application or other applications
- Infecting the victim with other malicious code using a vulnerability in the web browser itself - possibly taking over the victim's compute



Types of Attacks through XSS (ctd.)

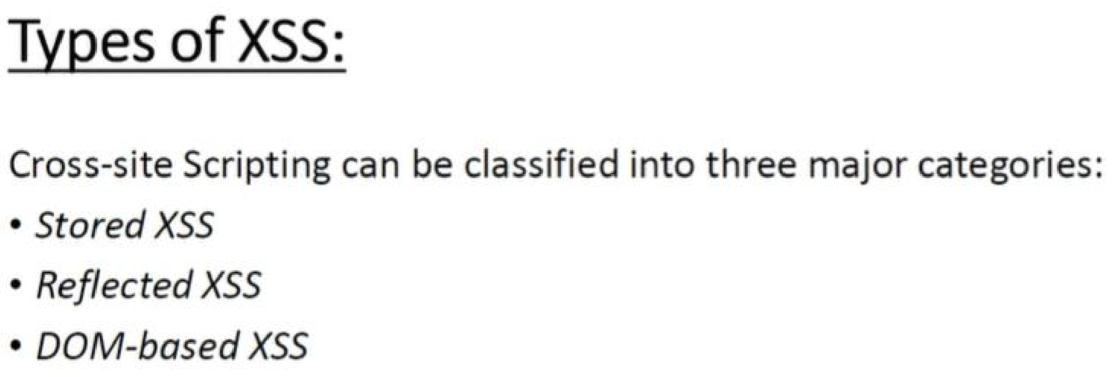
Some attacks can be sent to the application server and executed against many other users of that application, for example:

- A vulnerable forum site or comment system allows a user submitted post to be viewed by many other users that each become victims of the attack
- A vulnerable contact form sends a malicious message to site administrators that gives the attacker access to the "admin panel" when viewed by an administrative user



How Do I Prevent This From Happening?

- To prevent this vulnerability, developers must validate all input to the application and encode all input that is included in output.
- This is an essential part of application development and will help prevent many different types of vulnerabilities, not just XSS.

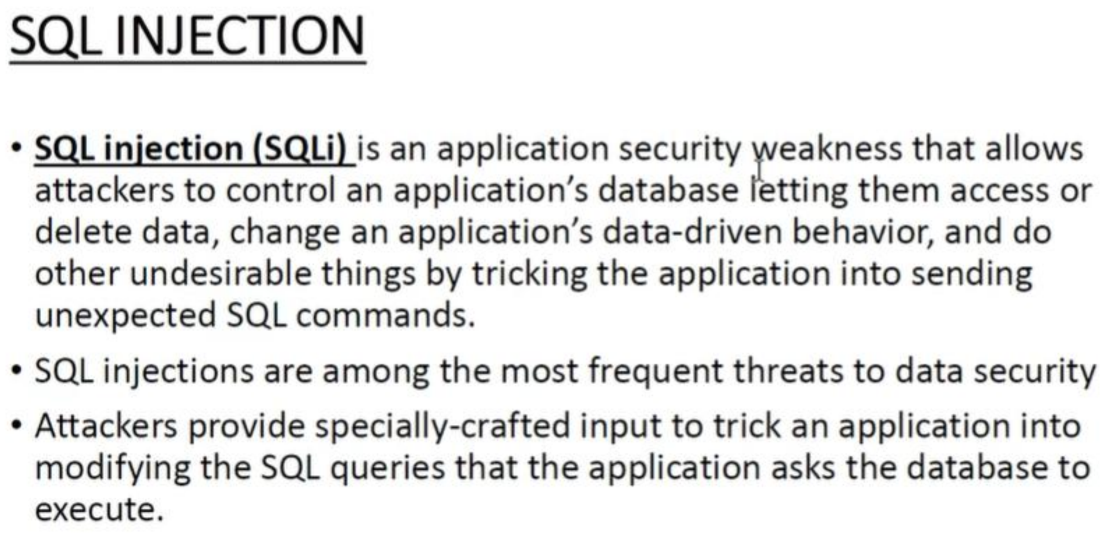


Types of XSS:

Cross-site Scripting can be classified into three major categories:

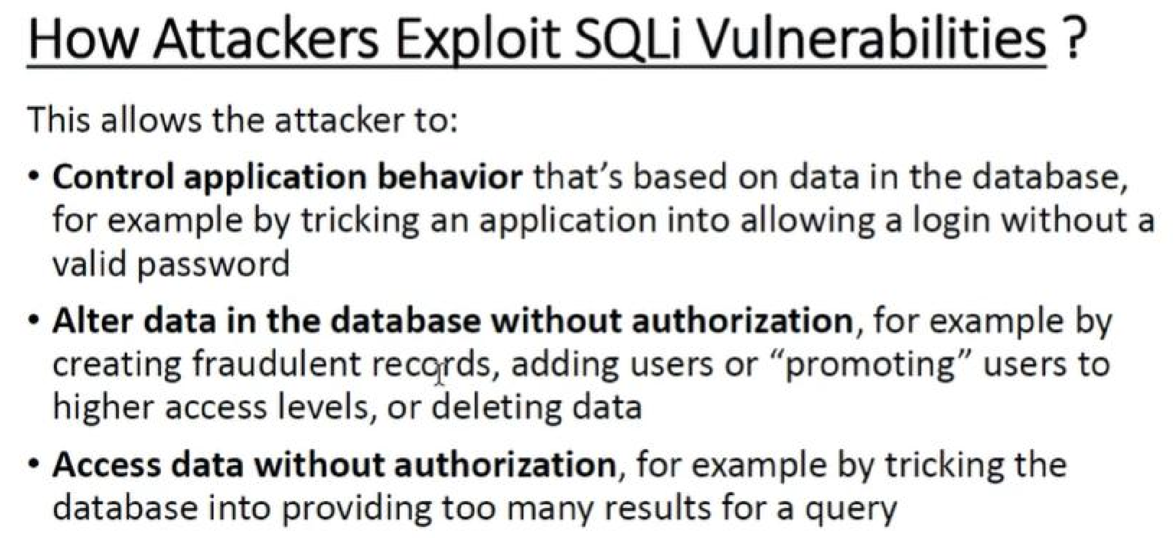
- *Stored XSS*
- *Reflected XSS*
- *DOM-based XSS*

Sql
Injection



SQL INJECTION

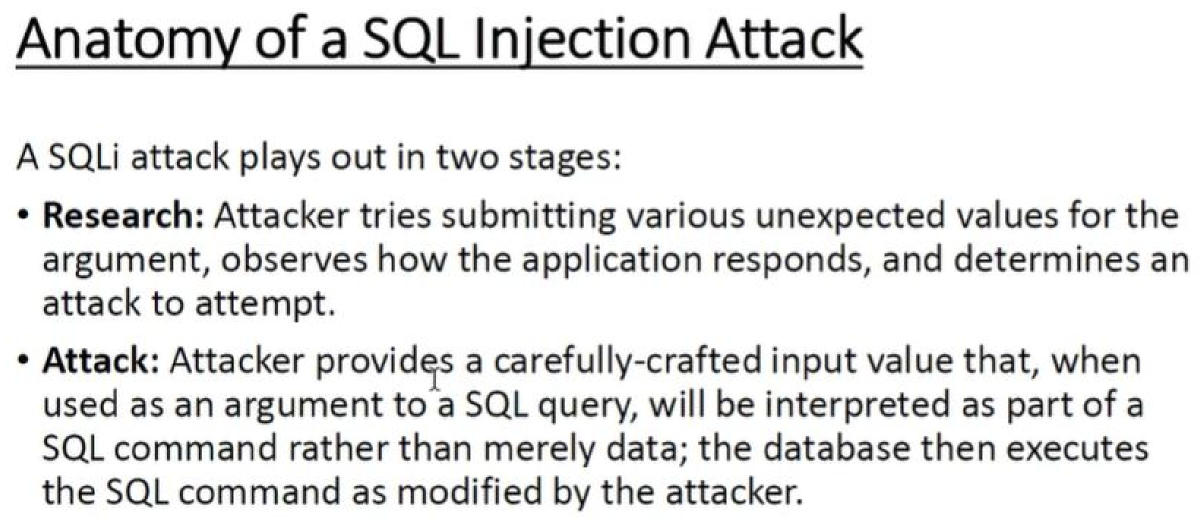
- **SQL injection (SQLi)** is an application security weakness that allows attackers to control an application's database letting them access or delete data, change an application's data-driven behavior, and do other undesirable things by tricking the application into sending unexpected SQL commands.
- SQL injections are among the most frequent threats to data security
- Attackers provide specially-crafted input to trick an application into modifying the SQL queries that the application asks the database to execute.



How Attackers Exploit SQLi Vulnerabilities ?

This allows the attacker to:

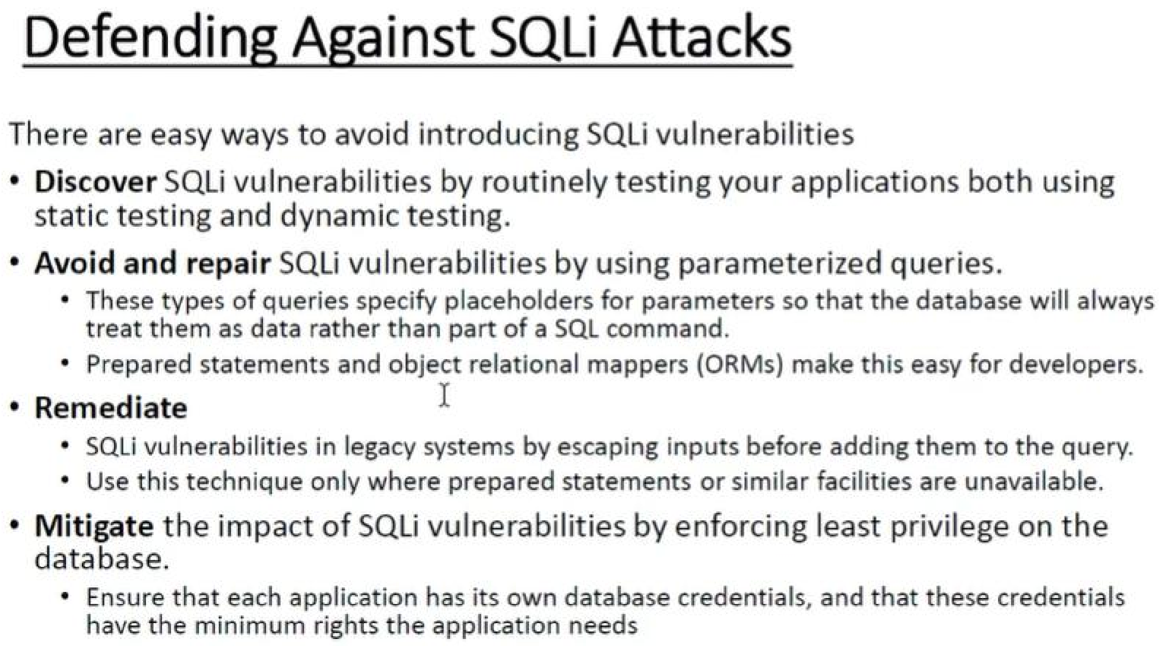
- **Control application behavior** that's based on data in the database, for example by tricking an application into allowing a login without a valid password
- **Alter data in the database without authorization**, for example by creating fraudulent records, adding users or "promoting" users to higher access levels, or deleting data
- **Access data without authorization**, for example by tricking the database into providing too many results for a query



Anatomy of a SQL Injection Attack

A SQLi attack plays out in two stages:

- **Research:** Attacker tries submitting various unexpected values for the argument, observes how the application responds, and determines an attack to attempt.
- **Attack:** Attacker provides a carefully-crafted input value that, when used as an argument to a SQL query, will be interpreted as part of a SQL command rather than merely data; the database then executes the SQL command as modified by the attacker.

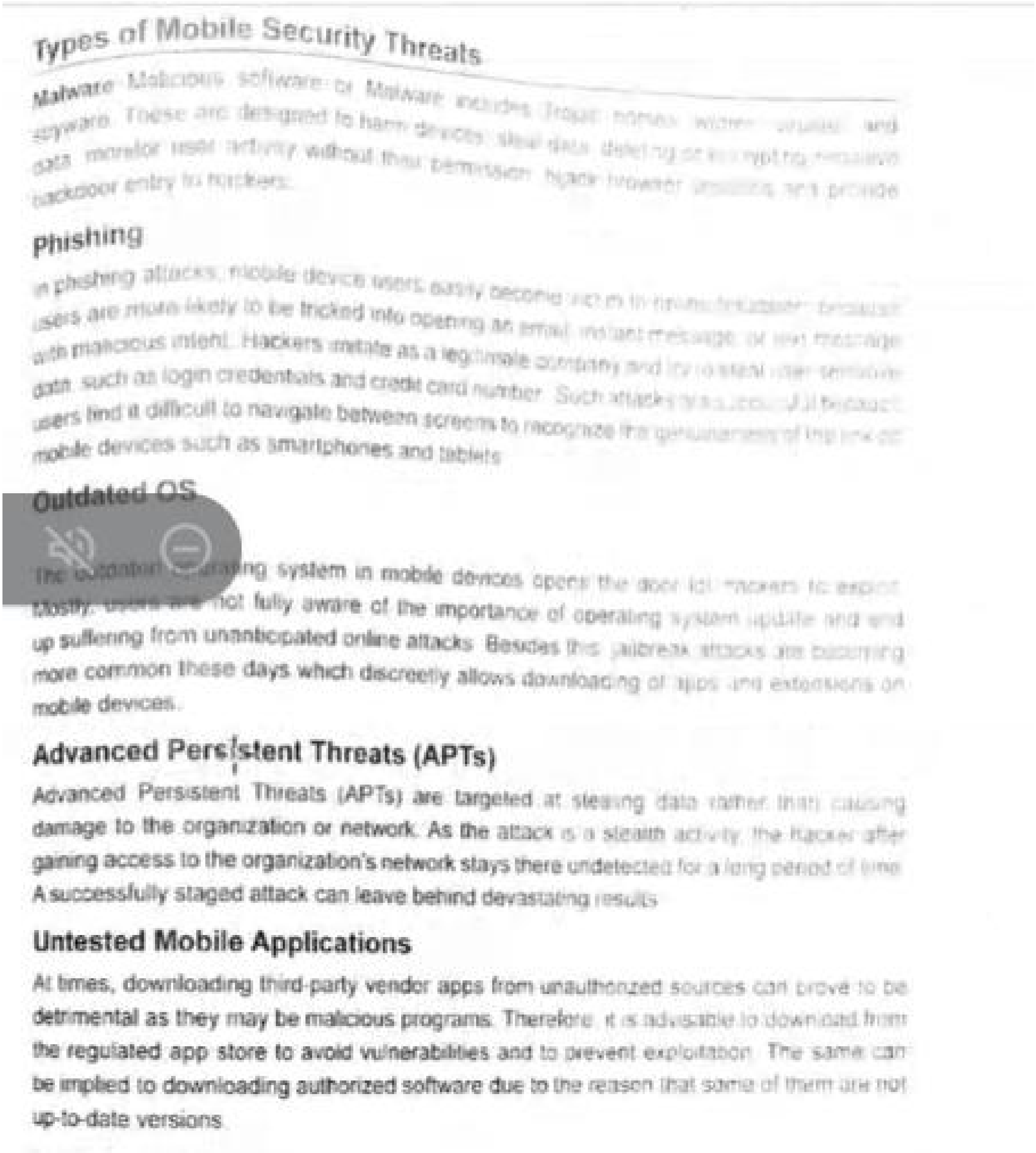


Defending Against SQLi Attacks

There are easy ways to avoid introducing SQLi vulnerabilities

- **Discover** SQLi vulnerabilities by routinely testing your applications both using static testing and dynamic testing.
- **Avoid and repair** SQLi vulnerabilities by using parameterized queries.
  - These types of queries specify placeholders for parameters so that the database will always treat them as data rather than part of a SQL command.
  - Prepared statements and object relational mappers (ORMs) make this easy for developers.
- **Remediate**
  - SQLi vulnerabilities in legacy systems by escaping inputs before adding them to the query.
  - Use this technique only where prepared statements or similar facilities are unavailable.
- **Mitigate** the impact of SQLi vulnerabilities by enforcing least privilege on the database.
  - Ensure that each application has its own database credentials, and that these credentials have the minimum rights the application needs

Types of mobile security threats



Types of Mobile Security Threats

Malware Malicious software or Malware includes Trojan horses, worms, viruses, and spyware. These are designed to harm devices, steal data, deleting or encrypting sensitive data, monitor user activity without their permission, hijack browser sessions and provide backdoor entry to hackers.

Phishing

In phishing attacks, mobile device users easily become victim to fraudsters because users are more likely to be tricked into opening an email, instant message, or text message with malicious intent. Hackers imitate as a legitimate company and try to steal user sensitive data, such as login credentials and credit card number. Such attacks are successful because users find it difficult to navigate between screens to recognize the genuineness of the link on mobile devices such as smartphones and tablets.

Outdated OS

The outdated operating system in mobile devices opens the door for hackers to exploit. Mostly, users are not fully aware of the importance of operating system update and end up suffering from unanticipated online attacks. Besides this, jailbreak attacks are becoming more common these days which discreetly allows downloading of apps and extensions on mobile devices.

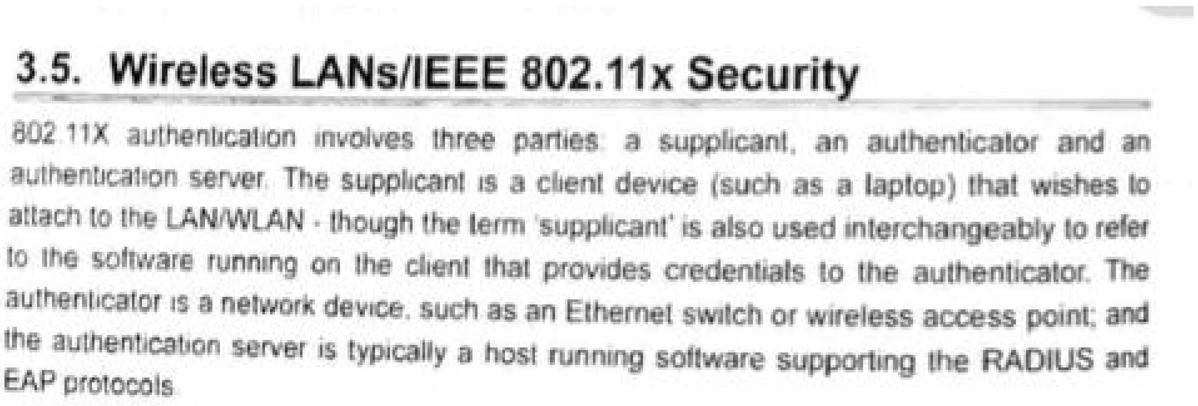
Advanced Persistent Threats (APTs)

Advanced Persistent Threats (APTs) are targeted at stealing data rather than causing damage to the organization or network. As the attack is a stealth activity, the hacker after gaining access to the organization's network stays there undetected for a long period of time. A successfully staged attack can leave behind devastating results.

Untested Mobile Applications

At times, downloading third-party vendor apps from unauthorized sources can prove to be detrimental as they may be malicious programs. Therefore, it is advisable to download from the regulated app store to avoid vulnerabilities and to prevent exploitation. The same can be implied to downloading authorized software due to the reason that some of them are not up-to-date versions.

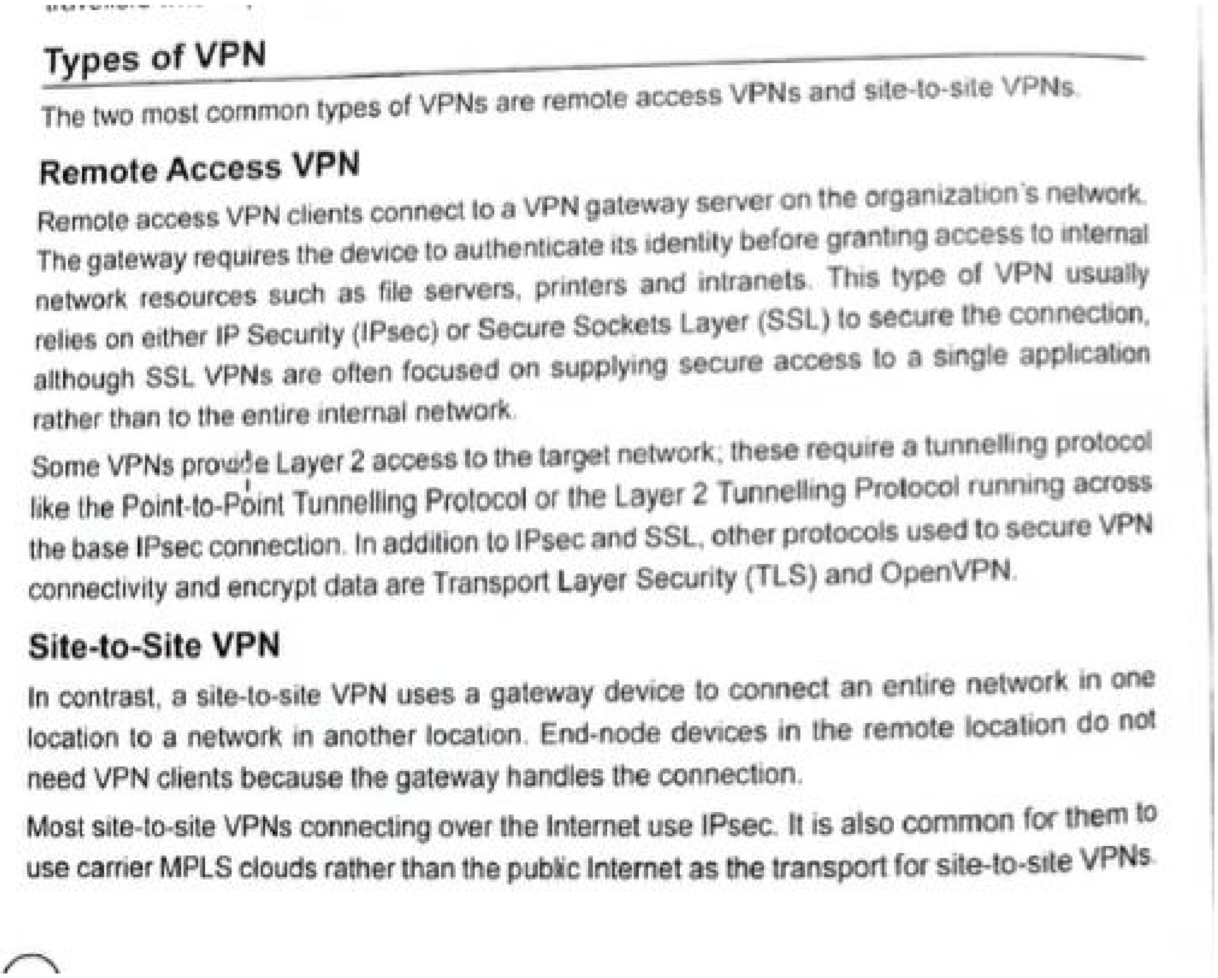
Wireless lans ieee 802.11 Security



3.5. Wireless LANs/IEEE 802.11x Security

802.11X authentication involves three parties: a supplicant, an authenticator and an authentication server. The supplicant is a client device (such as a laptop) that wishes to attach to the LAN/WLAN - though the term 'supplicant' is also used interchangeably to refer to the software running on the client that provides credentials to the authenticator. The authenticator is a network device, such as an Ethernet switch or wireless access point; and the authentication server is typically a host running software supporting the RADIUS and EAP protocols.

Types of vpn



Types of VPN

The two most common types of VPNs are remote access VPNs and site-to-site VPNs.

Remote Access VPN

Remote access VPN clients connect to a VPN gateway server on the organization's network. The gateway requires the device to authenticate its identity before granting access to internal network resources such as file servers, printers and intranets. This type of VPN usually relies on either IP Security (IPsec) or Secure Sockets Layer (SSL) to secure the connection, although SSL VPNs are often focused on supplying secure access to a single application rather than to the entire internal network.

Some VPNs provide Layer 2 access to the target network; these require a tunnelling protocol like the Point-to-Point Tunnelling Protocol or the Layer 2 Tunnelling Protocol running across the base IPsec connection. In addition to IPsec and SSL, other protocols used to secure VPN connectivity and encrypt data are Transport Layer Security (TLS) and OpenVPN.

Site-to-Site VPN

In contrast, a site-to-site VPN uses a gateway device to connect an entire network in one location to a network in another location. End-node devices in the remote location do not need VPN clients because the gateway handles the connection.

Most site-to-site VPNs connecting over the Internet use IPsec. It is also common for them to use carrier MPLS clouds rather than the public Internet as the transport for site-to-site VPNs.

Mobile VPN

In a mobile VPN, a VPN server still sits at the edge of the company network, enabling secure tunnelled access by authenticated, authorized VPN clients. Mobile VPN tunnels are not tied to physical IP addresses, however. Instead, each tunnel is bound to a logical IP address. That logical IP address sticks to the mobile device no matter where it may roam. An effective mobile VPN provides continuous service to users and can seamlessly switch across access technologies and multiple public and private networks.

Hardware VPN

Hardware VPNs offer a number of advantages over the software-based VPN. In addition to enhanced security, hardware VPNs can provide load balancing to handle large client loads. The administration is managed through a Web browser interface. A hardware VPN is more expensive than a software VPN. Because of the cost, hardware VPNs are a more realistic option for large businesses than for small businesses or branch offices. Several vendors, including Irish vendor InvizBox, offer devices that can function as hardware VPNs.

VPN Appliance

A VPN appliance, also known as a VPN gateway appliance, is a network device equipped with enhanced security features. Also known as an SSL (Secure Sockets Layer) VPN appliance, it is in effect a router that provides protection, authorization, authentication and encryption for VPNs.

VPN Security

VPN uses encryption to provide data confidentiality. Once connected, the VPN makes use of the tunneling mechanism described above to encapsulate encrypted data into a secure tunnel, with openly read headers that can cross a public network. Packets passed over a public network in this way are unreadable without proper decryption keys, thus ensuring that data is not disclosed or changed in any way during transmission.