INSTITUTE OF DISTANCE AND OPEN LEARNING

UNIVERSITY OF MUMBAI

M.Sc.

(Computer Science)

SEMESTER - II (CBCS)

ELECTIVE I - TRACK B : CYBER AND INFORMATION SECURITY (NETWORK AND COMMUNICATION SECURITY)

SUBJECT CODE : PSCS 2032

Programme Co-ordinator : **Shri. Mandar Bhanushe**
Head, Faculty of Science and Technology,
IDOL, University of Mumbai, Mumbai

Course Co-ordinator : **Mr. Sumedh Shejole**
Asst. Professor,
IDOL, University of Mumbai, Mumbai

Editor : **Rani Narayan Podichetty**
Assistant Professor,
K. B. College of Arts and Commerce for
women, kopri, Thane

Course Writers : **Nisha Ameya Vanjari**
: **Mrs. Trupti KulkarniKaujalgi**
: **Abhijeet Pawaskar**

October 2023, Print - I

Published by : Director
Institute of Distance and Open Learning ,
University of Mumbai,
Vidyanagari, Mumbai - 400 098.

DTP Composed and Printed by : Mumbai University Press
Vidyanagari, Santacruz (E), Mumbai - 400098

CONTENTS

❖❖❖❖

Syllabus

PSCS2032

Elective I - Track B: Cyber and Information Security (Network and Communication Security)

Unit I: Computer Security Principles of Security, Different Attacks: malicious and non-malicious program, Types of Computer Criminals. Operating System Security: Protected objects and methods of protection. Memory address protection: Fence, Relocation, Base/Bound Registers, Tagged Architecture, Segmentation, Paging, Directory, access control list. Database Security: Security requirements, Integrity, Confidentiality, Availability, Reliability of Database, Sensitive data, Multilevel database, Proposals for multilevel security. .

Unit II: Network Security Different types of network layer attacks, Firewall (ACL, Packet Filtering, DMZ, Alerts and Audit Trials) – IDS,IPS and its types (Signature based, Anomaly based, Policy based, Honeypot based). Web Server Security: SSL/TLS Basic Protocol-computing the keys-client authentication-PKI as deployed by SSL Attacks fixed in v3- ExportabilityEncoding-Secure Electronic Transaction (SET), Kerberos.

Unit III: Cloud Security How concepts of Security apply in the cloud, User authentication in the cloud; How the cloud provider can provide this- Virtualization System Security Issues: e.g. ESX and ESXi Security, ESX file system security- storage considerations, backup and recoveryVirtualization System Vulnerabilities, security management standards- SaaS, PaaS, IaaS availability management- access control- Data security and storage in cloud.

Unit IV: Mobile Security: Mobile system architectures, Overview of mobile cellular systems, GSM and UMTS Security & Attacks, Vulnerabilities in Cellular Services, Cellular Jamming Attacks & Mitigation, Security in Cellular VoIP Services, Mobile application security. Securing Wireless Networks: Overview of Wireless Networks, Scanning and Enumerating 802.11

Networks, Attacking 802.11 Networks, Bluetooth Scanning and Reconnaissance, Bluetooth Eavesdropping, Attacking & Exploiting Bluetooth, Zigbee Security & Attacks.

Text book: Security in Computing 4th edition, Charles P. Pfleeger, Charles P. Pfleeger, Shari Lawrence Pfleeger, Prentice Hall; 4th edition (2006) Mobile and Wireless Security and Privacy, Kia Makki, Peter Reiher, Springer, (2007). Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance (Theory and practice), Tim Mather, Subra Kumaraswamy, Shahed Latif., O'Reilly Media; 1 edition (2009).

Reference: Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Ronald L. Krutz, Russell Dean Vines, Wiley (2010) Network Security, Charlie Kaufman, Radia Perlam, Mike Speciner, Prentice Hall, 2nd Edition (2002) Cryptography and Network Security 3rd edition, Atul Kahate, Tata McGraw Hill Education Private Limited (2013) Network Security, Charlie Kaufman, Radia Perlam, Mike Speciner, Prentice Hall, 2nd Edition (2002) Cryptography and Network Security: Principles and practice 6th edition, William Stallings, Pearson Education (2013).

❖❖❖❖

1

COMPUTER SECURITY

Unit Structure :

1.0 OBJECTIVES

1. To understand the basics of security
2. To understand the attacks and types of computer criminals
3. To understand why security is needed.
4. To understand the necessity to maintain a safe network.

1.1 INTRODUCTION

Computer security basically is the protection of computer systems and information from harm, theft, and unauthorized use. It is the process of preventing and detecting unauthorized use of your computer system. There are many types of computer security which is used to protect the valuableorganizational information. Security is concerned with the protection of systems, networks, applications, and information. In some cases, it is also called electronic information security or information technology security. Every organisation willingly perform audits of security to check the loopwholes of the computer systems and used to restrict other outside devices usage in organisation and related ports or services are restricted by their security experts to protect the information of organisation.

There are different types of computer security such as application security, network security, internet security, data security, information security and end user security as shown in figure.

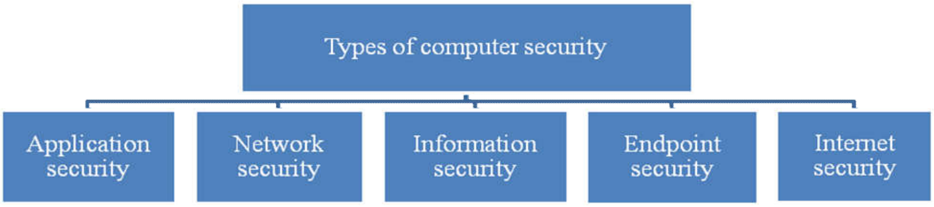
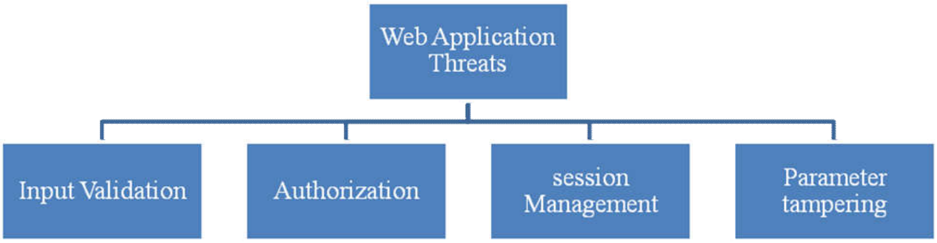


Fig 1.1.1: Types of security

1. Application Security

Application security is the adding security features within applications to prevent from attacks. The attacks can be SQL injection, DoS attacks.Firewall, antivirus, etc are security tools which can help to prevent from attacks.



Web Application Threats

Fig 1.1.2: Web Application Threats

Here are the most common categories of **application threat**s related to software or application, which are given bellows:

A. Input validation

Input validation or data validation is the process of correct testing of any input that is provide by users. It is difficult to detect a malicious user who is trying to attack the software and applications. So, it should check and validate all input data which will enter into a system.

Following figures shows some of vulnerabilities that could be solved just by validating input.

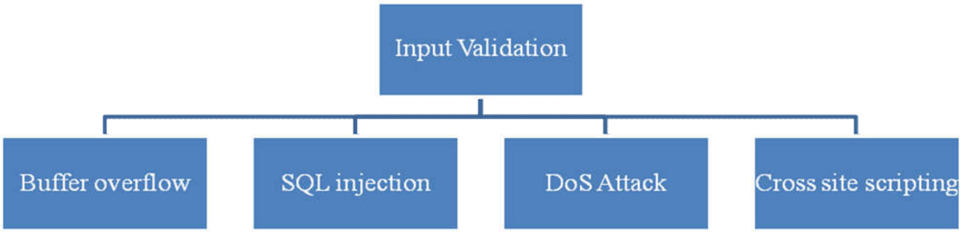


Fig 1.1.3: Input Validation

B. Authorization

It is nothing but user priviladgemechanism such as computer programs, files, services, data, etc.

C. Session management

Session management is a process used by the web container to securing multiple requests to a service from the same user or entity. In short track the frequency of visits to an application and movement within the site.

There are two types of session management: one is **cookie-based and another one is URL rewriting.**

D. Parameter tampering

Parameter tampering is a technique which malicious hackers attempt to compromise an application through manipulating parameters in the URL string.

It is a simple attack targeting the application business logic in order to modify application data, such as user credentials and permissions, price and quantity of products.

For example, a shopping site uses hidden fields to refer to its items, as follows:

<input type=”hidden” id=”1234” name=”payment” **value**=”240.00”>

Here, an attacker can modify or alter the “**value**” information of a specific item, thus lowering its cost.

2. Information Security

Information security (IS) is a **types of computer security which** refers to the process and methodology to protect the **confidentiality**, **integrity** and **availability** of computer system from unauthorized access, use, modification and destruction.

Information security focuses on the **CIA triad** model, which ensure **confidentiality**, **integrity**, and **availability** of data, without affecting organization productivity.

3. Network Security

Network security is other **types of computer security** which process of preventing and protecting against unauthorized intrusion into computer networks. It is a set of rules and configurations which designed to protect the confidentiality, integrity and accessibility of computer networks system and information using both software and hardware technologies.

Network Security Methods

There are different components or methods to improve network security. Here, we have mentioned the most common network security components.

- Antivirus Software
- Application Security

- Email Security
- Firewalls
- Web Security
- Wireless Security
- Network Access Control (NAC)

4. Endpoint Security

Human error is a major weak point which is easily exploited by cyber criminals. End users are becoming the largest security risk in any organizations. However, end user has no fault of their own, and mostly due to a lack of awareness and ICT policy. They can unintentional open the virtual gates to cyber attackers.

That's why comprehensive security policies, procedures and protocols have to be understood in depth by users who accessing the sensitive information. It is better to provide security awareness training program to them and should cover the following topics:

- Cyber security and its importance
- Phishing and Social Engineering attack
- Password creation and usages
- Device Security
- Physical Security

5. Internet Security

Internet security is the important **types of computer security** which has defined as a process to create set of rules and actions to protect computers system that are connected to the Internet. It is a branch of computer security that deals specifically with internet-based threats such as:

A. Hacking

A Hacker is a person who finds weakness and exploits the vulnerability in computer systems or network to gain access. Hacking refers to activities that exploit a computer system or a network in order to gain unauthorized access or control over systems for illegal purpose.

B. Computer Viruses

A computer virus is a software program that can spread from one computer system to another computer without the user's knowledge and performs malicious actions. It has capability to corrupt or damage data, destroy files, format hard drives or make disks unreadable.

There are different types of **computer viruses** which are as follows:

- Boot Sector Virus
- Direct Action Virus
- Resident Virus
- Macro Virus
- Multipartite Virus
- File Infector Virus
- Browser Hijacker
- Polymorphic Virus
- Web Scripting Virus

A computer virus may spread on your computer and other devices as the following ways:

- Downloads Software Or Files
- E-Mail Attachments
- Phishing Emails
- External Devices
- Online Advertisements
- Click On Malicious File
- Infected Website
- Copying Data From Infected Computer
- Unsolicited E-Mail
- Social Media Scam Links

C. Denial-of-Service Attacks

It is an attack that shut down a system and making it inaccessible to the users. It occurs when an attacker prevents legitimate users from accessing specific computer systems, devices or other resources and flooding a target system.

D. Malware

Malware is short for "**malicious software**" that typically consists of software program or code. It is developed by cyber attackers which are designed to extensive damage to data and systems.

There are different types of malware such as

- Computerviruses
- Spyware
- Ransomware
- Worms
- Trojan horses

Any type of malicious code.

1.2 WHAT IS SECURITY [1]

Security for information technology (IT) refers to the methods, tools and personnel used to defend an organization's digital assets. The goal of IT security is to protect these assets, devices and services from being interrupted, whipped or broken by unauthorized users. These threats can be external or internal and malicious or accidental.

An effective security strategy uses a range of approaches to minimize vulnerabilities and target many types of cyberthreats. Detection, prevention and response to security threats involve the use of security policies, software tools and IT services.

1.3 PRINCIPLE OF SECURITY [2]

Below figure shows the security principles.

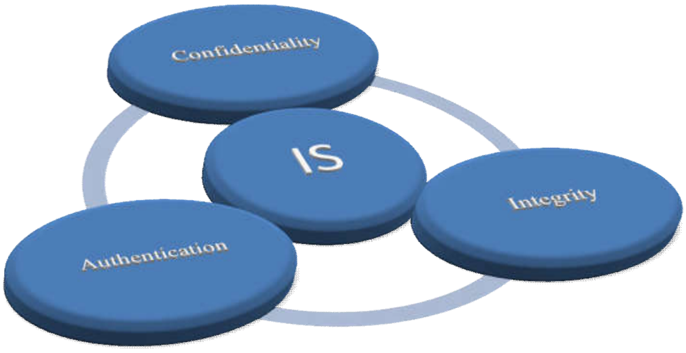


Fig 1.3.1: Principles of security

1. Confidentiality

The confidentiality principle of security states that only their intended sender and receiver should be able to access messages, if an unauthorized person gets access to this message then the confidentiality gets compromised. For example, suppose user X wants to send a message to user Y, and X does not want some else to get access to this message, or if it gets access, he/she does not come to know about the details. But if user Z somehow gets access to this secret message, which is not desired, then the purpose of this confidentiality gets fail. This leads to the interception. i.e. if user Z access the secret message or email sent by user X to Y without permission of X and Y, then it is called an interception. Interception causes loss of message confidentiality.

2. Authentication

The authentication principle of security establishes proof of identity, it ensures that the origin of a document or electronic message is correctly identified. For example suppose user Z sends a message to user Y, however, the trouble is that user Z posed as user X while sending a message to user Y. How user Y would knows that message comes from Z, not X. This leads to the fabrication attack. For example

The attacker can act as user X and sends fund transfer request(from X' account to attacker account) to a bank, and the bank will transfer the amount as requested from X's account to attacker, as banks think fund transfer request comes from user X. Fabrication is possible in absence of proper authentication mechanism.

3. Integrity

The integrity principle of security states that the message should not be altered. In other words, we can say that, when the content of the message changes after the sender sends it, but before it reaches the intended receiver, we can say that integrity of the message is lost. For example, suppose user X sends a message to User Y, and attacker Z somehow gets access to this message during transmission and changes the content of the message and then sends it to user Y. User Y and User X does not have any knowledge that the content of the message was changed after user X send it to Y. This leads to a modification. Modification causes loss of message integrity.

4. Non-repudiation

Non-repudiation principle of security does not allow the sender of a message to refute the claim of not sending that message. There are some situations where the user sends a message and later on refuses that he/she had sent that message. For example, user X sends requests to the bank for fund transfer over the internet. After the bank performs fund transfer based on user X request, User X cannot claim that he/she never sent the fund transfer request to the bank. This principle of security defeats such possibilities of denying something after having done it.

5. Access control

Access control principles of security determine who should be able to access what. i.e. we can specify that what users can access which functions, for example, we can specify that user X can view the database record but cannot update them, but user Y can access both, can view record, and can update them. This principle is broadly related to two areas – role management and rule management where role management concentrates on the user side. i.e. which user can do what and rule management concentrate on the resources side i.e. which resource is available. Based on this matrix is prepared, which lists the user against q list of items they can access. The access control list is a subset of the access control matrix.

6. Availability

The availability principle of security states that resources should be available to the authorized person at all times. For example, because of the intentional action of another unauthorized user Z, an authorized user x may not be able to contact server Y, this leads to an interruption attack, interruption puts the availability of resources in danger. A real-life example of this could be, suppose attacker or unauthorized person Z tries to access the FB Account of user X, as User Z does not know the password of user X, he/she tries to log in to the X's account using a random password. after attempting maxim limit for the password, if it is not correct then X's account will be blocked, therefore because of unauthorized person Z, user X could not access his account.

7. Ethical and legal issues

Ethical issues in the security system are classified into the following categories

- **Privacy:** It deals with the individual's right to access the personal information
- **Accuracy:** It deals with the responsibility of authentication, fidelity, and accuracy of information
- **Property:** It deals with the owner of the information
- **Accessibility:** It deals with what information does an organization has the right to collect.

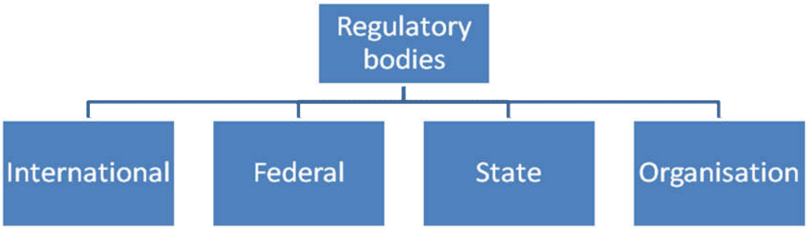


Fig 1.3.2: Types of Regulatory bodies

While dealing with legal issues, we must remember that there is a hierarchy of regulatory bodies that govern the legality of information security, it can be classified into the following categories

- International
- Federal
- State
- Organization

1.4 ATTACKS [3]

A cyber attack is a set of actions performed by threat actors, who try to gain unauthorized access, steal data or cause damage to computers, computer networks, or other computing systems. A cyber attack can be launched from any location. The attack can be performed by an individual or a group using one or more tactics, techniques and procedures (TTPs).

The individuals who launch cyber attacks are usually referred to as cybercriminals, threat actors, bad actors, or hackers. They can work alone, in collaboration with other attackers, or as part of an organized criminal group. They try to identify vulnerabilities—problems or weaknesses in computer systems—and exploit them to further their goals.

Categories of attack

There are mainly two categories where attack can be classify like active attack and passive attack.

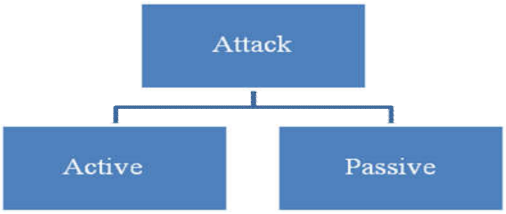


Fig 1.4.1: Categories of Attacks

Active attack means some modification or some harmful changes done in system.

passive attack means monitor the activities to create strategy for attacking.

Following table will give glance about these categories

Sr. No.	Active Attack	Passive Attack
1	Access information and then do the modification	Only access information
2	System will get impacted	No harm for system
3	Easier to detect attack	Difficult to detect attack
4	Threat to integrity and availability	Threat to confidentiality
5	Example – Repudiation, DOS	Example – Network traffic analysis

Cyber Attack Statistics

The global cost of cyber attacks is expected to grow by 15% per year and is expected to reach over $10 trillion. A growing part of this cost is Ransomware attacks, which now cost businesses in the US $20 billion per year.The average cost of a data breach in the US is $3.8 million. Another alarming statistic is that public companies lose an average of 8% of their stock value after a successful breach.

In a recent survey, 78% of respondents said they believe their company's cybersecurity measures need to be improved. As many as 43% of small businesses do not have any cyber defenses in place. At the same time, organizations of all sizes are facing a global cybersecurity skills shortage, with almost 3.5 million open jobs worldwide, 500,000 of them in the US alone.

Cyber Attack Examples

Here are a few recent examples of cyber attacks that had a global impact.

Kaseya Ransomware Attack

Kaseya, a US-based provider of remote management software, experienced a supply chain attack, which was made public on July 2, 2021. The company announced that attackers could use its VSA product to infect customer machines with ransomware.

SolarWinds Supply Chain Attack

This was a massive, highly innovative supply chain attack detected in December 2020, and named after its victim, Austin-based IT management company SolarWinds. It was conducted by APT 29, an organized cybercrime group connected to the Russian government.

Amazon DDoS Attack

In February 2020, Amazon Web Services (AWS) was the target of a large-scale distributed denial of service (DDoS) attack. The company experienced and mitigated a 2.3 Tbps (terabits per second) DDoS attack, which had a packet forwarding rate of 293.1 Mpps and a request rate per second (rps) of 694,201. It is considered one of the largest DDoS attacks in history.

Microsoft Exchange Remote Code Execution Attack

In March 2021, a large-scale cyber attack was carried out against Microsoft Exchange, a popular enterprise email server. It leveraged four separate zero-day vulnerabilities discovered in Microsoft Exchange servers.

Twitter Celebrities Attack

In July 2020, Twitter was breached by a group of three attackers, who took over popular Twitter accounts. They used social engineering attacks to steal employee credentials and gain access to the company's internal management systems, later identified by Twitter as vishing (phone phishing).

Other Notable Attacks

2018

- Marriott’s Starwood Hotels announced a breach that leaked the personal data of more than 500 million guests.
- UnderArmor’s MyFitnessPal brand leaked the email addresses and login information of 150 million user accounts.

2017

- The WannaCry ransomware attack affected more than 300,000 computers in 150 countries, causing billions of dollars in damages.
- Equifax experienced open source vulnerability in an unpatched software component, which leaked the personal information of 145 million people.

2016

- The NotPetya attack hit targets around the world, with several waves continuing for more than a year, costing more than $10 billion in damage.
- An attack on the FriendFinder adult dating website compromised the data of 412 million users.
- Yahoo’s data breach incident compromised the accounts of 1 billion users, not long after a previous attack exposed personal information contained in 500 million user accounts.

Types of Cyber Attacks

While there are thousands of known variants of cyber attacks, here are a few of the most common attacks experienced by organizations every day.

Ransomware

Ransomware is malware that uses encryption to deny access to resources (such as the user’s files), usually in an attempt to compel the victim to pay a ransom. Once a system has been infected, files are irreversibly encrypted, and the victim must either pay the ransom to unlock the encrypted resources, or use backups to restore them.

Ransomware is one of the most prevalent types of attacks, with some attacks using extortion techniques, such as threatening to expose sensitive data if the target fails to pay the ransom. In many cases, paying the ransom is ineffective and does not restore the user’s data.

Malware

There are many types of malware, of which ransomware is just one variant. Malware can be used for a range of objectives from stealing information, to defacing or altering web content, to damaging a computing system permanently.

The malware landscape evolves very quickly, but the most prevalent forms of malware are:

- **Botnet Malware**—adds infected systems to a botnet, allowing attackers to use them for criminal activity
- **Cryptominers**—mines cryptocurrency using the target's computer
- **Infostealers**—collects sensitive information on the target's computer
- **Banking trojans**—steals financial and credential information for banking websites
- **Mobile Malware**—targets devices via apps or SMS
- **Rootkits**—gives the attacker complete control over a device's operating system

DoS and DDoS Attacks

Denial-of-service (DoS) attacks overwhelm the target system so it cannot respond to legitimate requests. Distributed denial-of-service (DDoS) attacks are similar but involve multiple host machines. The target site is flooded with illegitimate service requests and is forced to deny service to legitimate users. This is because servers consume all available resources to respond to the request overload.

These attacks don't provide the attacker with access to the target system or any direct benefit. They are used purely for the purpose of sabotage, or as a diversion used to distract security teams while attackers carry out other attacks.

Firewalls and network security solutions can help protect against small-scale DoS attacks. To protect against large scale DDoS, organizations leverage cloud-based DDoS protection which can scale on demand to respond to a huge number of malicious requests.

Phishing and Social Engineering Attacks

Social engineering is an attack vector that relies heavily on human interaction, used in over 90% of cyberattacks. It involves impersonating a trusted person or entity, and tricking individuals into granting an attacker sensitive information, transferring funds, or providing access to systems or networks.

Phishing attacks occur when a malicious attacker obtains sensitive information from a target and sends a message that appears to be from a trusted and legitimate source. The name "phishing" alludes to the fact that attackers are "fishing" for access or sensitive information, baiting the unsuspecting user with an emotional hook and a trusted identity.

As part of a phishing message, attackers typically send links to malicious websites, prompt the user to download malicious software, or request sensitive information directly through email, text messaging systems or social media platforms. A variation on phishing is "spear phishing", where attackers send carefully crafted messages to individuals with special privileges, such as network administrators, executives, or employees in financial roles.

MitM Attacks

Man-in-the-Middle (MitM) attacks are breaches that allow attackers to intercept the data transmitted between networks, computers or users. The attacker is positioned in the "middle" of the two parties and can spy on their communication, often without being detected. The attacker can also modify messages before sending them on to the intended recipient.

You can use VPNs or apply strong encryption to access points to protect yourself from MitM attacks.

Fileless Attacks

Fileless attacks are a new type of malware attack, which takes advantage of applications already installed on a user's device. Unlike traditional malware, which needs to deploy itself on a target machine, fileless attacks use already installed applications that are considered safe, and so are undetectable by legacy antivirus tools.

Fileless malware attacks can be triggered by user-initiated actions, or may be triggered with no user action, by exploiting operating system vulnerabilities. Fileless malware resides in the device's RAM and typically access native operating system tools, like PowerShell and Windows Management Instrumentation (WMI) to inject malicious code.

A trusted application on a privileged system can carry out system operations on multiple endpoints, making them ideal targets for fileless malware attacks.

1.4.1 Malicious program

Malicious programs can be divided into the following groups: worms, viruses, trojans, hacker utilities and other malware. All of these are designed to damage the infected machine or other networked machines. This category includes programs that propagate via LANs or the Internet with the following objectives: Penetrating remote machines.

1.4.2 Non-malicious program

Nonmalicious Program Errors Being human, programmers and other developers make many mistakes, most of which are unintentional and **nonmalicious**. Many such errors cause program malfunctions but do not lead to more serious security vulnerabilities.

1.5 TYPES OF COMPUTER CRIMINALS[4]

1) Script kiddies: A wannabe hacker. Someone who wants to be a hacker (or thinks they are) but lacks any serious technical expertise. They are usually only able to attack very weakly secured systems.

2) Scammers: Your email inbox is probably full of their work. Discount pharmaceuticals, time-shares, personal ads from available women in Russia…sound familiar?

3) Hacker groups: Usually work anonymously and create tools for hacking. They often hack computers for no criminal reason and are sometimes even hired by companies wanting to test their security.

4) Phishers: Gotten an email recently claiming your bank account is about to expire? Don't fall for these jerks. They want your personal information and, most likely, your identity, by directing you to a phony websites.

5) Political/religious/commercial groups: Tend to not be interested in financial gain. These guys develop malware for political ends. If you think this group is harmless, think Stuxnet. The Stuxnet worm which attacked Iran's Atomic Program of Its Nuclear Facilities was believed to be created by a foreign government.

6) Insiders: They may only be 20% of the threat, but they produce 80% of the damage. These attackers are considered to be the highest risk. To make matters worse, as the name suggests, they often reside within an organization.

1.6 SUMMARY

Information Security is not only about securing information from unauthorized access. Information Security is basically the practice of preventing unauthorized access, use, disclosure, disruption, modification, inspection, recording or destruction of information. Information can be physical or electronic one.

1.7 REFERENCE FOR FURTHER READING

1. https://www.techtarget.com/searchsecurity/definition/securityecurity? (techtarget.com)
2. https://www.techtarget.com/searchsecurity/definition/securityles of Security (educba.com)
3. https://www.faronics.com/news/blog/7-types-of-cyber-criminalscom)
4. https://www.faronics.com/news/blog/7-types-of-cyber-criminals
5. https://cyberthreatportal.com/types-of-computer-security/ rthreatportal.com)

❖❖❖❖

2

OPERATING SYSTEM SECURITY

Unit Structure :

2.0 OBJECTIVES

1. To understand the concepts of memory address protection

2. To understand access control list

3. To understand the importance of Memory Management

4. To understand the how segmentation can help in proper Memory Management.

2.1 INTRODUCTION

Operating system security (OS security) is the process of ensuring OS integrity, confidentiality and availability.OS security refers to specified steps or measures used to protect the OS from threats, viruses, worms, malware or remote hacker intrusions. OS security encompasses all preventive-control techniques, which safeguard any computer assets capable of being stolen, edited or deleted if OS security is compromised.

2.2 PROTECTED OBJECTS AND METHODS OF PROTECTION

To make your operating system secure, we need to protect operating system objects like memory, sharable I/O devices, such as disk, serially reusable I/O devices, such as printers and tape drives, sharable programs and sub-procedures, networks and sharable data.

2.3 MEMORY 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.

2.3.1 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 was a predefined memory address, enabling the operating system to reside on one side and the user to stay on the other. Unfortunately, this kind of implementation was very restrictive because a predefined amount of space was always reserved for the operating system, whether it was needed or not. If less than the predefined space was required, the excess space was wasted. Conversely, if the operating system needed more space, it could not grow beyond the fence boundary.

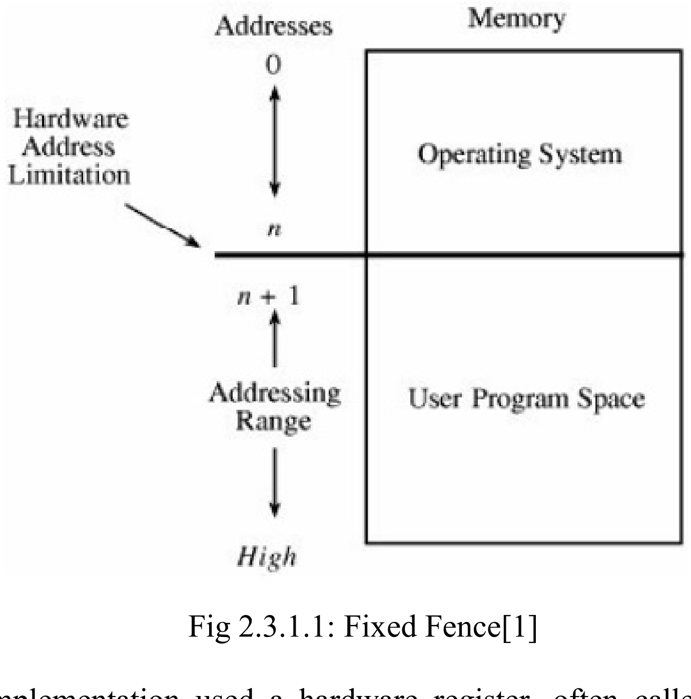
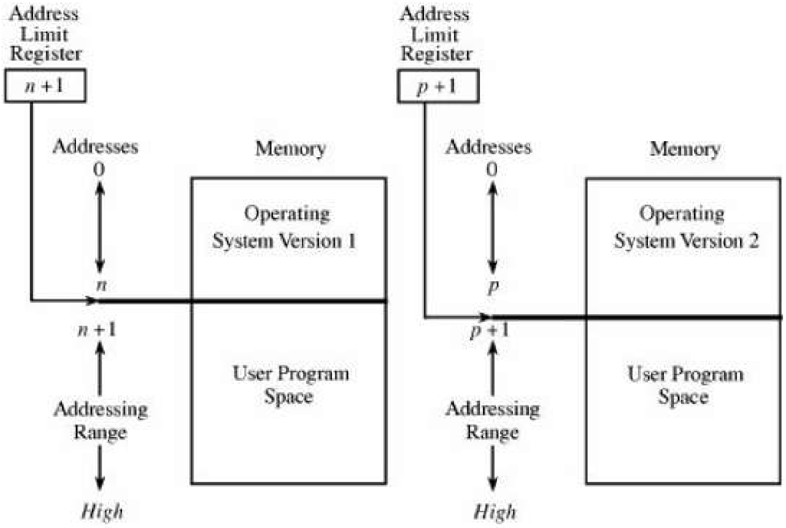


Fig 2.3.1.1: Fixed Fence[1]

Another implementation used a hardware register, often called a **fence register**, containing the address of the end of the operating system. In contrast to a fixed fence, in this scheme the location of the fence could be changed. Each time a user program generated an address for data modification, the address was automatically compared with the fence address. If the address was greater than the fence address (that is, in the user area), the instruction was executed; if it was less than the fence address (that is, in the operating system area), an error condition was raised.

Figure 4-2. Variable Fence Register.



2.3.1.2: Variable Fence Register[1]

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).

2.3.2 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.

2.3.3 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 multiuser 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. 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.

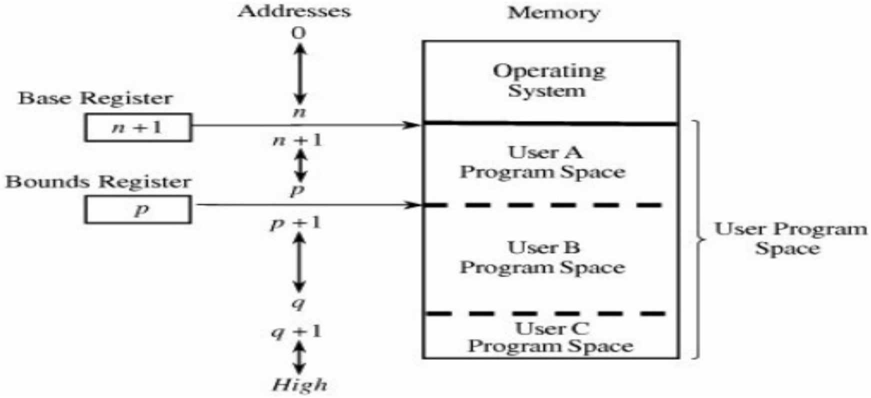


Fig 2.3.3.1 : Pair of Base/Bounds Registers

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 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 the 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 the 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.

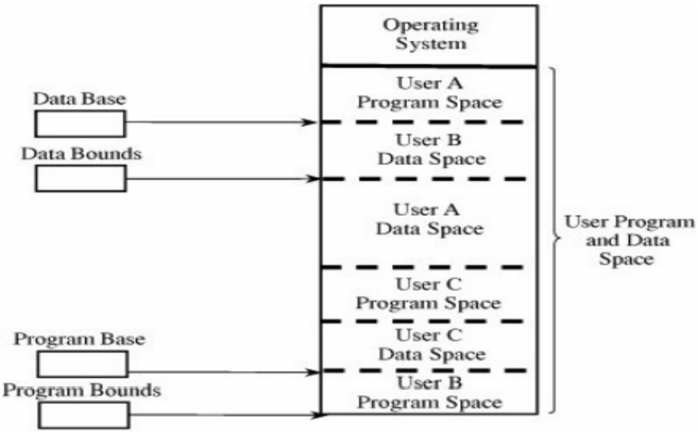


Fig 2.3.3.2: Two pairs of Base/Bounds Registers

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), something in the machine code of each instruction 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.

2.3.4 Tagged Architecture

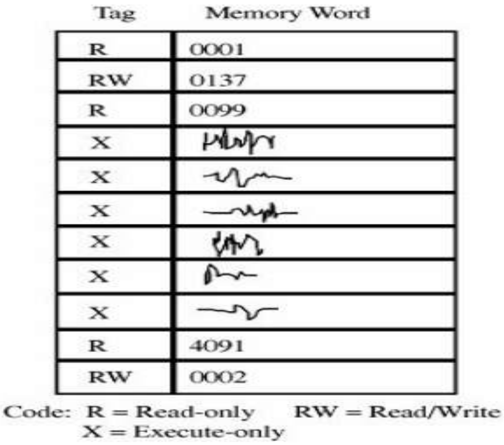
Another problem with using base/bounds registers for protection or relocation is their contiguous nature. Each pair of registers confines accesses to a consecutive range of addresses. A compiler or loader can easily rearrange a program so that all code sections are adjacent and all data sections are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

However, in some cases you may want to protect some data values but not all. For example, a personnel 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 the other programs being run. These characteristics 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.

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 were 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.

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.



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

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

Fig 2.3.4: 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 2^{256} different protection domains.

Compatibility of code presented a problem with the acceptance of a tagged architecture. A tagged architecture may not be as useful as more 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 the 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.

2.3.5 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. Although these approaches are ancient by computing's standardsthey were designed between 1965 and 1975they 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 shownin fig.

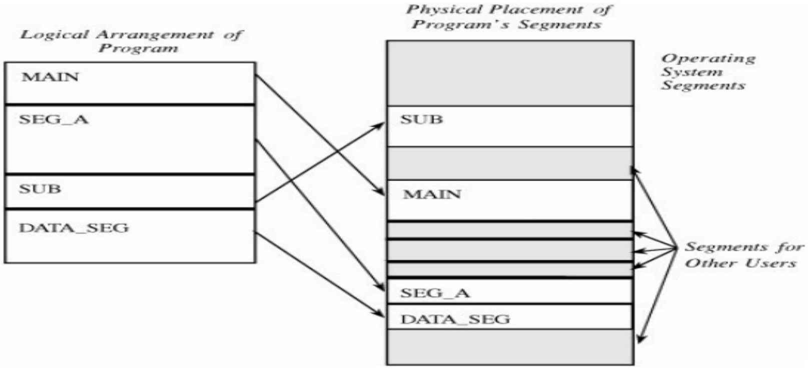

Logical Arrangement of Program
MAIN
SEG_A
SUB
DATA_SEG
Physical Placement of Program's Segments
Operating System Segments
SUB
MAIN
Segments for Other Users
SEG_A
DATA_SEG

Fig 2.3.5.1: 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. 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.

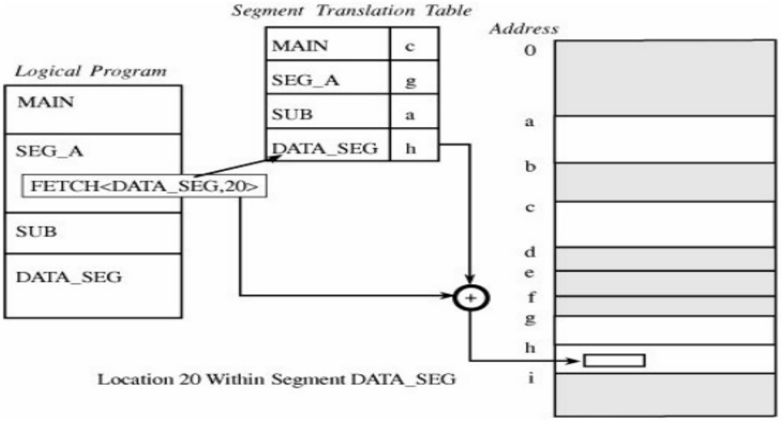


Fig 2.3.5.2: Translation of Segment Address

Thus, a user's program does not know what true memory addresses it uses. It has no wayand no needto 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. Because it translates all address references by a segment address table, the operating system needs only update the address in that one table when a segment is moved.
- 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.
- Many 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 concerns 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 must be the same.

2.3.6 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>.

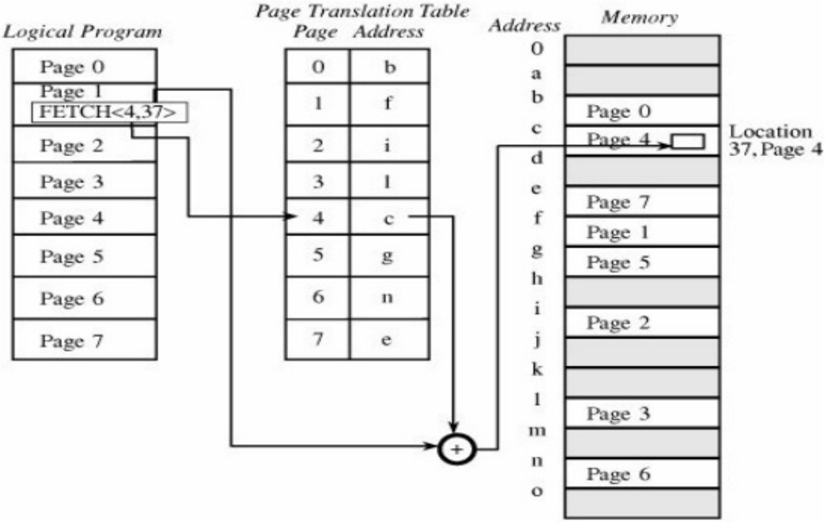


Fig 2.3.6.1: Page Address Translation

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 = 2^{10}), 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.

Combined Paging 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.

Figure 4-9. Paged Segmentation.

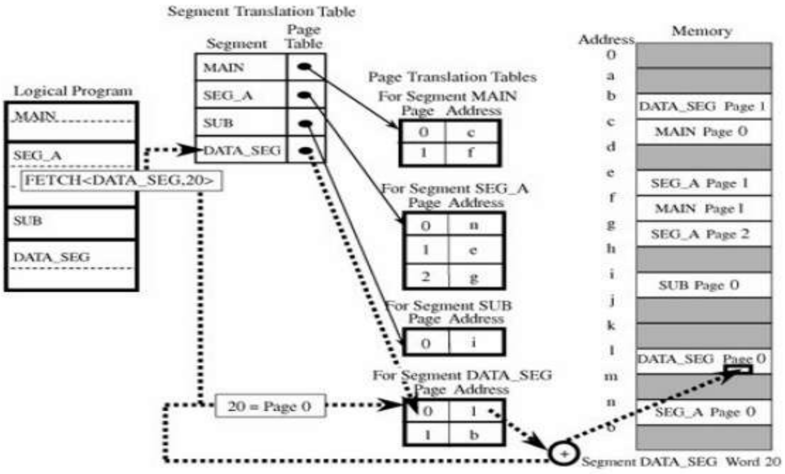


Fig 2.3.6.2: Page Segmentation

2.3.7 Directory

One simple way to protect an object is to use a mechanism that works like a file directory. Imagine we are trying to protect files (the set of objects) from users of a computing system (the set of subjects). Every file has a unique owner who possesses "control" access rights (including the rights to declare who has what access) and to revoke access to any person at any time. Each user has a file directory, which lists all the files to which that user has access.

Clearly, no user can be allowed to write in the file directory because that would be a way to forge access to a file. Therefore, the operating system must maintain all file directories, under commands from the owners of files. The obvious rights to files are the common read, write, and execute familiar on many shared systems. Furthermore, another right, owner, is possessed by the owner, permitting that user to grant and revoke access rights. Figure shows an example of a file directory.

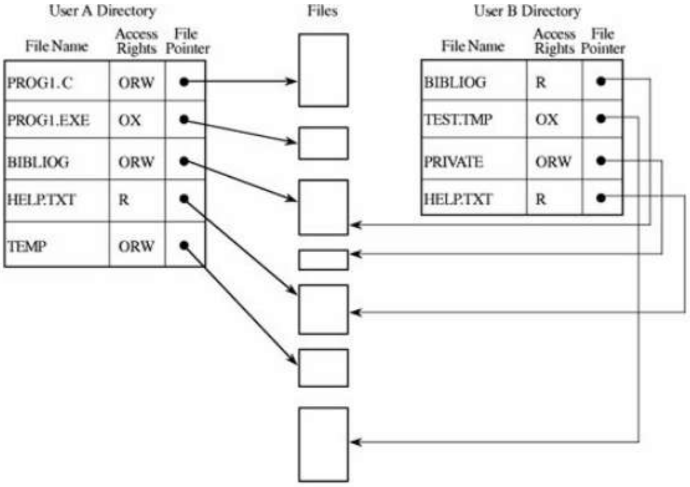


Fig 2.3.7: Directory Access

This approach is easy to implement because it uses one list per user, naming all the objects that user is allowed to access. However, several difficulties can arise. First, the list becomes too large if many shared objects, such as libraries of subprograms or a common table of users, are accessible to all users. The directory of each user must have one entry for each such shared object, even if the user has no intention of accessing the object. Deletion must be reflected in all directories.

A second difficulty is **revocation of access**. If owner A has passed to user B the right to read file F, an entry for F is made in the directory for B. This granting of access implies a level of trust between A and B. If A later questions that trust, A may want to revoke the access right of B. The operating system can respond easily to the single request to delete the right of B to access F because that action involves deleting one entry from a specific directory. But if A wants to remove the rights of everyone to access F, the operating system must search each individual directory for the entry F, an activity that can be time consuming on a large system. For example, large timesharing systems or networks of smaller systems can easily have 5,000 to 10,000 active accounts. Moreover, B may have passed the access right for F to another user, so A may not know that F's access exists and should be revoked. This problem is particularly serious in a network.

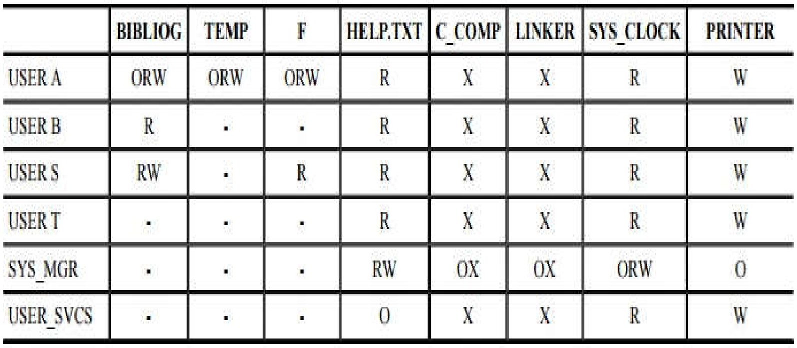
2.3.8 Access Control List

access-control list (ACL) is a **list of permissions associated with a system resource (object)**. An ACL specifies which users or system processes are granted access to objects, as well as what operations are allowed on given objects. Each entry in a typical ACL specifies a subject and an operation.

We can think of the directory as a listing of objects accessible by a single subject, and the access list as a table identifying subjects that can access a single object. The data in these two representations are equivalent, the distinction being the ease of use in given situations.

As an alternative, we can use an **access control matrix**, a table in which each row represents a subject, each column represents an object, and each entry is the set of access rights for that subject to that object. An example representation of an access control matrix is shown following table. In general, the access control matrix is sparse (meaning that most cells are empty): Most subjects do not have access rights to most objects. The access matrix can be represented as a list of triples, having the form <subject, object, rights>. Searching a large number of these triples is inefficient enough that this implementation is seldom used.

Table 2.3.8: Access Control Matrix



	BIBLIOG	TEMP	F	HELP.TXT	C_COMP	LINKER	SYS_CLOCK	PRINTER
USER A	ORW	ORW	ORW	R	X	X	R	W
USER B	R	-	-	R	X	X	R	W
USER S	RW	-	R	R	X	X	R	W
USER T	-	-	-	R	X	X	R	W
SYS_MGR	-	-	-	RW	OX	OX	ORW	O
USER_SVCS	-	-	-	O	X	X	R	W

2.4 SUMMARY

Basically the operating system objects are plays important role for operating system and it is just like protecting one user's programs and data from other users' programs became an important issue in multi-programmed operating systems and memory. An ACL specifies which users or system processes are granted access to objects, as well as what operations are allowed on given objects. Each entry in a typical ACL specifies a subject and an operation.

2.5 REFERENCE FOR FURTHER READING

1. https://www.brainkart.com/article/File-Protection-Mechanisms_9610/

❖❖❖❖

3

NETWORK SECURITY-I

Unit Structure :

3.0 OBJECTIVES

After going through this unit, you will be able to:

1. To understand the need for network security
2. To identify and classify particular examples of attacks
3. To define the terms vulnerability, threat and attack
4. To identify physical points of vulnerability in simple networks
5. To identify firewall, Intrusion detection and prevention system.

3.1 INTRODUCTION

Network security mainly refers to an evaluation which is taken by any enterprise or an organization to secure or to make safe its computer network. Its main role is to maintain confidentiality and accessibility of the data and network. In every enterprise or an organization which generally manages or handles a large amount of data needs some infusion aginst many cyber threats. The most common example of network security is to handle the password protection. Network security has become the main subject of cyber security. Network security deals with various levels which helps in performing the activities needed for handles the large amount of data in a secure manner.

3.2 NETWORK SECURITY

Network security is a broad term that covers a multitude of technologies, devices and processes. In its simplest term, it is a set of rules and configurations designed to protect the integrity, confidentiality and accessibility of computer networks and data using both software and hardware technologies. Every organization, regardless of size, industry or infrastructure, requires a degree of network security solutions in place to protect it from the ever-growing landscape of cyber threats in the wild today.

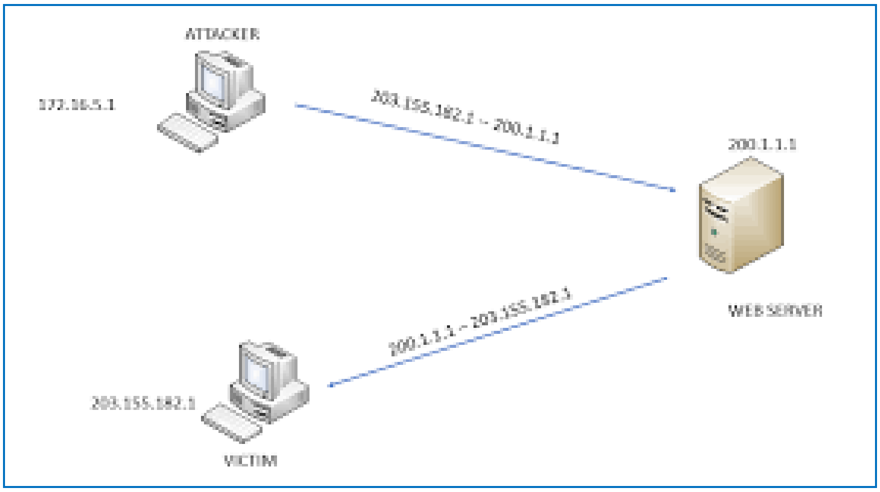
Today's network architecture is complex and is faced with a threat environment that is always changing and attackers that are always trying to find and exploit vulnerabilities. These vulnerabilities can exist in a broad number of areas, including devices, data, applications, users and locations. For this reason, there are many network security management tools and applications in use today that address individual threats and exploits and also regulatory non-compliance. When just a few minutes of downtime can cause widespread disruption and massive damage to an organization's bottom line and reputation, it is essential that these protection measures are in place.

3.2.1 Different types of network layer attacks

The main responsibility of the network layer is to transmit the packets from the source to the destination by finding the best route, which is the route that has the lowest cost and shortest path from the source to the destination. The goal of the attacks on the Network Layer is to disrupt the path between the source and destination that is chosen from the routing protocols. [3]. Some of the most used methods to attack the network layer are below:

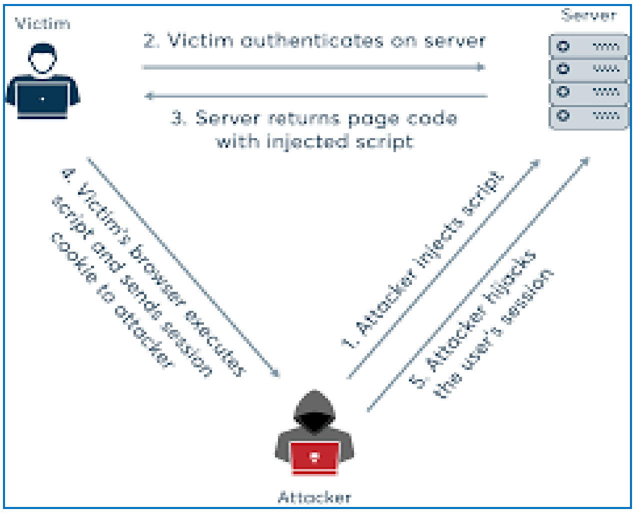
a. IP spoofing attack:

This technique is used from the attackers to gain unauthorized access to the servers. The attacker will send messages to the server not with his own IP address, but with a "trusted" IP address. In this way the server will not understand that it is getting traffic from an attacker. After the attacker will find the "trusted IP" address, will modify the headers of the packets in the way that the attacked server will think the packets are coming from "trusted" IP. The main route cause of DDoS (Distributed Denial of Service) attacks is IP spoofing.



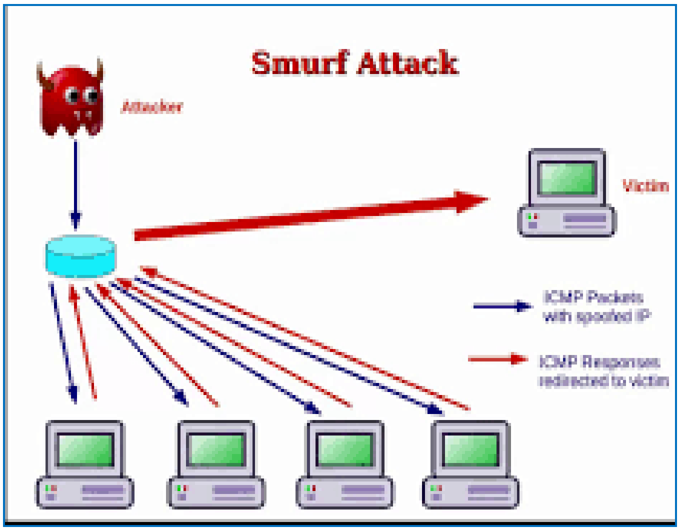
b. Hijacking attack

Another method used to attack the network layer is hijacking. These attacks are easy to implement, but difficult to detect. The basic idea of the attack is to disrupt a session between client and server and take over the IP address of the trusted client. The next step of the attacker is to discontinue the communication between the server and the trusted client and to create a new session with server by pretending to be the trusted client. After the new connection is created, the attacker can take the data he wants from server until this attack will be detect from the victim client (trusted IP) or from the server. These attacks happens when the server is unware for a certain amount of time of the existence of an attacker on the network and the legitimate client that is disconnected from the network (from the attacker), the attacker will make sure to keep it unaware of the attack. The attackers usually use different tools to monitor the victim's network, such as WirelessMon, Ethereal, Netstumbler. If the wireless network will not use advanced encryption methods, and different authentication mechanism, the possibilities to be attacked using hijacking will be high.


Victim
Server
2. Victim authenticates on server
3. Server returns page code with injected script
4. Victim's browser executes script and sends session cookie to attacker
1. Attacker injects script
5. Attacker hijacks the user's session
Attacker

c. The Smurf attack

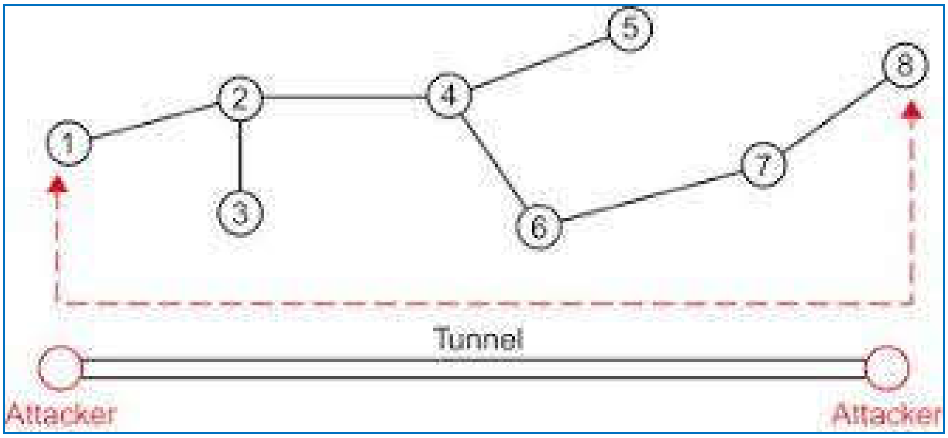
This attacking technique is a DoS (Denial of Service) attack that happen on the network layer. These attacks are very easy to implement. The idea of this attack is to overload a server with packets. The attacker will send a high number of packets from a spoofed IP address to the server. The main goal of these attacks is to disable the service the network is providing. Many techniques of attacking are used to achieve this goal. When the attacker wants to realize a Smurf attack, he will transmit to the intended victim a large number of Internet Control Message Protocol (ICMP) by using an IP broadcast address. To achieve this, the attackers use a program called “smurf” that builds a network packet which appears at the attacked server as it is coming from the trusted IP address. When the attacked server will receive this ICMP packets, by default the server will response to the request. The “smurf” program will generate the necessary amount of ICMP requests to overload the victim with ICMP requests and responses until this device will not be able to provide the necessary services on the network.


Smurf Attack
Attacker
Victim
ICMP Packets with spoofed IP
ICMP Responses redirected to victim

d. Wormhole attacks

These attacks are the most severe attacks and complicated attacks in wireless network. Wormhole attacks are very hard to detect and to protect from them. Even when all the communication on the wireless network provides authenticity and confidentiality, a wormhole attack can happen. The attackers will record the packets at one point of the network and retransmits them to another point of the network using private highspeed network, and then replays them into the network from that point. These kinds of attacks are a serious threat against network routing protocols.

Usually in the wireless network, routing protocols have implemented different mechanisms to defend against wormhole attacks; otherwise, the routing protocols will not be able to route more than one hop.



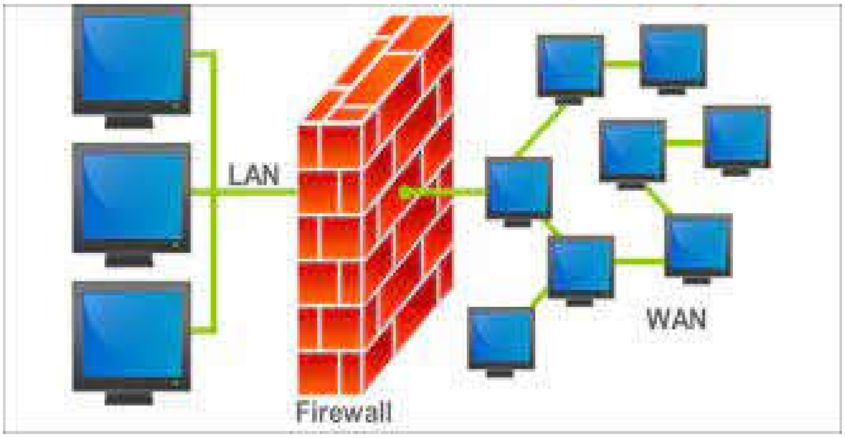
3.2.2 Firewall

Firewalls were officially invented in the early 1990s.

A firewall is a device that filters all traffic between a protected or "inside" network and a less trustworthy or "outside" network. Usually, a firewall runs on a dedicated device; because it is a single point through which traffic is channeled, performance is important, which means nonfirewall functions should not be done on the same machine. Because a firewall is executable code, an attacker could compromise that code and execute from the firewall's device. Thus, the fewer pieces of code on the device, the fewer tools the attacker would have by compromising the firewall. Firewall code usually runs on a proprietary or carefully minimized operating system.

The purpose of a firewall is to keep "bad" things outside a protected environment. To accomplish that, firewalls implement a security policy that is specifically designed to address what bad things might happen. For example, the policy might be to prevent any access from outside (while still allowing traffic to pass from the inside to the outside). Alternatively, the policy might permit accesses only from certain places, from certain users, or for certain activities. Part of the challenge of protecting a network with a firewall is determining which security policy meets the needs of the installation.

firewall is a type of host; it often is as programmable as a good-quality workstation.



Personal Firewalls:

Firewalls typically protect a (sub)network of multiple hosts. University students and employees in offices are behind a real firewall. Increasingly, home users, individual workers, and small businesses use cable modems or DSL connections with unlimited, always-on access. These people need a firewall, but a separate firewall computer to protect a single workstation can seem too complex and expensive. These people need a firewall's capabilities at a lower price. A personal firewall is an application program that runs on a workstation to block unwanted traffic, usually from the network. A personal firewall can complement the work of a conventional firewall by screening the kind of data a single host will accept, or it can compensate for the lack of a regular firewall, as in a private DSL or cable modem connection. Just as a network firewall screens incoming and outgoing traffic for that network, a personal firewall screens traffic on a single workstation. A workstation could be vulnerable to malicious code or malicious active agents' leakage of personal data stored on the workstation, and vulnerability scans to identify potential weaknesses.

3.2.3 ACL (Access Control List)

An access control list (ACL) contains rules that grant or deny access to certain digital environments. There are two types of ACLs:

- Filesystem ACLs—filter access to files and/or directories. Filesystem ACLs tell operating systems which users can access the system, and what privileges the users are allowed.
- Networking ACLs—filter access to the network. Networking ACLs tell routers and switches which type of traffic can access the network, and which activity is allowed.
- Originally, ACLs were the only way to achieve firewall protection. Today, there are many types of firewalls and alternatives to ACLs.

However, organizations continue to use ACLs in conjunction with technologies like virtual private networks (VPNs) that specify which traffic should be encrypted and transferred through a VPN tunnel.

Reasons to use an ACL:

- Traffic flow control
- Restricted network traffic for better network performance
- A level of security for network access specifying which areas of the server/network/service can be accessed by a user and which cannot
- Granular monitoring of the traffic exiting and entering the system

How ACL Works

A filesystem ACL is a table that informs a computer operating system of the access privileges a user has to a system object, including a single file or a file directory. Each object has a security property that connects it to its access control list. The list has an entry for every user with access rights to the system.

Typical privileges include the right to read a single file (or all the files) in a directory, to execute the file, or to write to the file or files. Operating systems that use an ACL include, for example, Microsoft Windows NT/2000, Novell's Netware, Digital's OpenVMS, and UNIX-based systems.

When a user requests an object in an ACL-based security model, the operating system studies the ACL for a relevant entry and sees whether the requested operation is permissible.

Networking ACLs are installed in routers or switches, where they act as traffic filters. Each networking ACL contains predefined rules that control which packets or routing updates are allowed or denied access to a network.

Routers and switches with ACLs work like packet filters that transfer or deny packets based on filtering criteria. As a Layer 3 device, a packet-filtering router uses rules to see if traffic should be permitted or denied access. It decides this based on source and destination IP addresses, destination port and source port, and the official procedure of the packet.

Types of Access Control Lists

Access control lists can be approached in relation to two main categories:

Standard ACL

An access-list that is developed solely using the source IP address. These access control lists allow or block the entire protocol suite. They don't differentiate between IP traffic such as UDP, TCP, and HTTPS. They use numbers 1-99 or 1300-1999 so the router can recognize the address as the source IP address.

Extended ACL

An access-list that is widely used as it can differentiate IP traffic. It uses both source and destination IP addresses and port numbers to make sense of IP traffic. You can also specify which IP traffic should be allowed or denied. They use the numbers 100-199 and 2000-2699.

Access Control Lists (ACLs) are a collection of permits and deny conditions, called rules, that provide security by blocking unauthorized users and allowing authorized users to access specific resources.

ACLs can also provide traffic flow control, restrict contents of routing updates, and decide which types of traffic are forwarded or blocked. Normally ACLs reside in a firewall router or in a router connecting two internal networks.

You can set up ACLs to control traffic at Layer 2, Layer 3, or Layer 4. MAC ACLs operate on Layer 2. IP ACLs operate on Layers 3 and 4.

ACL support features include Flow-based Mirroring and ACL Logging.

- Flow-based mirroring is the ability to mirror traffic that matches a permit rule to a specific physical port or LAG. Flow-based mirroring is similar to the redirect function, except that in flow-based mirroring a copy of the permitted traffic is delivered to the mirror interface while the packet itself is forwarded normally through the device. You cannot configure a given ACL rule with mirror and redirect attributes.

- ACL Logging provides a means for counting the number of “hits” against an ACL rule. When you configure ACL Logging, you augment the ACL deny rule specification with a ‘log’ parameter that enables hardware hit count collection and reporting. FASTPATH uses a fixed five minute logging interval, at which time trap log entries are written for each ACL logging rule that accumulated a non-zero hit count during that interval. You cannot configure the logging interval.

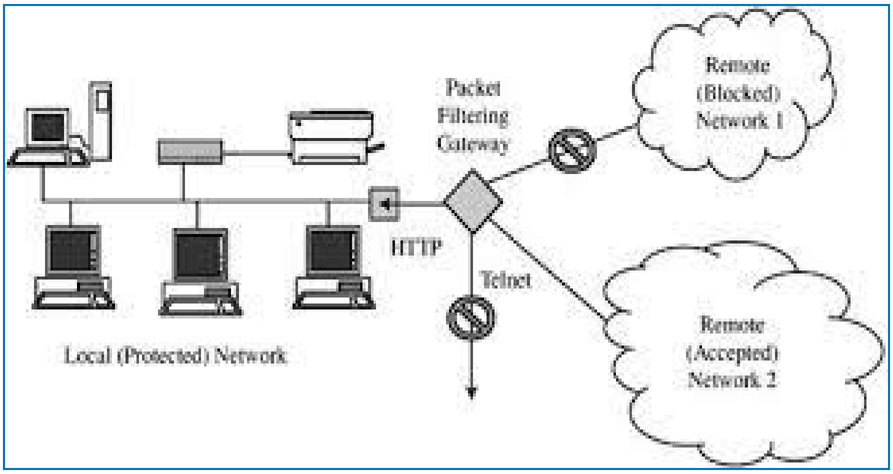
Using ACLs to mirror traffic is called flow-based mirroring because the traffic flow is defined by the ACL classification rules. This is in contrast to port mirroring, where all traffic encountered on a specific interface is replicated on another interface.

To configure ACL we need to follow steps:

1. Create a MAC ACL by specifying a name.
2. Create an IP ACL by specifying a number.
3. Add new rules to the ACL.
4. Configure the match criteria for the rules.
5. Apply the ACL to one or more interfaces.

3.2.4 Packet Filtering

A packet filtering gateway or screening router is the simplest, and in some situations, the most effective type of firewall. A packet filtering gateway controls access to packets on the basis of packet address (source or destination) or specific transport protocol type (such as HTTP web traffic). As described earlier in this chapter, putting ACLs on routers may severely impede their performance. But a separate firewall behind (on the local side) of the router can screen traffic before it gets to the protected network. Figure shows a packet filter that blocks access from (or to) addresses in one network; the filter allows HTTP traffic but blocks traffic using the Telnet protocol.


Packet Filtering Gateway
Remote (Blocked) Network 1
HTTP
Telnet
Remote (Accepted) Network 2
Local (Protected) Network

3.2.5 DMZ

A demilitarized zone (DMZ) is a perimeter network that protects an organization's internal local-area network (LAN) from untrusted traffic.

A common demilitarized zone meaning is a subnetwork that sits between the public internet and private networks. It exposes external-facing services to untrusted networks and adds an extra layer of security conditions to protect the sensitive data stored on internal networks, using firewalls to filter traffic.

The end goal of a DMZ is to allow an organization to access untrusted networks, such as the internet, while ensuring its private network or LAN remains secure. Organizations typically store external-facing services and resources, as well as servers for the Domain Name System (DNS), File Transfer Protocol (FTP), mail, proxy, Voice over Internet Protocol (VoIP), and web servers, in the DMZ.

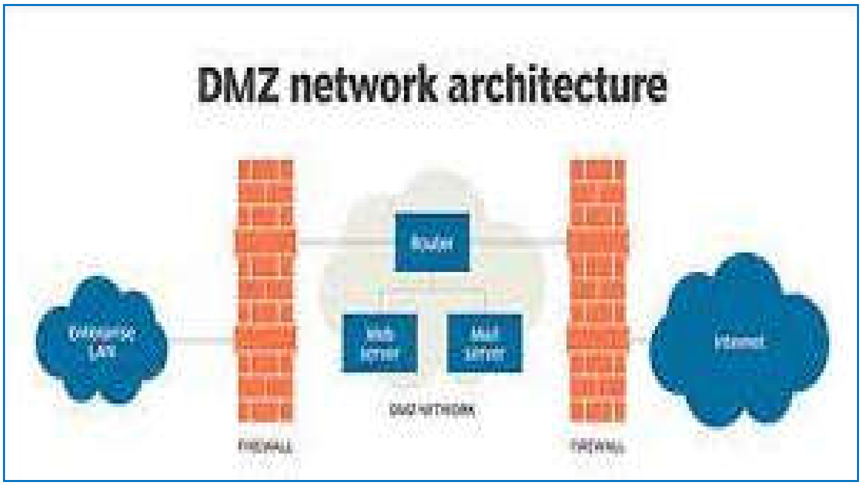
These servers and resources are isolated and given limited access to the LAN to ensure they can be accessed via the internet but the internal LAN cannot. As a result, a DMZ approach makes it more difficult for a hacker to gain direct access to an organization's data and internal servers via the internet.

A DMZ network provides a buffer between the internet and an organization's private network. The DMZ is isolated by a security gateway, such as a firewall, that filters traffic between the DMZ and a LAN. The default DMZ server is protected by another security gateway that filters traffic coming in from external networks.

It is ideally located between two firewalls, and the DMZ firewall setup ensures incoming network packets are observed by a firewall—or other security tools—before they make it through to the servers hosted in the DMZ. This means that even if a sophisticated attacker is able to get past the first firewall, they must also access the hardened services in the DMZ before they can do damage to a business.

If an attacker is able to penetrate the external firewall and compromise a system in the DMZ, they then also have to get past an internal firewall before gaining access to sensitive corporate data. A highly skilled bad actor may well be able to breach a secure DMZ, but the resources within it should sound alarms that provide plenty of warning that a breach is in progress.

Organizations that need to comply with regulations, such as the Health Insurance Portability and Accountability Act (HIPAA), will sometimes install a proxy server in the DMZ. This enables them to simplify the monitoring and recording of user activity, centralize web content filtering, and ensure employees use the system to gain access to the internet.



3.2.6 Alerts and Audit Trials

Audit trails maintain a record of system activity both by system andapplication processes and by user activity of systems and applications. Inconjunction with appropriate tools and procedures, audit trails can assistin detecting security violations, performance problems, and flaws in

applications. This bulletin focuses on audit trails as a technical controland discusses the benefits and objectives of audit trails, the types ofaudit trails, and some common implementation issues.

An audit trail is a series of records of computer events, about anoperating system, an application, or user activities. A computer systemmay have several audit trails, each devoted to a particular type ofactivity. Auditing is a review and analysis of management, operational,and technical controls. The auditor can obtain valuable information aboutactivity on a computer system from the audit trail. Audit trails improvethe auditability of the computer system.

Audit trails may be used as either a support for regular system operationsor a kind of insurance policy or as both of these. As insurance, audittrails are maintained but are not used unless needed, such as after asystem outage. As a support for operations, audit trails are used to helpsystem administrators ensure that the system or resources have not beenharmed by hackers, insiders, or technical problems.

BENEFITS AND OBJECTIVES

Audit trails can provide a means to help accomplish severalsecurity-related objectives, including individual accountability,reconstruction of events (actions that happen on a computer system), intrusion detection, and problem analysis.

Individual Accountability

Audit trails are a technical mechanism that help managers maintainindividual accountability. By advising users that they are personallyaccountable for their actions, which are tracked by an audit trail thatlogs user activities, managers can help promote proper user behavior.Users are less likely to attempt to circumvent security policy if they knowthat their actions will be recorded in an audit log.

For example, audit trails can be used in concert with access controls toidentify and provide information about users suspected of impropermodification of data (e.g., introducing errors into a database). An audittrail may record "before" and "after" versions of records. (Depending upon

the size of the file and the capabilities of the audit logging tools, thismay be very resource-intensive.) Comparisons can then be made between theactual changes made to records and what was expected. This can helpmanagement determine if errors were made by the user, by the system orapplication software, or by some other source.

Audit trails work in concert with logical access controls, which restrictuse of system resources. Granting users access to particular resourcesusually means that they need that access to accomplish their job.

Authorized access, of course, can be misused, which is where audit trailanalysis is useful. While users cannot be prevented from using resourcesto which they have legitimate access authorization, audit trail analysis isused to examine their actions. For example, consider a personnel office in which users have access to those personnel records for which they areresponsible. Audit trails can reveal that an individual is printing farmore records than the average user, which could indicate the selling ofpersonal data. Another example may be an engineer who is using a computerfor the design of a new product. Audit trail analysis could reveal that anoutgoing modem was used extensively by the engineer the week beforequitting. This could be used to investigate whether proprietary data fileswere sent to an unauthorized party.

Reconstruction of Events

Audit trails can also be used to reconstruct events after a problem hasoccurred. Damage can be more easily assessed by reviewing audit trails ofsystem activity to pinpoint how, when, and why normal operations ceased.Audit trail analysis can often distinguish between operator-induced errors

(during which the system may have performed exactly as instructed) orsystem-created errors (e.g., arising from a poorly tested piece ofreplacement code). If, for example, a system fails or the integrity of afile (either program or data) is questioned, an analysis of theaudit trail can reconstruct the series of steps taken by the system, theusers, and the application. Knowledge of the conditions that existed atthe time of, for example, a system crash, can be useful in avoiding futureoutages. Additionally, if a technical problem occurs (e.g., the corruptionof a data file) audit trails can aid in the recovery process (e.g., byusing the record of changes made to reconstruct the file).

Intrusion Detection

Intrusion detection refers to the process of identifying attempts topenetrate a system and gain unauthorized access. If audit trails have beendesigned and implemented to record appropriate information, they can assistin intrusion detection. Although normally thought of as a real-time

effort, intrusions can be detected in real time, by examining audit recordsas they are created (or through the use of other kinds of warningflags/notices), or after the fact (e.g., by examining audit records in abatch process).

Real-time intrusion detection is primarily aimed at outsiders attempting to gain unauthorized access to the system. It may also be used to detectchanges in the system's performance indicative of, for example, a virus orworm attack (forms of malicious code). There may be difficulties inimplementing real-time auditing, including unacceptable system performance.

After-the-fact identification may indicate that unauthorized access wasattempted (or was successful). Attention can then be given to damageassessment or reviewing controls that were attacked.

Problem Analysis

Audit trails may also be used as on-line tools to help identify problemsother than intrusions as they occur. This is often referred to asreal-time auditing or monitoring. If a system or application is deemed tobe critical to an organization's business or mission, real-time auditingmay be implemented to monitor the status of these processes (although, asnoted above, there can be difficulties with real-time analysis). Ananalysis of the audit trails may be able to verify that the system operatednormally (i.e., that an error may have resulted from operator error, asopposed to a

system-originated error). Such use of audit trails may be complemented bysystem performance logs. For example, a significant increase in the use ofsystem resources (e.g., disk file space or outgoing modem use) couldindicate a security problem.

AUDIT TRAILS AND LOGS

A system can maintain several different audit trails concurrently. Thereare typically two kinds of audit records, (1) an event-oriented log and (2)a record of every keystroke, often called keystroke monitoring.Event-based logs usually contain records describing system events,application events, or user events.

An audit trail should include sufficient information to establish whatevents occurred and who (or what) caused them. In general, an event recordshould specify when the event occurred, the user ID associated with the event, the program or command used to initiate the event, and the result.

Date and time can help determine if the user was a masquerade or theactual person specified.

3.2.7 IDS (Intrusion Detection Systems)

An intrusion detection system (IDS) is a device, typically another separate computer, that monitors activity to identify malicious or suspicious events. Kemmerer and Vigna [KEM02] survey the history of IDSs. An IDS is a sensor, like a smoke detector, that raises an alarm if specific things occur. A model of an IDS is shown in. The components in the figure are the four basic elements of an intrusion detection system, based on the Common Intrusion Detection Framework of [STA96]. An IDS receives raw inputs from sensors. It saves those inputs, analyzes them, and takes some controlling action.

IDSs perform a variety of functions:

- monitoring users and system activity
- auditing system configuration for vulnerabilities and misconfigurations
- assessing the integrity of critical system and data files

- recognizing known attack patterns in system activity
- identifying abnormal activity through statistical analysis
- managing audit trails and highlighting user violation of policy or normal activity
- correcting system configuration errors
- installing and operating traps to record information about intruders

Types of IDSs

The two general types of intrusion detection systems are **signature based and heuristic**.

Signature-based intrusion detection systems perform simple pattern-matching and report situations that match a pattern corresponding to a known attack type.

Heuristic intrusion detection systems, also known as anomaly based, build a model of acceptable behavior and flag exceptions to that model; for the future, the administrator can mark a flagged behavior as acceptable so that the heuristic IDS will now treat that previously unclassified behavior as acceptable. Intrusion detection devices can be network based or host based. A network-based IDS is a stand-alone device attached to the network to monitor traffic throughout that network; a host-based IDS runs on a single workstation or client or host, to protect that one host.

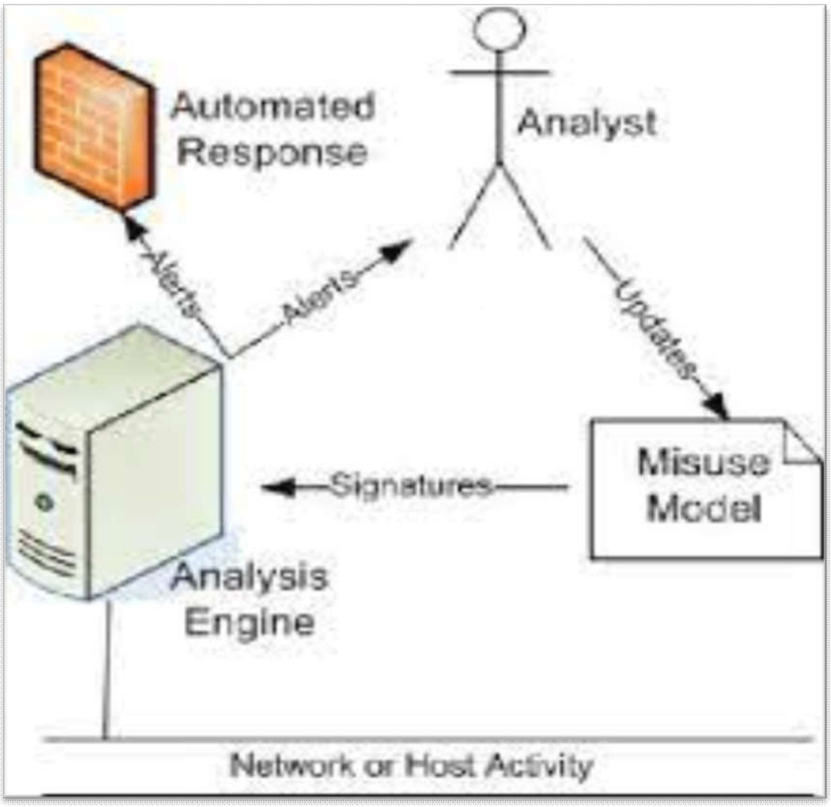
3.2.8 Signature-Based Intrusion Detection

A simple signature for a known attack type might describe a series of TCP SYN packets sent to many different ports in succession and at times close to one another, as would be the case for a port scan. An intrusion detection system would probably find nothing unusual in the first SYN, say, to port 80, and then another (from the same source address) to port 25. But as more and more ports receive SYN packets, especially ports that are not open, this pattern reflects a possible port scan. Similarly, some implementations of the protocol stack fail if they receive an ICMP packet with a data length of 65535 bytes, so such a packet would be a pattern for which to watch.

The problem with signature-based detection is the signatures themselves. An attacker will try to modify a basic attack in such a way that it will not match the known signature of that attack. For example, the attacker may convert lowercase to uppercase letters or convert a symbol such as "blank space" to its character code equivalent %20. The IDS must necessarily work from a canonical form of the data stream in order to recognize that %20 matches a pattern with a blank space. The attacker may insert malformed packets that the IDS will see, to intentionally cause a pattern mismatch; the protocol handler stack will discard the packets because of the malformation. Each of these variations could be detected by an IDS, but more signatures require additional work for the IDS, which reduces performance.

Of course, signature-based IDSs cannot detect a new attack for which a signature is not yet installed in the database. Every attack type starts as a new pattern at some time, and the IDS is helpless to warn of its existence.

Signature-based intrusion detection systems tend to use statistical analysis. This approach uses statistical tools both to obtain sample measurements of key indicators (such as amount of external activity, number of active processes, number of transactions) and to determine whether the collected measurements fit the predetermined attack signatures. Ideally, signatures should match every instance of an attack, match subtle variations of the attack, but not match traffic that is not part of an attack. However, this goal is grand but unreachable.



3.2.9 Heuristic Intrusion Detection (Anomaly based)

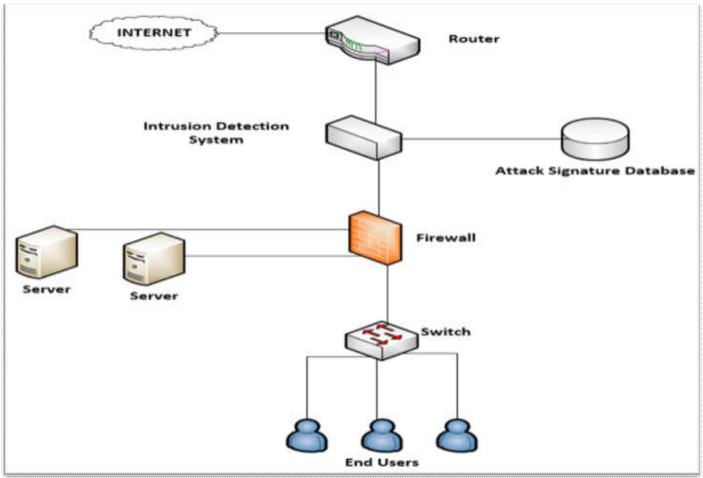
Because signatures are limited to specific, known attack patterns, another form of intrusion detection becomes useful. Instead of looking for matches, heuristic intrusion detection looks for behavior that is out of the ordinary. The original work in this area (for example, [TEN90]) focused on the individual, trying to find characteristics of that person that might be helpful in understanding normal and abnormal behavior. For example, one user might always start the day by reading e-mail, write many documents using a word processor, and occasionally back up files. These actions would be normal.

This user does not seem to use many administrator utilities. If that person tried to access sensitive system management utilities, this new behavior might be a clue that someone else was acting under the user's identity. If we think of a compromised system in use, it starts clean, with no intrusion, and it ends dirty, fully compromised. There may be no point in the trace of use in which the system changed from clean to dirty; it was more likely that little dirty events occurred, occasionally at first and then increasing as the system became more deeply compromised.

Any one of those events might be acceptable by itself, but the accumulation of them and the order and speed at which they occurred could have been signals that something unacceptable was happening. The inference engine of an intrusion detection system performs continuous analysis of the system, raising an alert when the system's dirtiness exceeds a threshold. Inference engines work in two ways. Some, called state-based intrusion detection systems, see the system going through changes of overall state or configuration.

They try to detect when the system has veered into unsafe modes. Others try to map current activity onto a model of unacceptable activity and raise an alarm when the activity resembles the model. These are called model-based intrusion detection systems. This approach has been extended to networks in [MUK94]. Later work (for example, [FOR96, LIN99]) sought to build a dynamic model of behavior, to accommodate variation and evolution in a person's actions over time. The technique compares real activity with a known representation of normality. Alternatively, intrusion detection can work from a model of known bad activity. For example, except for a few utilities (login, change password, create user), any other attempt to access a password file is suspect. This form of intrusion detection is known as misuse intrusion detection. In this work, the real activity is compared against a known suspicious area.

All heuristic intrusion detection activity is classified in one of three categories: good/benign, suspicious, or unknown. Over time, specific kinds of actions can move from one of these categories to another, corresponding to the IDS's learning whether certain actions are acceptable or not. As with pattern-matching, heuristic intrusion detection is limited by the amount of information the system has seen (to classify actions into the right category) and how well the current actions fit into one of these categories.



3.2.10 IPS (Intrusion Prevention System)

An **Intrusion Prevention System** (IPS) is a network security/threat prevention technology that examines network traffic flows to detect and prevent vulnerability exploits. Vulnerability exploits usually come in the form of malicious inputs to a target application or service that attackers use to interrupt and gain control of an application or machine. Following a successful exploit, the attacker can disable the target application (resulting in a denial-of-service state), or can potentially access to all the rights and permissions available to the compromised application.

The IPS often sits directly behind the firewall and provides a complementary layer of analysis that negatively selects for dangerous content. Unlike its predecessor the Intrusion Detection System (IDS)—which is a passive system that scans traffic and reports back on threats—the IPS is placed inline (in the direct communication path between source and destination), actively analyzing and taking automated actions on all traffic flows that enter the network. Specifically, these actions include:

- Sending an alarm to the administrator (as would be seen in an IDS)
- Dropping the malicious packets
- Blocking traffic from the source address
- Resetting the connection

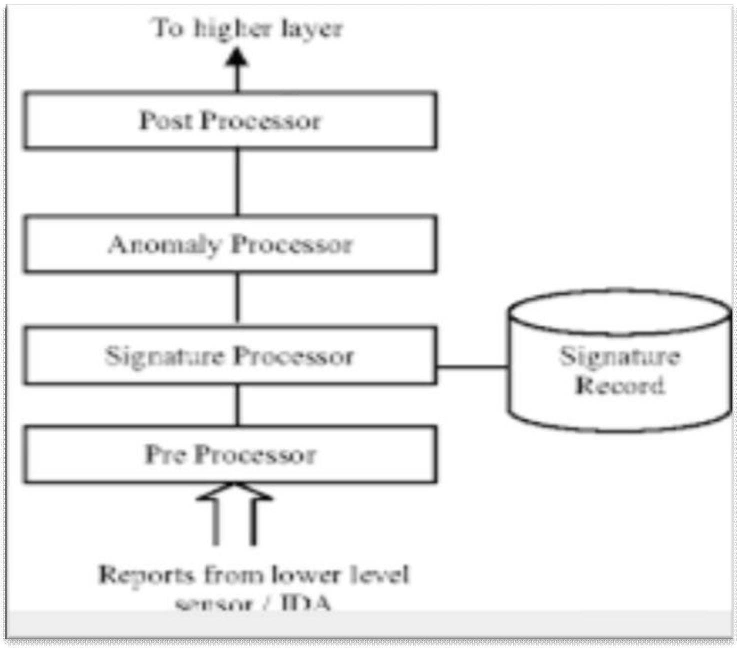
As an inline security component, the IPS must work efficiently to avoid degrading network performance. It must also work fast because exploits can happen in near real-time. The IPS must also detect and respond accurately, so as to eliminate threats and false positives (legitimate packets misread as threats).

3.2.11 policy-based detection

policy-based detection in which the IPS first requires administrators to make security policies -- when an event occurs that breaks a defined security policy, an alert is sent to system administrators.

If any threats are detected, an IPS tool is typically capable of sending alerts to the administrator, dropping any malicious network packets, and resetting connections by reconfiguring firewalls, repackaging payloads and removing infected attachments from servers.

IPS tools can help fend off denial-of-service (DoS) attacks, distributed denial-of-service (DDoS) attacks, worms, viruses or exploits, such as a zero-day exploit. According to Michael Reed, formerly of Top Layer Networks (acquired by Corero), an effective intrusion prevention system should perform more complex monitoring and analysis, such as watching and responding to traffic patterns, as well as individual packets. "Detection mechanisms can include address matching, HTTP [Hypertext Transfer Protocol] string and substring matching, generic pattern matching, TCP [Transmission Control Protocol] connection analysis, packet anomaly detection, traffic anomaly detection and TCP/UDP [User Datagram Protocol] port matching."

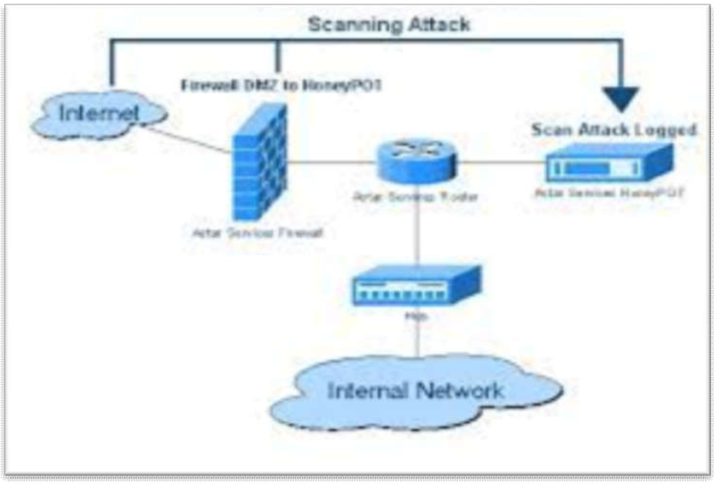


3.2.12 Honeypot based

How do you catch a mouse? You set a trap with bait (food the mouse finds attractive) and catch the mouse after it is lured into the trap. You can catch a computer attacker the same way. In a very interesting book, Cliff Stoll [STO89] details the story of attracting and monitoring the actions of an attacker. Cheswick [CHE90, CHE02] and Bellovin [BEL92c] tell a similar story.

These two cases describe the use of a honeypot: a computer system open to attackers. 480 480 You put up a honeypot for several reasons: to watch what attackers do, in order to learn about new attacks (so that you can strengthen your defenses against these new attacks) • to lure an attacker to a place in which you may be able to learn enough to identify and stop the attacker • to provide an attractive but diversionary playground, hoping that the attacker will leave your real system alone • A honeypot has no special features. It is just a computer system or a network segment, loaded with servers and devices and data. It may be protected with a firewall, although you want the attackers to have some access. There may be some monitoring capability, done carefully so that the monitoring is not evident to the attacker. The two difficult features of a honeypot are putting up a believable, attractive false environment and confining and monitoring the attacker surreptitiously.

Spitzner [SPI02, SPI03a] has done extensive work developing and analyzing honeypots. He thinks like the attacker, figuring what the attacker will want to see in an invaded computer, but as McCarty [MCC03] points out, it is always a race between attacker and defender. Spitzner also tries to move much of his data off the target platform so that the attacker will not be aware of the analysis and certainly not be able to modify or erase the data gathered. Raynal [RAY04a. RAY04b] discusses how to analyze the data collected.



3.3 WEB SERVER SECURITY

Web server security refers to the tools, technologies and processes that enable information security (IS) on a Web server. This broad term encompasses all processes that ensure that a working Internet server operates under a security policy.

Web server security is the security of any server that is deployed on a Worldwide Web domain or the Internet. It is implemented through several methods and in layers, typically, including the base operating system (OS) security layer, hosted application security layer and network security layer. OS security, which ensures access to authorized users only, operates a Web server's critical components and services. Application layer security ensures control over the content and services hosted on the Web server. Network security provides protection against Internet-based security exploits, viruses and attacks.

Secure Sockets Layer (SSL) certificates, HTTP Secure protocol and firewalling are several tools and technologies used to implement Web server security.

3.3.1 SSL (Secure Sockets Layer)

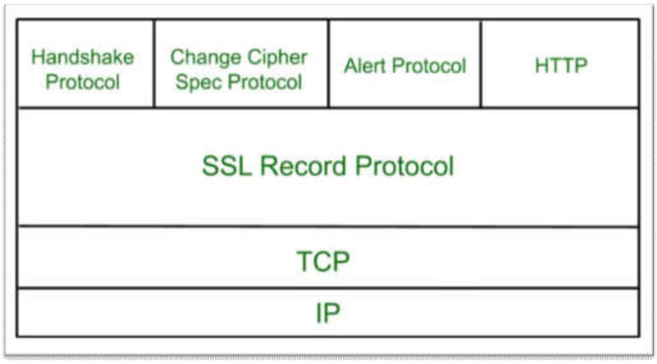
The SSL (Secure Sockets Layer) protocol was originally designed by Netscape to protect communication between a web browser and server. It is also known now as TLS, for transport layer security. SSL interfaces between applications (such as browsers) and the TCP/IP protocols to provide server authentication, optional client authentication, and an encrypted communications channel between client and server. Client and server 467 467 negotiate a mutually supported suite of encryption for session encryption and hashing; possibilities include triple DES and SHA1, or RC4 with a 128-bit key and MD5. To use SSL, the client requests an SSL session. The server responds with its public key certificate so that the client can determine the authenticity of the server. The client returns part of a symmetric session key encrypted under the server's public key. Both the server and client compute the session key, and then they switch to encrypted communication, using the shared session key. The protocol is simple but effective, and it is the most widely used secure communication protocol on the Internet. However, remember that SSL protects only from the client's browser to the server's decryption point (which is often only to the server's firewall or, slightly stronger, to the computer that runs the web application). Data are exposed from the user's keyboard to the browser and throughout the recipient's company. Blue Gem Security has developed a product called LocalSSL that encrypts data after it has been typed until the operating system delivers it to the client's browser, thus thwarting any keylogging Trojan horse that has become implanted in the user's computer to reveal everything the user types.

provides security to the data that is transferred between web browser and server. SSL encrypts the link between a web server and a browser which ensures that all data passed between them remain private and free from attack.

Secure Socket Layer Protocols:

- SSL record protocol
- Handshake protocol
- Change-cipher spec protocol
- Alert protocol

SSL Protocol Stack:

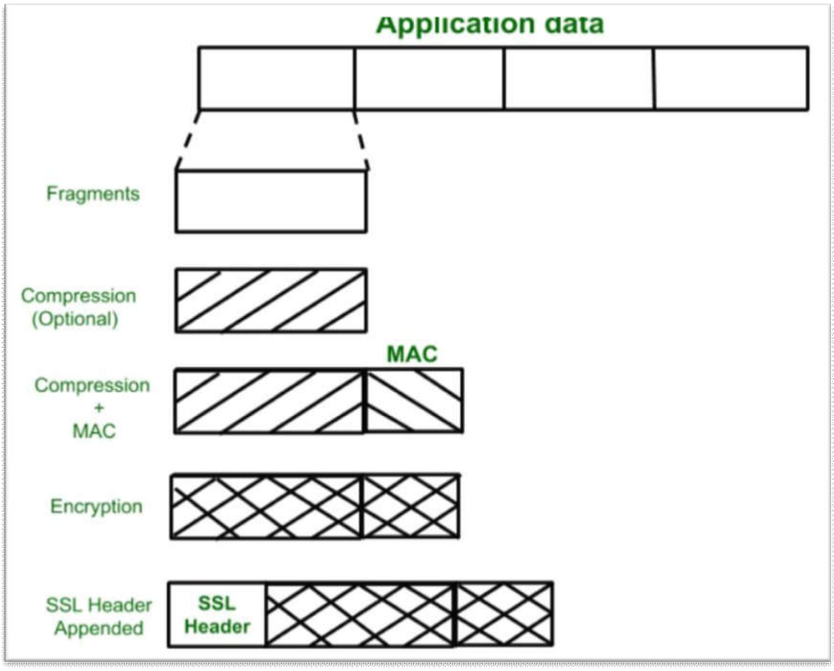


SSL Record Protocol:

SSL Record provides two services to SSL connection.

- Confidentiality
- Message Integrity

In the SSL Record Protocol application data is divided into fragments. The fragment is compressed and then encrypted MAC (Message Authentication Code) generated by algorithms like SHA (Secure Hash Protocol) and MD5 (Message Digest) is appended. After that encryption of the data is done and in last SSL header is appended to the data.



Handshake Protocol:

Handshake Protocol is used to establish sessions. This protocol allows the client and server to authenticate each other by sending a series of messages to each other. Handshake protocol uses four phases to complete its cycle.

- **Phase-1:** In Phase-1 both Client and Server send hello-packets to each other. In this IP session, cipher suite and protocol version are exchanged for security purposes.
- **Phase-2:** Server sends his certificate and Server-key-exchange. The server end phase-2 by sending the Server-hello-end packet.
- **Phase-3:** In this phase Client reply to the server by sending his certificate and Client-exchange-key.
- **Phase-4:** In Phase-4 Change-cipher suite occurred and after this Handshake Protocol ends

Change-cipher Protocol:

This protocol uses the SSL record protocol. Unless Handshake Protocol is completed, the SSL record Output will be in a pending state. After handshake protocol, the Pending state is converted into the current state. Change-cipher protocol consists of a single message which is 1 byte in length and can have only one value. This protocol's purpose is to cause the pending state to be copied into the current state.

Alert Protocol:

This protocol is used to convey SSL-related alerts to the peer entity. Each message in this protocol contains 2 bytes.

The level is further classified into two parts:

- **Warning:**
  This Alert has no impact on the connection between sender and receiver.
- **Fatal Error:**
- This Alert breaks the connection between sender and receiver.
- **Silent Features of Secure Socket Layer:**

  The advantage of this approach is that the service can be tailored to the specific needs of the given application.
- Secure Socket Layer was originated by Netscape.
- SSL is designed to make use of TCP to provide reliable end-to-end secure service.
- This is a two-layered protocol.

3.3.2 TLS (Transport Layer Security)

Transport Layer Security (TLS), the successor of the now-deprecated Secure Sockets Layer (SSL), is a cryptographic protocol designed to provide communications security over a computer network. The protocol is widely used in applications such as email, instant messaging, and voice over IP, but its use as the Security layer in HTTPS remains the most publicly visible.

The TLS protocol aims primarily to provide privacy and data integrity between two or more communicating computer applications. It runs in the application layer of the Internet and is itself composed of two layers: the TLS record and the TLS handshake protocols.

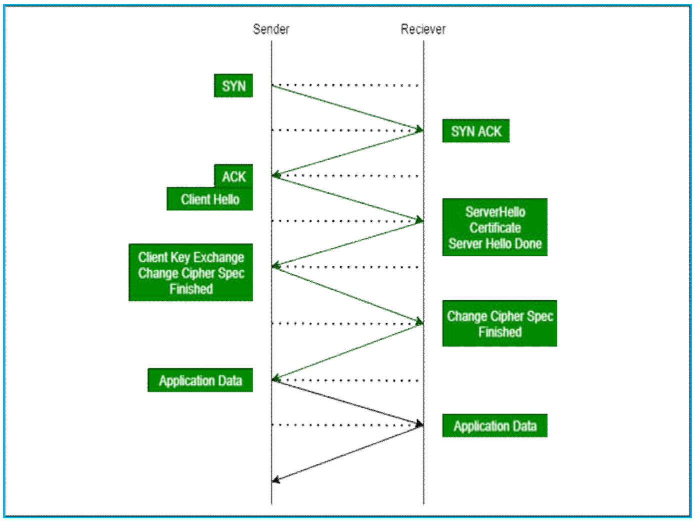
TLS is a proposed Internet Engineering Task Force (IETF) standard, first defined in 1999, and the current version is TLS 3.3 defined in August 2018. TLS builds on the earlier SSL specifications (1994, 1995, 1996) developed by Netscape Communications for adding the HTTPS protocol to their Navigator web browser.

Client-server applications use the TLS protocol to communicate across a network in a way designed to prevent eavesdropping and tampering.

Since applications can communicate either with or without TLS (or SSL), it is necessary for the client to request that the server sets up a TLS connection.[1] One of the main ways of achieving this is to use a different port number for TLS connections. For example, port 80 is typically used for unencrypted HTTP traffic while port 443 is the common port used for encrypted HTTPS traffic. Another mechanism is for the client to make a protocol-specific request to the server to switch the connection to TLS; for example, by making a STARTTLS request when using the mail and news protocols.

Once the client and server have agreed to use TLS, they negotiate a stateful connection by using a handshaking procedure.[2] The protocols use a handshake with an asymmetric cipher to establish not only cipher settings but also a session-specific shared key with which further communication is encrypted using a symmetric cipher. During this handshake, the client and server agree on various parameters used to establish the connection's security:

- The handshake begins when a client connects to a TLS-enabled server requesting a secure connection and the client presents a list of supported cipher suites (ciphers and hash functions).
- From this list, the server picks a cipher and hash function that it also supports and notifies the client of the decision.
- The server usually then provides identification in the form of a digital certificate. The certificate contains the server name, the trusted certificate authority (CA) that vouches for the authenticity of the certificate, and the server's public encryption key.
- The client confirms the validity of the certificate before proceeding.



3.3.3 Client Authentication

Client Authentication is the process by which users **securely access a server or remote computer by exchanging a Digital Certificate**. The Digital Certificate is in part seen as your 'Digital ID' and is used to cryptographically bind a customer, employee, or partner's identity to a unique Digital Certificate (typically including the name, company name and location of the Digital Certificate owner). The Digital Certificate can then be mapped to a user account and used to provide access control to network resources, web services and websites.

Just as organizations need to control which individual users have access to corporate networks and resources, they also need to be able to identify and control which machines and servers have access. Implementing device authentication means only machines with the appropriate credentials can access, communicate, and operate on corporate networks.

Organizations can leverage the registry information stored in Active Directory to automatically issue template-based and optionally configured certificates to all machines and servers residing within a single domain, or multiple domains in a single or multiple forest configuration.

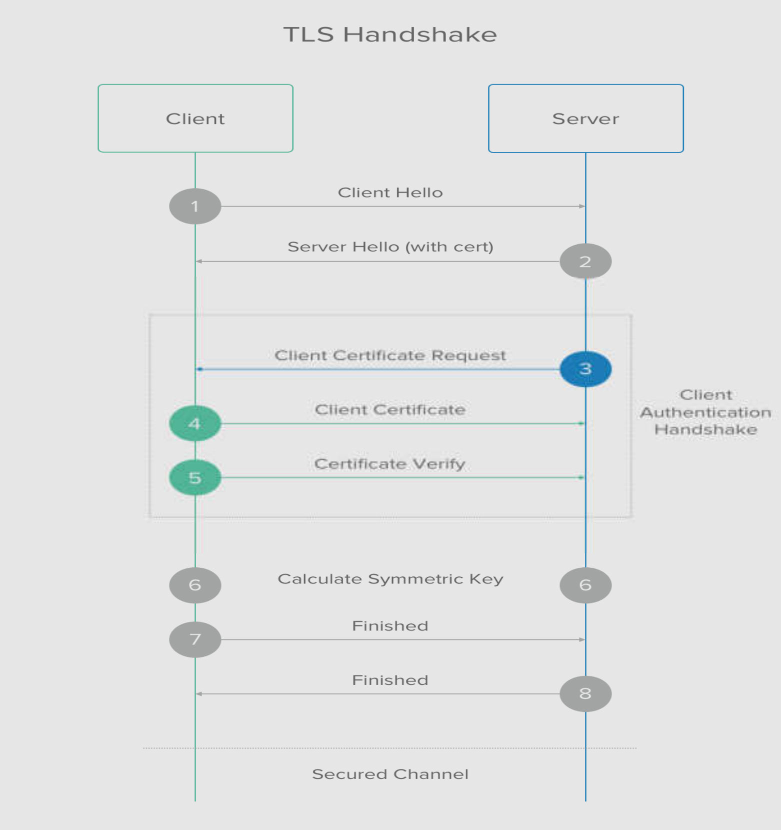
The Digital Certificates used for client and device authentication may look the same as any other Digital Certificate that you may already be using within your organization, such as certificates for securing web services (SSL) or email/document signatures (digital signatures), but Digital Certificates are likely to have a few different properties depending on the use.

Client authentication can be used to prevent unauthorized access, or simply to add a second layer of security to your current username and password combination. Client authentication and access control also enables organizations to meet regulatory and privacy compliancy, as well as fulfil internal security policies using PKI-based two-factor authentication – 'something you have' (a GlobalSign Digital Certificate) and 'something you know' (an internally managed password).

Client authentication has multiple benefits as an authentication method especially when compared to the basic username and password method:

- You can decide whether or not a user is required to enter a username and password
- Encrypts transactions over the network, identifies the server and validates any messages sent
- Validates the user identity using a trusted party (the Certificate Authority) and allows for centralized management of certificates which enables easy revocation
- Optional - you can configure the certificate so it cannot be exported to other devices, making it unique to the device it is installed on

- Restrict access by user, group, roles, or device based on Active Directory (using GlobalSign's Auto Enrolment Gateway (AEG) solution)
- Serves more purposes than authentication such as integrity and confidentiality
- Prevents malicious attacks/problems, including but not limited to phishing, keystroke logging and man-in-the-middle (MITM) attacks



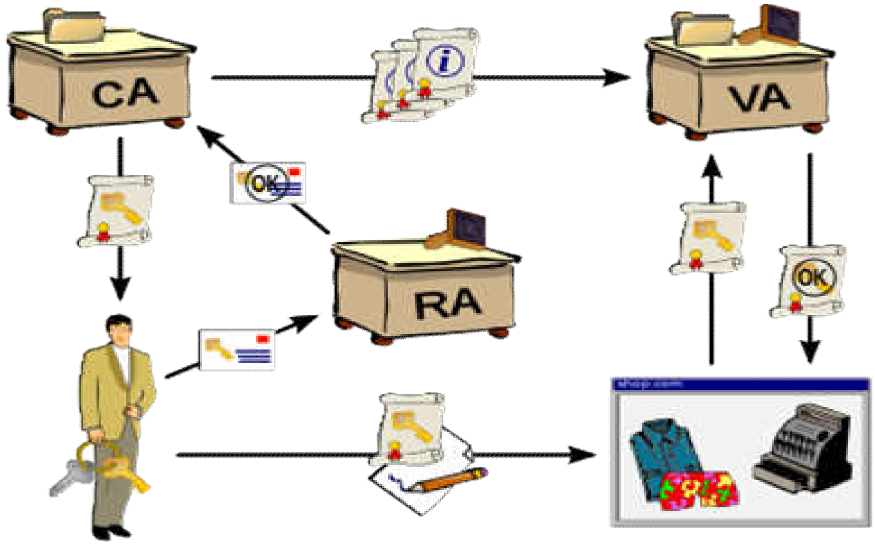
3.3.4 PKI (Public Key Infrastructure)

A **public key infrastructure** (**PKI**) is a set of roles, policies, hardware, software and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption. The purpose of a PKI is to facilitate the secure electronic transfer of information for a range of network activities such as e-commerce, internet banking and confidential email. It is required for activities where simple passwords are an inadequate authentication method and more rigorous proof is required to confirm the identity of the parties involved in the communication and to validate the information being transferred.

In cryptography, a PKI is an arrangement that binds public keys with respective identities of entities (like people and organizations). The binding is established through a process of registration and issuance of certificates at and by a certificate authority (CA). Depending on the assurance level of the binding, this may be carried out by an automated process or under human supervision. When done over a network, this requires using a secure certificate enrolment or certificate management protocol such as CMP.

The PKI role that may be delegated by a CA to assure valid and correct registration is called a registration authority (RA). Basically, an RA is responsible for accepting requests for digital certificates and authenticating the entity making the request.[1] The Internet Engineering Task Force's RFC 3647 defines an RA as "An entity that is responsible for one or more of the following functions: the identification and authentication of certificate applicants, the approval or rejection of certificate applications, initiating certificate revocations or suspensions under certain circumstances, processing subscriber requests to revoke or suspend their certificates, and approving or rejecting requests by subscribers to renew or re-key their certificates. RAs, however, do not sign or issue certificates (i.e., an RA is delegated certain tasks on behalf of a CA).While Microsoft may have referred to a subordinate CA as an RA,[3] this is incorrect according to the X.509 PKI standards. RAs do not have the signing authority of a CA and only manage the vetting and provisioning of certificates. So in the Microsoft PKI case, the RA functionality is provided either by the Microsoft Certificate Services web site or through Active Directory Certificate Services which enforces Microsoft Enterprise CA and certificate policy through certificate templates and manages certificate enrollment (manual or auto-enrollment). In the case of Microsoft Standalone CAs, the function of RA does not exist since all of the procedures controlling the CA are based on the administration and access procedure associate with the system hosting the CA and the CA itself rather than Active Directory. Most non-Microsoft commercial PKI solutions offer a stand-alone RA component.

An entity must be uniquely identifiable within each CA domain on the basis of information about that entity. A third-party validation authority (VA) can provide this entity information on behalf of the CA.



3.3.5 Encoding

Encoding is defined as the process of converting data from one form to another and has nothing to do with cryptography. It guarantees none of the 3 cryptographic properties of confidentiality, integrity, and authenticity because it involves no secret and is completely reversible. Encoding methods are considered public and are **used for data handling**. For example, data transmitted over the Internet require a specific format and URL-encoding our data will allow us to transmit them over the Internet. Similarly, in an HTML context, HTML-encoding our data is needed to adhere to the required HTML character format. Another popular encoding algorithm is base64. Base64 encoding is commonly used to encode binary data that need to be stored or transferred in media which are designed to process textual data. The examples above aim to point out that encoding's use case is only data handling and provides no protection for the encoded data.

In the Encoding method, data is transformed from one form to another. The main aim of encoding is to transform data into a form that is readable by most of the systems or that can be used by any external process. It can't be used for securing data, various publicly available algorithms are used for encoding.

Encoding can be used for reducing the size of audio and video files. Each audio and video file format has a corresponding coder-decoder (codec) program that is used to code it into the appropriate format and then decodes for playback.

Encoding data is a process involving changing data into a new format using a scheme. Encoding is a reversible process; data can be encoded to a new format and decoded to its original format. Encoding typically involves a publicly available scheme that is easily reversed. Encoding data is typically used to ensure integrity and usability of data and is commonly used when data cannot be transferred in its current format between systems or applications.

Encoding is not used to protect or secure data because it is easy to reverse.

An example of encoding is: Base64

Take a scenario where a user wants to upload a resume to a job application website and the web server stores.

3.3.6 Secure Electronic Transaction (SET

Secure Electronic Transaction or SET is a system that ensures the security and integrity of electronic transactions done using credit cards in a scenario. SET is not some system that enables payment but it is a security protocol applied to those payments. It uses different encryption and hashing techniques to secure payments over the internet done through credit cards. The SET protocol was supported in development by major organizations like Visa, Mastercard, Microsoft which provided its

Secure Transaction Technology (STT), and Netscape which provided the technology of Secure Socket Layer (SSL).

SET protocol restricts the revealing of credit card details to merchants thus keeping hackers and thieves at bay. The SET protocol includes Certification Authorities for making use of standard Digital Certificates like X.509 Certificate.

Before discussing SET further, let's see a general scenario of electronic transactions, which includes client, payment gateway, client financial institution, merchant, and merchant financial institution.

Requirements in SET :

The SET protocol has some requirements to meet, some of the important requirements are :

- It has to provide mutual authentication i.e., customer (or cardholder) authentication by confirming if the customer is an intended user or not, and merchant authentication.
- It has to keep the PI (Payment Information) and OI (Order Information) confidential by appropriate encryptions.
- It has to be resistive against message modifications i.e., no changes should be allowed in the content being transmitted.
- SET also needs to provide interoperability and make use of the best security mechanisms.

Participants in SET :

In the general scenario of online transactions, SET includes similar participants:

1. **Cardholder –** customer
2. **Issuer –** customer financial institution
3. **Merchant**
4. **Acquirer –** Merchant financial
5. **Certificate authority –** Authority that follows certain standards and issues certificates(like X.509V3) to all other participants.

SET functionalities :

- **Provide Authentication**
- **Merchant Authentication** – To prevent theft, SET allows customers to check previous relationships between merchants and financial

institutions. Standard X.509V3 certificates are used for this verification.

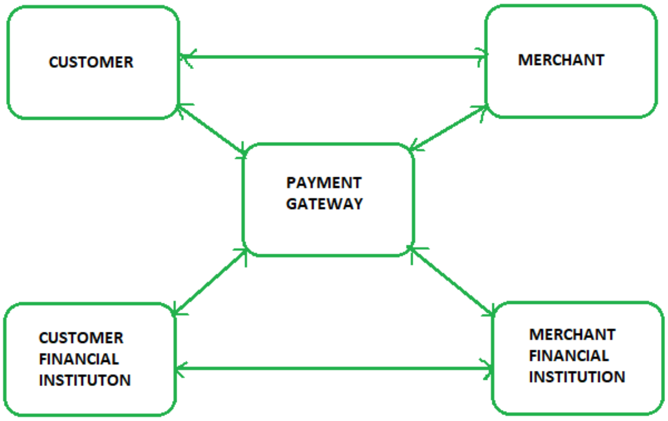
- **Customer / Cardholder Authentication** – SET checks if the use of a credit card is done by an authorized user or not using X.509V3 certificates.

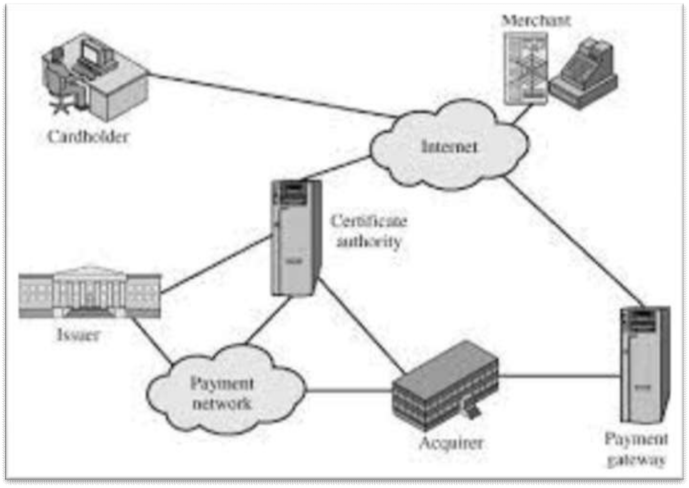
- **Provide Message Confidentiality**: Confidentiality refers to preventing unintended people from reading the message being transferred. SET implements confidentiality by using encryption techniques. Traditionally DES is used for encryption purposes.

- **Provide Message Integrity**: SET doesn't allow message modification with the help of signatures. Messages are protected against unauthorized modification using RSA digital signatures with SHA-1 and some using HMAC with SHA-1,

Dual Signature :

The dual signature is a concept introduced with SET, which aims at connecting two information pieces meant for two different receivers : **Order Information (OI) for merchant Payment Information (PI) for bank**





3.3.7 Kerberos

Kerberos is a system that supports authentication in distributed systems. Originally designed to work with secret key encryption, Kerberos, in its latest version, uses public key technology to support key exchange. The Kerberos system was designed at Massachusetts Institute of Technology [STE88, KOH93]. Kerberos is used for authentication between intelligent processes, such as client-to-server tasks, or a user's workstation to other hosts. Kerberos is based on the idea that a central server provides authenticated tokens, called tickets, to requesting applications.

A ticket is an unforgeable, nonrepayable, authenticated object. That is, it is an encrypted data structure naming a user and a service that user is allowed to obtain. It also contains a time value and some control information.

Here are the principal entities involved in the typical Kerberos workflow:

- Client. The client acts on behalf of the user and initiates communication for a service request

- Server. The server hosts the service the user wants to access

- Authentication Server (AS). The AS performs the desired client authentication. If the authentication happens successfully, the AS issues the client a ticket called TGT (Ticket Granting Ticket). This ticket assures the other servers that the client is authenticated

- Key Distribution Center (KDC). In a Kerberos environment, the authentication server logically separated into three parts: A database (db), the Authentication Server (AS), and the Ticket Granting Server (TGS). These three parts, in turn, exist in a single server called the Key Distribution Center

- Ticket Granting Server (TGS). The TGS is an application server that issues service tickets as a serviceirst, there are three crucial secret keys involved in the Kerberos flow. There are unique secret keys for the client/user, the TGS, and the server shared with the AS.

- Client/user. Hash derived from the user's password

- TGS secret key. Hash of the password employed in determining the TGS

- Server secret key. Hash of the password used to determine the server providing the service.

The protocol flow consists of the following steps:

Step 1: Initial client authentication request. The user asks for a Ticket Granting Ticket (TGT) from the authentication server (AS). This request includes the client ID.

Step 2: KDC verifies the client's credentials. The AS checks the database for the client and TGS's availability. If the AS finds both values, it generates a client/user secret key, employing the user's password hash.

The AS then computes the TGS secret key and creates a session key (SK1) encrypted by the client/user secret key. The AS then generates a TGT containing the client ID, client network address, timestamp, lifetime, and SK3. The TGS secret key then encrypts the ticket.

Step 3: The client decrypts the message. The client uses the client/user secret key to decrypt the message and extract the SK1 and TGT, generating the authenticator that validates the client's TGS.

Step 4: The client uses TGT to request access. The client requests a ticket from the server offering the service by sending the extracted TGT and the created authenticator to TGS.

Step 5: The KDC creates a ticket for the file server. The TGS then uses the TGS secret key to decrypt the TGT received from the client and extracts the SK3. The TGS decrypts the authenticator and checks to see if it matches the client ID and client network address. The TGS also uses the extracted timestamp to make sure the TGT hasn't expired.

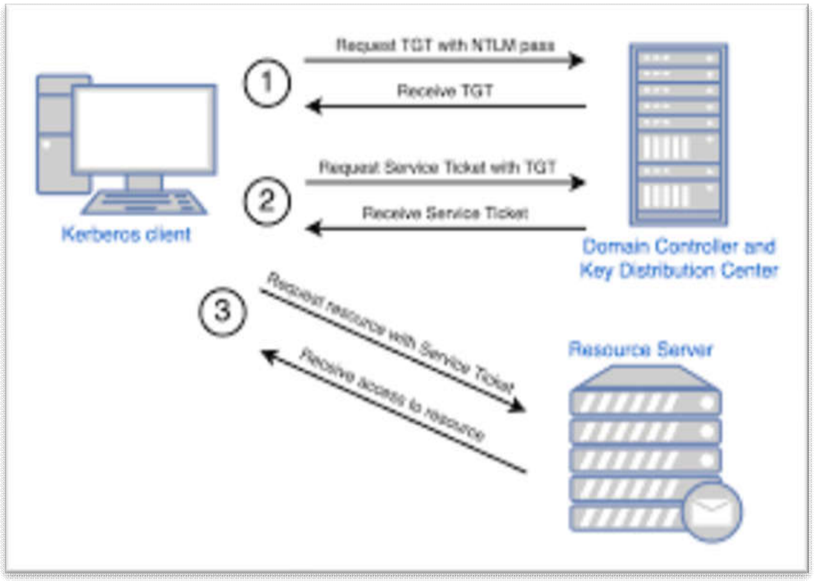
If the process conducts all the checks successfully, then the KDC generates a service session key (SK2) that is shared between the client and the target server.

Finally, the KDC creates a service ticket that includes the client id, client network address, timestamp, and SK2. This ticket is then encrypted with the server's secret key obtained from the db. The client receives a message containing the service ticket and the SK2, all encrypted with SK3.

Step 6: The client uses the file ticket to authenticate. The client decrypts the message using SK1 and extracts SK2. This process generates a new authenticator containing the client network address, client ID, and timestamp, encrypted with SK2, and sends it and the service ticket to the target server.

Step 7: The target server receives decryption and authentication. The target server uses the server's secret key to decrypt the service ticket and extract the SK2. The server uses SK2 to decrypt the authenticator, performing checks to make sure the client ID and client network address from the authenticator and the service ticket match. The server also checks the service ticket to see if it's expired.

Once the checks are met, the target server sends the client a message verifying that the client and the server have authenticated each other. The user can now engage in a secure session.


Request TGT with NTLM pass
Receive TGT
1
Request Service Ticket with TGT
Receive Service Ticket
2
Kerberos client
Domain Controller and Key Distribution Center
3
Request resource with Service Ticket
Receive access to resource
Resource Server

3.4 LET US SUM UP

Network security is any activity designed to protect the usability and integrity of your network and data.

- It includes both hardware and software technologies
- It targets a variety of threats
- It stops them from entering or spreading on your network
- Effective network security manages access to the network
- A firewall is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules.
- A firewall can be hardware, software, or both

3.5 LIST OF REFERENCES

A. http://www.ijmer.com/papers/Vol8_issue12/D0812012327.pdf

B. https://dokumen.tips/download/link/security-in-computing-4th-edition-2006#google_vignette

C. https://www.cisco.com/c/en_in/products/security/firewalls/what-is-a-firewall.html

3.6 SUMMARY

Network security helps in protecting the network and data from violation, intrusions and other threats. It is the most critical one because it prevents from cybercriminals from gaining the access to sensitive and valuable data, information.

3.7 BIBLIOGRAPHY

A. dokumen.tips_security-in-computing-4th-edition-2006.pdf

B. Network Security, Charlie Kaufman, Radia Perlam, Mike Speciner, Prentice Hall, 2nd Edition (2002)

C. Cryptography and Network Security 3rd edition, Atul Kahate, Tata McGraw Hill Education Private Limited (2013)

3.8 UNIT END EXERCISE

1. Number of phases in the handshaking protocol?
   a) 2

   b) 3

   c) 4

   d) 5

2. Which one of the following is not a higher –layer SSL protocol?
   a) Alert Protocol

   b) Handshake Protocol

   c) Alarm Protocol

   d) Change Cipher Spec Protocol

3. The full form of SSL is

   a) Serial Session Layer

   b) Secure Socket Layer

   c) Session Secure Layer

   d) Series Socket Layer

4. Which of the following is not a secure shell protocol?
   a) Transport Layer Protocol

   b) Secure Layer Protocol

   c) Connection Protocol

   d) User Authentication Protocol

5. Network layer firewall works as a __________
   a) Frame filter

   b) Packet filter

   c) Content filter

   d) Virus filter

6. A firewall is installed at the point where the secure internal network and untrusted external network meet which is also known as __________

   a) Chock point

   b) Meeting point

   c) Firewall point

   d) Secure point

7. What is one advantage of setting up a DMZ with two firewalls?
   a) You can control where traffic goes in three networks

   b) You can do stateful packet filtering

   c) You can do load balancing

   d) Improved network performance

8. What are the different ways to classify an IDS?
   a) Zone based

   b) Host & Network based

   c) Network & Zone based

   d) Level based

9. What are the characteristics of anomaly-based IDS?
   a) It models the normal usage of network as a noise characterization

   b) It doesn't detect novel attacks

   c) Anything distinct from the noise is not assumed to be intrusion activity

   d) It detects based on signature

10. What are the characteristics of signature-based IDS?
    a) Most are based on simple pattern matching algorithms

    b) It is programmed to interpret a certain series of packets

    c) It models the normal usage of network as a noise characterization

    d) Anything distinct from the noise is assumed to be intrusion activity

11. What is the major drawback of anomaly detection IDS?
    a) These are very slow at detection

    b) It generates many false alarms

    c) It doesn't detect novel attacks

    d) None of the mentioned

12. what are the drawbacks of signature-based IDS?

    a) They are unable to detect novel attacks

    b) They suffer from false alarms

    c) They have to be programmed again for every new pattern to be detected

    d) All of the mentioned

13. Public key encryption/decryption is not preferred because

    a) it is slow

    b) it is hardware/software intensive

    c) it has a high computational load

    d) all of the mentioned

14. ___________ ensures the integrity and security of data that are passing over a network.

    a) Firewall

    b) Antivirus

    c) Pen testing Tools

    d) Network-security protocols

15. TSL (Transport Layer Security) is a cryptographic protocol used for securing HTTP/HTTPS based connection.

    a) True

    b) False

❖❖❖❖

4

NETWORK SECURITY-II

Unit Structure :

4.0 OBJECTIVES

1. To understand the basics of web security
2. To understand Kerberos
3. To understand the Basic Protocols
4. To understand the need of Secure Electronic Transaction.

4.1 INTRODUCTION

Network security mainly refers to an evaluation which is taken by any enterprise or an organization to secure or to make safe its computer network. Its main role is to maintain confidentiality and accessibility of the data and network. In every enterprise or an organization which generally manages or handles a large amount of data needs some infusion aginst many cyber threats. The most common example of network security is to handle the password protection. Network security has become the main subject of cyber security. Network security deals with various levels which helps in performing the activities needed for handles the large amount of data in a secure manner.

4.2 WEB SERVER SECURITY

Web server security is the protection of information assets that can be accessed from a Web server. Web server security is important for any organization that has a physical or virtual Web server connected to the Internet.

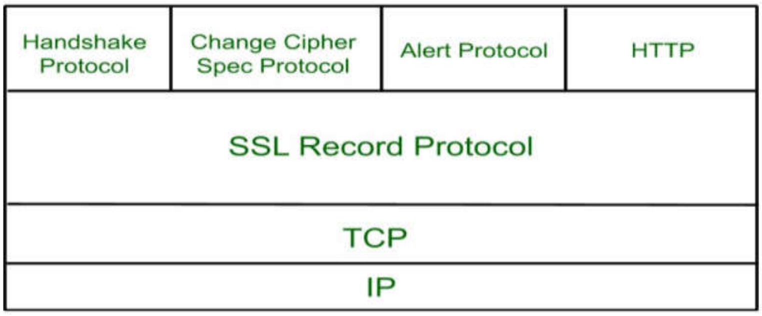
4.3 Ssl/Tls[2,3,4]

Secure Socket Layer (SSL) provides security to the data that is transferred between web browser and server. SSL encrypts the link between a web server and a browser which ensures that all data passed between them remain private and free from attack.

Secure Socket Layer Protocols:

- SSL record protocol
- Handshake protocol
- Change-cipher spec protocol
- Alert protocol

SSL Protocol Stack:

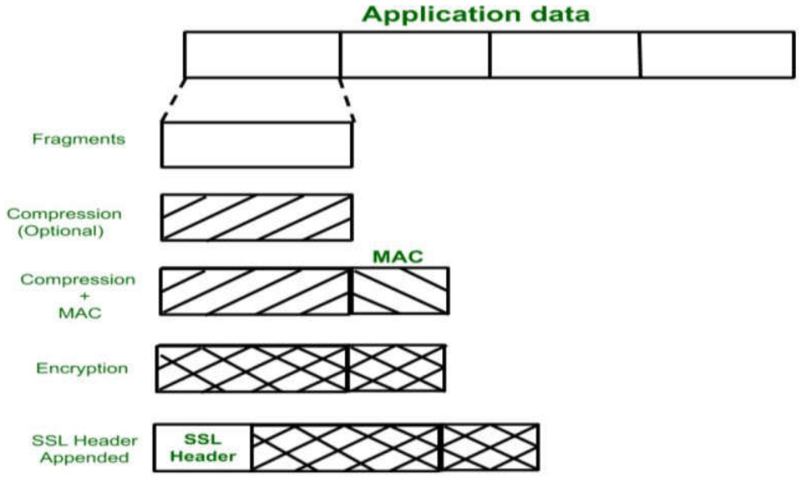


SSL Record Protocol:

SSL Record provides two services to SSL connection.

- Confidentiality
- Message Integrity

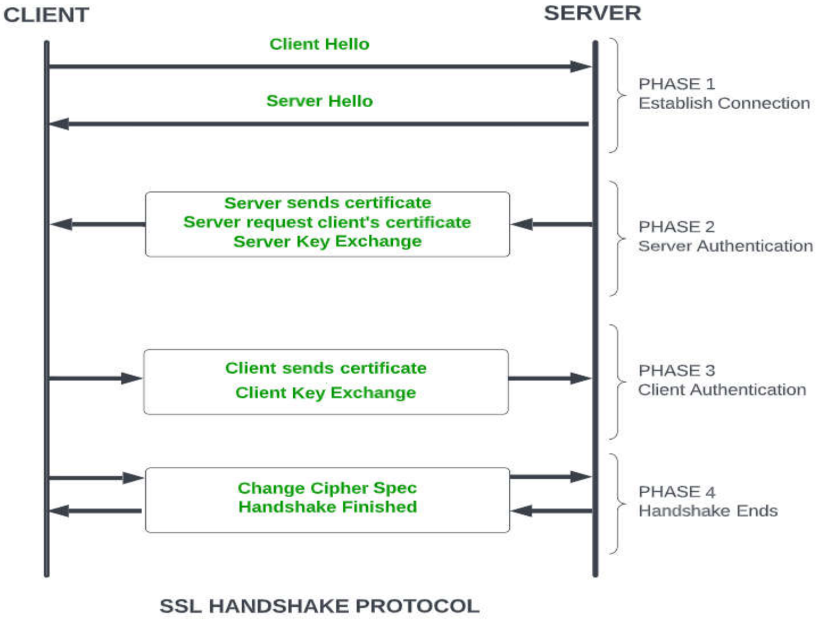
In the SSL Record Protocol application data is divided into fragments. The fragment is compressed and then encrypted MAC (Message Authentication Code) generated by algorithms like SHA (Secure Hash Protocol) and MD5 (Message Digest) is appended. After that encryption of the data is done and in last SSL header is appended to the data.



Handshake Protocol:

Handshake Protocol is used to establish sessions. This protocol allows the client and server to authenticate each other by sending a series of messages to each other. Handshake protocol uses four phases to complete its cycle.

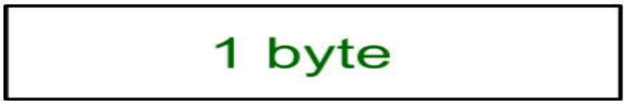
- **Phase-1:** In Phase-1 both Client and Server send hello-packets to each other. In this IP session, cipher suite and protocol version are exchanged for security purposes.
- **Phase-2:** Server sends his certificate and Server-key-exchange. The server end phase-2 by sending the Server-hello-end packet.
- **Phase-3:** In this phase, Client replies to the server by sending his certificate and Client-exchange-key.
- **Phase-4:** In Phase-4 Change-cipher suite occurred and after this Handshake Protocol ends.



SSL Handshake Protocol Phases diagrammatic representation

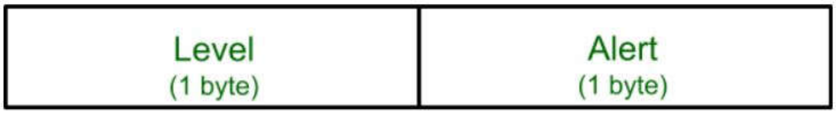
Change-cipher Protocol:

This protocol uses the SSL record protocol. Unless Handshake Protocol is completed, the SSL record Output will be in a pending state. After the handshake protocol, the Pending state is converted into the current state. Change-cipher protocol consists of a single message which is 1 byte in length and can have only one value. This protocol's purpose is to cause the pending state to be copied into the current state.



Alert Protocol:

This protocol is used to convey SSL-related alerts to the peer entity. Each message in this protocol contains 2 bytes.



The level is further classified into two parts:

Warning (level = 1):

This Alert has no impact on the connection between sender and receiver. Some of them are:

Bad certificate: When the received certificate is corrupt.

No certificate: When an appropriate certificate is not available.

Certificate expired: When a certificate has expired.

Certificate unknown: When some other unspecified issue arose in processing the certificate, rendering it unacceptable.

Close notify: It notifies that the sender will no longer send any messages in the connection.

Fatal Error (level = 2):

This Alert breaks the connection between sender and receiver. The connection will be stopped, cannot be resumed but can be restarted. Some of them are :

Handshake failure: When the sender is unable to negotiate an acceptable set of security parameters given the options available.

Decompression failure: When the decompression function receives improper input.

Illegal parameters: When a field is out of range or inconsistent with other fields.

Bad record MAC: When an incorrect MAC was received. **Unexpected message:** When an inappropriate message is received.

The second byte in the Alert protocol describes the error.

Silent Features of Secure Socket Layer:

- The advantage of this approach is that the service can be tailored to the specific needs of the given application.
- Secure Socket Layer was originated by Netscape.
- SSL is designed to make use of TCP to provide reliable end-to-end secure service.
- This is a two-layered protocol.

Versions of SSL:

SSL 1 – Never released due to high insecurity.

SSL 2 – Released in 1995.

SSL 3 – Released in 1996.

TLS 4.0 – Released in 1999.

TLS 4.1 – Released in 2006.

TLS 4.2 – Released in 2008.

TLS 4.3 – Released in 2018.

Transport Layer Securities

Transport Layer Securities (TLS) are designed to provide security at the transport layer. TLS was derived from a security protocol called Secure Socket Layer (SSL). TLS ensures that no third party may eavesdrop or tampers with any message.

There are several benefits of TLS:

Encryption:
TLS/SSL can help to secure transmitted data using encryption.

- **Interoperability:**

  TLS/SSL works with most web browsers, including Microsoft Internet Explorer and on most operating systems and web servers.

- **Algorithm flexibility:**

  TLS/SSL provides operations for authentication mechanism, encryption algorithms and hashing algorithm that are used during the secure session.

- **Ease of Deployment:**

  Many applications TLS/SSL temporarily on a windows server 2003 operating systems.

- **Ease of Use:**

  Because we implement TLS/SSL beneath the application layer, most of its operations are completely invisible to client.

Working of TLS:

The client connect to server (using TCP), the client will be something. The client sends number of specification:

1. Version of SSL/TLS.
2. which cipher suites, compression method it wants to use.

The server checks what the highest SSL/TLS version is that is supported by them both, picks a cipher suite from one of the clients option (if it supports one) and optionally picks a compression method. After this the basic setup is done, the server provides its certificate. This certificate must be trusted either by the client itself or a party that the client trusts. Having verified the certificate and being certain this server really is who he claims to be (and not a man in the middle), a key is exchanged. This can be a public key, “PreMasterSecret” or simply nothing depending upon cipher suite.

Both the server and client can now compute the key for symmetric encryption. The handshake is finished and the two hosts can communicate securely. To close a connection by finishing. TCP connection both sides will know the connection was improperly terminated. The connection cannot be compromised by this through, merely interrupted.

4.4 SSL/TLS BASIC PROTOCOL[2,3,4]

SSL stands for Secure Sockets Layer and was originally created by Netscape. SSLv2 and SSLv3 are the 2 versions of this protocol (SSLv1 was never publicly released). After SSLv3, SSL was renamed to TLS.

TLS stands for Transport Layer Security and started with TLSv4.0 which is an upgraded version of SSLv3.

Those protocols are standardized and described by RFCs.

OpenSSL provides an implementation for those protocols and is often used as the reference implementation for any new feature.

The goal of SSL was to provide secure communication using classical TCP sockets with very few changes in API usage of sockets to be able to leverage security on existing TCP socket code.

SSL/TLS is used in every browser worldwide to provide https (http secure) functionality.

4.5 COMPUTING THE KEYS - CLIENT AUTHENTICATION

Server Authentication

Server Certificate

This is **Public Key** Certified by a Certificate with Trust from the client. Trust from the client can be done automatically with Certificate Authority trust.

It is crucial that clients check the Server Certificate against the expected hostname Hostname_validation

No Authentication Aka Anonymous

Even if it look like is a strange idea, it is possible to select cipher suite that does not provide any server authentication but still provide confidentiality.

Selecting string cipher **aNULL** Manual:ciphers allows to select such cipher suite. Remark this is not same a **eNULL** that provides no confidentiality at all.

Anonymous Diffie_Hellman exchange (**DH**) and Anonymous Elliptic Curves Diffie Hellman Exchange (**ECDH**) methods provide this anonymous authentication.

Client Authentication

Client authentication is optional. In many cases the client does not authenticate at the ssl layer, but rather with the usage of protocols above ssl, for example with HTTP authentication methods.

Client Certificates

- **Certificate Request TLS v4.2**

Server can send a Certificate Request with digest algorithms and a list CA Distinguished names which will be used by the client to select the Client Certificate it will send.

- **Client Certificate TLS v4.2**

Client send its Client Certificate first then all intermediate Certificates, if any, up to the CA (optionally excluded).

- **CertificateVerify TLS v4.2**

The Client sends a Certificate Verify that is signed by the private key counterpart of its Client public key included in the Certificate with digest algorithm over whole handshake messages so far (excluding this one of course).

This proves that this client owns the private key that applies to this specific handshake and hence authenticates the client for this session.

Alternate Authentication Methods

Public Key Certificate

This is the most commonly used method. With X509 Certificates and Certficate Authorities.

It applies To **Server Certificate** or to **Client Certificate** authentication.

Depending on CipherSuite, for Server Public Key can be used to derive pre-master-key.

Pre-Shared Keys

TLS PSK Pre Shared Key

Kerberos

Password

TLS SRP : Secure Remote Password. Allows authentication with a password over TLS.

Supported by OpenSSL with version 4.0.4.

RFC5054

TLS SRP is negotiated with various ciphersuites, currently all use SHA to compute SRP.

With SRP trust is based on the fact that both parties should know the password (or Password Verifier) to complete the SRP Verify Handshake.

It is possible to use RSA or DSS additionaly to prove Server identity with Certificates.

4.6 PKI AS DEPLOYED BY SSL ATTACKS FIXED IN V3 - [1 - 4]

In the real world, when we want to make sure a service is honest and delivers us the promised goods we ask for some sort of “seal of trust”. That “seal of trust” usually comes in the form of a certificate signed by a trusted notary that vouches for the legitimacy and honesty of that provider. For that model to work, both the business and the consumer need to trust that notary.

The virtual world works in a similar fashion through an infrastructure called PKI. When we log onto a Web site, say a bank, the browser first requests the site provide it with a certificate guaranteeing that the site is what it appears to be. That certificate is “signed” by the digital equivalent of the notary, called the Certificate Authority (CA).

The certificate does not only guarantee the authenticity of the service, but also the confidentiality of communications between the user and the service. It does this by supplying the required components used to encrypt our transactions. Through the use of encryption, the protocol is able to guarantee the privacy and security of the transactions because it prevents some third party to eavesdrop or modify the transactions. The most popular method of encryption for these transactions is, generally speaking, SSL. When you see https:// (as opposed to just http://) in your browser, you know that SSL is in the works and your communication with the website is encrypted.

The increasing awareness to the confidentiality of our transactions (just recall how much media noise was made following the release of FireSheep) means that more companies are now deploying SSL. SSL is not restricted anymore only to our banking services. Take, Google for instance. At first, only the Gmail login page was encrypted. In time, the whole Gmail service supported encryption – by default. Google has now even added the search functionality to be accessed via https.

Threat #1: Attacks against PKI

It's easy to see the powerful role that the CA has in the PKI model. Since, at the base of this model is the underlying assumption that the CA is truthful, honest and legitimate. Consequently, a hacker who gains control on a CA can then use it to issue fraudulent certificates and then masquerade as any website.

Over the past year, attackers have repeatedly compromised various CA organizations. These include, DigiNotar, GlobalSign, Comodo and Digicert Malaysia. These attacks were a direct consequence of the commoditization of certificates, where smaller, less competent organizations have started to obtain a bigger share in the certificate authority market. As it stands now, any CA can issue a digital certificate for any application – without any required consent from application owner.

Last weekend, Trustwave published a blog entry which in itself shows how fragile is this system. As a CA, they had once issued a certificate specific to an unnamed private company which allowed the interception of all SSL communications within the company. As part of this admission, Trustwave also announced that they will not repeat this type of offering again.

Threat #2: The Theft of Issued Website Certificates

The problem here is that Web application certificates are not simply confined to being stored by the application. While SSL prevents access to traffic by attackers, it has no built-in mechanisms that allow restrictive access to it by collaborative third parties. For example, proxies, load balancers, content delivery networks (CDNs) need to access the certificate's private key in order to access application data. Also data loss prevention and Web application firewall solutions require similar key access. As a result, the digital certificate is now stored in many locations - some residing outside of the site's physical environment and out of the application's owner control. These open up additional attack points which imply a higher success rate for attackers.

Threat #3: The Theft of Issued Code Signing Certificates

The problem is not only with online services. Also applications present certificates that attest to their legitimacy before performing sensitive operations. Therefore, code signing certificates are too a prime target for malware distributers. We've already witnessed this in the wild – Stuxnet for instance used a stolen certificate. More recently, a malware strain used a stolen certificate belonging to the Malaysian government.

Threat #4: Denial of Service attacks

Because of the encryption component – there is a heavy computational burden incurred when initiating the SSL communication. Therefore, SSL-protected resources are prime candidates for effective Denial of Service (DoS) attacks. Together with an increased consumption of computer resources per session, a multitude of simple attacks can be devised very efficiently.

4.7 SECURE ELECTRONIC TRANSACTION (SET)

Secure Electronic Transaction or SET is a system that ensures the security and integrity of electronic transactions done using credit cards in a scenario. SET is not some system that enables payment but it is a security protocol applied to those payments. It uses different encryption and hashing techniques to secure payments over the internet done through credit cards. The SET protocol was supported in development by major organizations like Visa, Mastercard, Microsoft which provided its Secure Transaction Technology (STT), and Netscape which provided the technology of Secure Socket Layer (SSL).

SET protocol restricts the revealing of credit card details to merchants thus keeping hackers and thieves at bay. The SET protocol includes Certification Authorities for making use of standard Digital Certificates like X.509 Certificate.

Requirements in SET :

The SET protocol has some requirements to meet, some of the important requirements are :

- It has to provide mutual authentication i.e., customer (or cardholder) authentication by confirming if the customer is an intended user or not, and merchant authentication.
- It has to keep the PI (Payment Information) and OI (Order Information) confidential by appropriate encryptions.

- It has to be resistive against message modifications i.e., no changes should be allowed in the content being transmitted.
- SET also needs to provide interoperability and make use of the best security mechanisms.

Participants in SET :

In the general scenario of online transactions, SET includes similar participants:

1. **Cardholder –** customer
2. **Issuer –** customer financial institution
3. **Merchant**
4. **Acquirer –** Merchant financial
5. **Certificate authority –** Authority that follows certain standards and issues certificates(like X.509V3) to all other participants.

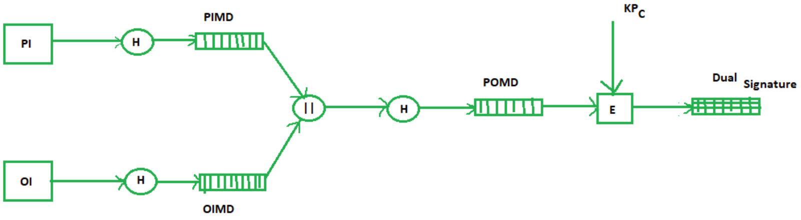
SET functionalities :

- **Provide Authentication**
- **Merchant Authentication** – To prevent theft, SET allows customers to check previous relationships between merchants and financial institutions. Standard X.509V3 certificates are used for this verification.
- **Customer / Cardholder Authentication** – SET checks if the use of a credit card is done by an authorized user or not using X.509V3 certificates.
- **Provide Message Confidentiality**: Confidentiality refers to preventing unintended people from reading the message being transferred. SET implements confidentiality by using encryption techniques. Traditionally DES is used for encryption purposes.
- **Provide Message Integrity**: SET doesn't allow message modification with the help of signatures. Messages are protected against unauthorized modification using RSA digital signatures with SHA-1 and some using HMAC with SHA-1,

Dual Signature :

The dual signature is a concept introduced with SET, which aims at connecting two information pieces meant for two different receivers : **Order Information (OI) for merchant Payment Information (PI) for bank**

You might think sending them separately is an easy and more secure way, but sending them in a connected form resolves any future dispute possible. Here is the generation of dual signature:



Where,

PI stands for payment information

OI stands for order information

PIMD stands for Payment Information Message Digest

OIMD stands for Order Information Message Digest

POMD stands for Payment Order Message Digest

H stands for Hashing

E stands for public key encryption

KPc is customer's private key

|| stands for append operation

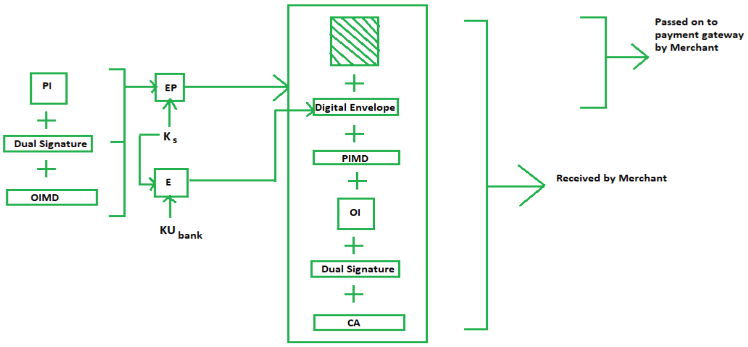
Dual signature, DS= E(KPc, [H(H(PI)||H(OI))])

Purchase Request Generation :

The process of purchase request generation requires three inputs:

- Payment Information (PI)
- Dual Signature
- Order Information Message Digest (OIMD)

The purchase request is generated as follows:



Here,

PI, OIMD, OI all have the same meanings as before.

The new things are :

EP which is symmetric key encryption

Ks is a temporary symmetric key

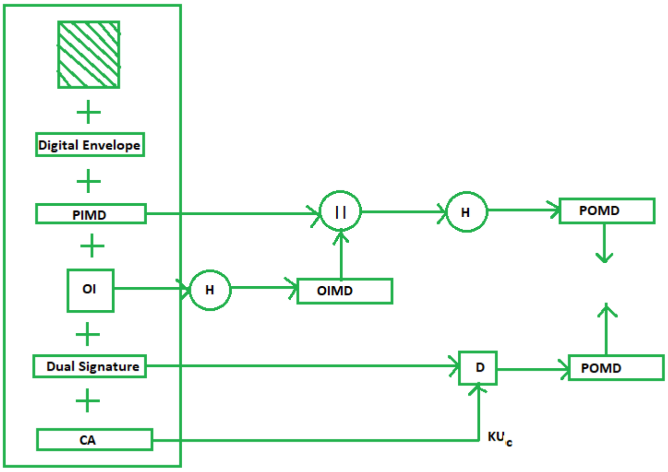
KUbank is public key of bank

CA is Cardholder or customer Certificate

Digital Envelope = E(KUbank, Ks)

Purchase Request Validation on Merchant Side :

The Merchant verifies by comparing POMD generated through PIMD hashing with POMD generated through decryption of Dual Signature as follows:



Since we used Customer's private key in encryption here we use KUC which is the public key of the customer or cardholder for decryption 'D'.

Payment Authorization and Payment Capture : Payment authorization as the name suggests is the authorization of payment information by the merchant which ensures payment will be received by the merchant. Payment capture is the process by which a merchant receives payment which includes again generating some request blocks to gateway and payment gateway in turn issues payment to the merchant.

4.8 KERBEROS[5]

Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users. In Kerberos Authentication server and database is used for client authentication. Kerberos runs as a third-party trusted server known as the Key Distribution Center (KDC). Each user and service on the network is a principal.

The main components of Kerberos are:

- **Authentication Server (AS):**

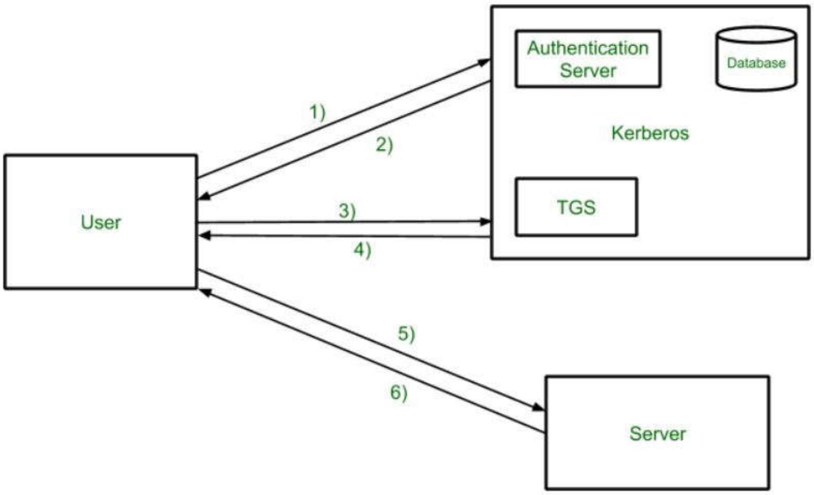
  The Authentication Server performs the initial authentication and ticket for Ticket Granting Service.

- **Database:**
  The Authentication Server verifies the access rights of users in the database.

- **Ticket Granting Server (TGS):**

  The Ticket Granting Server issues the ticket for the Server

Kerberos Overview:



- **Step-1:**
  User login and request services on the host. Thus user requests for ticket-granting service.

- **Step-2:**
  Authentication Server verifies user’s access right using database and then gives ticket-granting-ticket and session key. Results are encrypted using the Password of the user.

- **Step-3:**
  The decryption of the message is done using the password then send the ticket to Ticket Granting Server. The Ticket contains authenticators like user names and network addresses.

- **Step-4:**
  Ticket Granting Server decrypts the ticket sent by User and authenticator verifies the request then creates the ticket for requesting services from the Server.

- **Step-5:**
  The user sends the Ticket and Authenticator to the Server.

- **Step-6:**
  The server verifies the Ticket and authenticators then generate access to the service. After this User can access the services.

Kerberos Limitations

- Each network service must be modified individually for use with Kerberos
- It doesn't work well in a timeshare environment
- Secured Kerberos Server
- Requires an always-on Kerberos server
- Stores all passwords are encrypted with a single key
- Assumes workstations are secure
- May result in cascading loss of trust.
- Scalability

4.9 SUMMARY

Protection of this accessible information assets from a Web Server is known as Server Security. A security break can destructively affect the goodwill as well as the economic status of an organization. Web server security becomes highly important when it is connected to the internet.

4.10 REFERENCES FOR READING

1. Examining Threats Facing Public Key Infrastructu https://www.securityweek.com/examining-threats-facing-public-key-infrastructure-pki-and-secure-socket-layer-sslre (PKI) and Secure Socket Layer (SSL) | SecurityWeek.Com
2. https://www.geeksforgeeks.org/secure-socket-layer-ssl/
3. https://www.geeksforgeeks.org/secure-electronic-transaction-set-protocol/?ref=lbp
4. https://www.geeksforgeeks.org/kerberos/

❖❖❖❖

5

CLOUD SECURITY

Unit Structure:

5.0 OBJECTIVE

Three key cyber security objectives: ensuring confidentiality, integrity and availability of information resources and systems are high-priority concerns and potential risks to cloud technology. The implementation of cloud technology is subject to local physical threats as well as remote, external threats. Consistent with other IT applications, threat sources include accidents, natural disasters, and external loss of service, hostile governments, criminal organizations, terrorist groups, intentional and unintentional vulnerabilities through internal/external authorized and unauthorized system access, including but not limited to employees, contractors, vendors and intruders. The characteristics of cloud technology, specifically multi-tenancy and implications of three service and four deployment models, heighten the efforts to protect Postal Service data and systems, as well as physical boundaries.

5.1 INTRODUCTION

Cybersecurity is the protection of computer systems and networks from information disclosure, theft of or damage to their hardware, software, or electronic data, as well as from the disruption or misdirection of the services they provide.

The field is becoming increasingly significant due to the increased reliance on computer systems, the Internetand wireless network standards such as Bluetooth and Wi-Fi, and due to the growth of "smart" devices, including smartphones, televisions, and the various devices that constitute the "Internet of things". Owing to its complexity, both in terms of politics and technology, cybersecurity is also one of the major challenges in the contemporary world.

5.2 HOW CONCEPTS OF SECURITY APPLY IN THE CLOUD?

5.2.1 User authentication in the cloud

Cloud computing provides customers with highly scalable and on-mend computing resources. NIST specified three cloud service models:

Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructural as a Service (IaaS), each service models target a specific need of customers.

- Software as a Service offers applications that were provided by the cloud service providers and hosted by the cloud provider.
- Platform as a Service offers hosting environment for developers to develop and publish their applications.
- Infrastructural as a Service offers visualised computing resources such as virtual desktop, virtual storage, etc. Various cloud services and cloud service providers are beneficial for customers who seek specific computing resource, it creates some security challenges to the customers seeking different cloud services however.

1. Cloud service providers request customers to store their account information in the cloud and they have the access to this information. This presents a privacy issue to the customer’s privacy information.
2. Many SLAs have specified the privacy of the sensitive information. It is difficult for customers to make sure the proper rules are enforced.

There is a lack of transparency in the cloud that allows the customers to monitor their own privacy information.

3. When a customer decides to use multiple cloud service, the customer will have to store the password in multiple cloud. As the user takes cloud subscription of any cloud service that much number of copies of the users information are created. This is a security issue for the customers and the cloud service providers.

4. The multiple copies of account will lead to multiple authentication processes. For every cloud service, the customer needs to exchange their authentication information.

5. Cloud service providers use different authentication technologies for authenticating users, this may have less impact on SaaS than PaaS and IaaS, but it is present a challenge to the customers. The key concept to user authentication is that a user who established an identity by connection with cloud computing can use the same identity with other clouds also.

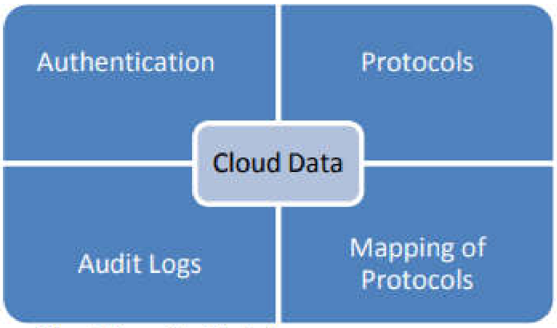
6. As users communicate with the Cloud, identity becomes an important issue to maintain security, visibility and control. In this distributed environment, it is essential for applications to authenticate the user's identity, understand what that user is authorized to do, create or update an account and audit their activities. Thus authentication and authorization are critical components of a cloud identity strategy and provide portability and extensibility beyond enterprise boundaries.

Authentication

Authentication is the process for confirming the identity of the user. The traditional authentication process allows the system to identify the user through a username and then validate their identity through password. There are even stronger methods of user authentication such as x.509 certificates, one-time passwords (OTP), and device fingerprinting. These can be combined to provide a stronger combination of authentication factors. Federated identity allows a user to access an application in one domain, such as a Software-as-asService (SaaS) application, using the authentication that occurred in another domain, such as a corporate Identity Management (IdM) system.

5.2.2 Techniques used in user Authentication

Identity and Access Control Service should provide identity management and access control to cloud resources for registered entities. Such entities can be people, software processes or other systems. In order to give a proper level of access to a resource, the identity of an entity should be verified first, which is the authentication process that precedes the authorization process. Besides authentication and authorization processes, audit logging mechanism should be used to keep track of all successful and failed operations regarding authentication and access attempts by the application. Confidentiality is achieved by different encryption mechanisms, which is the procedure of encoding data by means of cryptographic algorithms. Providing such a service will guarantee privacy of sensitive and private data and the intended entity can only decode it. Cryptographic algorithms, which are computationally hard to crack together with encryption and decryption procedures, digital signatures, hashing, certificates, key exchange and management form an encryption system which can be delivered as a service and assure confidentiality and non-repudiation in a cloud environment.



Cloud Security Model

5.2.3 Algorithms For User Authentication

The central idea behind the Security provision is to avoid the unwanted intrusion of unauthorized users and right at the entry point. That is all the users whether new of existing are not allowed to access the data or resources without proving their identity. The request from the users are first encrypted and then sent to the cloud files. The algorithms used to encryption process are discussed as follows:

1. **RSA Algorithm**: RSA encryption algorithm is used for making the communication safe. Usually the users' requests are encrypted while sending to the cloud service provider system. RSA algorithm using the system's public key is used for the encryption. Whenever the user requests for a file the system sends it by encrypting it via RSA encryption algorithm using the user’s public key. Same process is also applied about the user password requests, while logging in the system later. After receiving an encrypted file from the system the user’s browser will decrypt it with RSA algorithm using the user’s private key. Similarly when the system receives an encrypted file from the user it will immediately decrypt it using its private key. As a result the communication becomes secured between the user and the system.

2. **AES Algorithm & MD5 Hashing Algorithm**: When a file is uploaded by an user the system server encrypts the file using AES encryption algorithm. In this 128, 192, 256 bit key can be used. The key is generated randomly by the system server. A single key is used only once. That particular key is used for encrypting and decrypting a file of a user for that instance. This key is not further used in any instance later. The key is kept in the database table of the system server along with the user account name. Before inserting the user account name it is also hashed using md5 hashing. This insures that

unauthorized person cannot retrieve the key to decrypt a particular file for a particular user by simply gaining access and observing the database table of the system server. As a result the key for a particular file becomes hidden and safe. Again when the encrypted file is uploaded for storing to the storage server, the path of the encrypted file along with the user account is kept and maintained in the database table on the storage server. Here user name is used for synchronization between the database tables of main system server and the storage server. The encrypted files on the storage server are inserted not serially.

3. **OTP Password Algorithm**: In this algorithm one time password has been used for authenticating the user. The password is used to keep the user account secure and secret from the unauthorized user. But the user defined password can be compromised. To overcome this difficulty one time password is used in the proposed security model. Thus whenever a user logs in the system, he will be provided with a new password for using it in the next login. This is usually provided by the system itself. This password will be generated randomly. Each time a new password is created for a user, the previous password for that user will be erased from the system. New password will be updated for that particular user. A single password will be used for login only once. The password will be sent to the users authorized mail account. Therefore at a same time a check to determine the validity of the user is also performed. As a result only authorized user with a valid mail account will be able to connect to the cloud system.

4. **Data Encryption Standard Algorithm**: Data Encryption Standard algorithm is a type of symmetric-key encipherment algorithms. Symmetric-key encryption is a type of cryptosystem in which encryption and decryption are performed using a single (secret) key. As we can see, secret key play a very important role in DES security, so that a good key generation unit required. Using Dynamic key generator, the generated key has characteristics of unpredictability and unrepeatability. Using this approach the dynamic key generator can achieve the high speed and can be reduce logic complexity.

5. **Rijndael encryption Algorithm**: Rijndael is the standard symmetric key encryption algorithm to be used to encrypt sensitive information. Rijndael is an iterated block cipher, the encryption or decryption of a block of data is accomplished by the iteration (a round) of a specific transformation (a round function). As input, Rijndael accepts one-dimensional 8-bit byte arrays that create data blocks. The plaintext is input and then mapped onto state bytes. The cipher key is also a one-dimensional 8-bit byte array. With an iterated block cipher, the different transformations operate in sequence on intermediate cipher results (states).

5.2.4 Protocols Used In The Process of User Authentication

Identity and Access Control Service should provide identity management and access control to cloud resources for registered entities. Such entities can be people, software processes or other systems. In order to give a proper level of access to a resource, the identity of an entity should be verified first, which is the authentication process that precedes the authorization process. Besides authentication and authorization processes, audit logging mechanism should be used to keep track of all successful and failed operations regarding authentication and access attempts by the application. Confidentiality is achieved by different encryption mechanisms, which is the procedure of encoding data by means of cryptographic algorithms. Providing such a service will guarantee privacy of sensitive and private data and the intended entity can only decode it. Cryptographic algorithms, which are computationally hard to crack together with encryption and decryption procedures, digital signatures, hashing, certificates, key exchange and management form anencryption system which can be delivered as a service and assure confidentiality and non-repudiation in a cloud environment.

Authentication Protocols used are as follows:

1. **Extensible Authentication Protocol-CHAP**: EAP(Extensible Authentication Protocol) will implement on Cloud environment for authentication purpose. It is used for the transport and usage of keying material and parameters generated by EAP methods. In our purposed model we use Challenge-Handshake Authentication Protocol (CHAP) for authentication.

2. **Lightweight Directory Access Protocol**: Most companies are storing their important information in some type of Lightweight Directory Access Protocol server. SaaS providers can provide delegate the authentication process to the customer's internal LDAP/AD server, so that companies can retain control over the management of users.

3. **Single Sign-on (SSO) protocol**: This protocol is part of the shared security system of a cloud environment. The system consists of a SAML server which provides SSO services for application service providers: SAML server issues SAML ticket which contains an assertion about the client's identity verification, thus confirming that it has been properly authenticated or not. Once the user is authenticated, he or she can request access to different authorized resources at different application provider sites without the need to reauthenticate for each domain.

5.3 VIRTUALIZATION SYSTEM

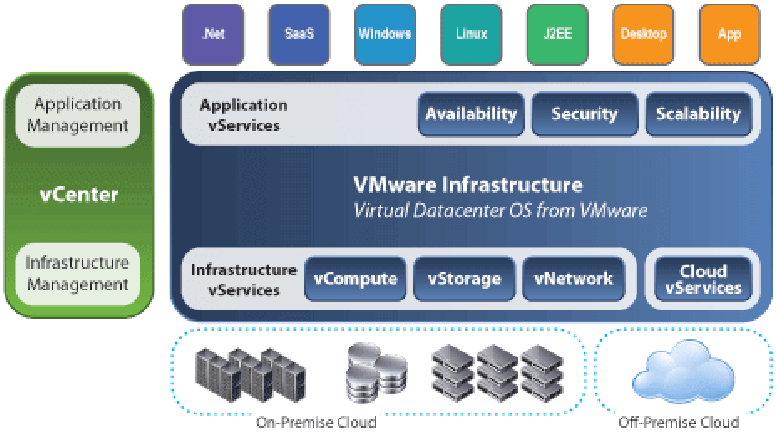
Virtualization is the "creation of a virtual (rather than actual) version of something, such as a server, a desktop, a storage device, an operating system or network resources".

In other words, Virtualization is a technique, which allows to share a single physical instance of a resource or an application among multiple customers and organizations. It does by assigning a logical name to a physical storage and providing a pointer to that physical resource when demanded.

Virtualization plays a very important role in the cloud computing technology, normally in the cloud computing, users share the data present in the clouds like application etc, but actually with the help of virtualization users shares the Infrastructure.

The **main usage of Virtualization Technology** is to provide the applications with the standard versions to their cloud users, suppose if the next version of that application is released, then cloud provider has to provide the latest version to their cloud users and practically it is possible because it is more expensive.

To overcome this problem we use basically virtualization technology, By using virtualization, all severs and the software application which are required by other cloud providers are maintained by the third party people, and the cloud providers has to pay the money on monthly or annual basis.



Creation of a virtual machine over existing operating system and hardware is known as Hardware Virtualization. A Virtual machine provides an environment that is logically separated from the underlying hardware.

The machine on which the virtual machine is going to create is known as **Host Machine** and that virtual machine is referred as a **Guest Machine**

5.3.1 Virtualization System Security Issues

Bad storage, server, and network configurations are just a few reasons why virtualization fails. These are technical in nature and are often easy to fix, but some organizations overlook the need to protect their entire virtualized environments, thinking that they're inherently more secure than traditional IT environments. Others use the same tools they use to protect their existing physical infrastructure. The bottom line is that a virtualized environment is more complex and requires a new management approach. These are the common problems talked about behind closed doors.

1.Resource distribution

The way virtualization partitions systems can result in varied ways — some might function really well, and others might not provide users access to enough resources to meet their needs. Resource distribution problems often occur in the shift to virtualization and can be fixed by working on capacity planning with your service provider.

2. VM Sprawl

VM sprawl, the unchecked growth of virtual machines in a virtual environment, as any virtualization admin knows, can cripple an otherwise healthy environment. It is problematic because its underlying cause often stays hidden until it manifests in resource shortages. You should look at how virtual machines will be managed, who will be doing what, and what systems you're going to use. One of the optimal times to develop an overall management plan is when you're in a testing phase, before migration.

3.Backward compatibility

Using legacy systems can cause problems with newer virtualized software programs. Compatibility issues can be time-consuming and difficult to solve. A good provider may be able to suggest upgrades and workarounds to ensure that everything functions the way they should.

4. Performance monitoring

Virtualized systems don't lend themselves to the same kind of performance monitoring as hardware like mainframes and hardware drives do. Try tools like VMmark to create benchmarks that measure performance on virtual networks and to monitor resource usage as well.

5.Backup

In a virtualized environment, there is no actual hard drive on which data and systems can be backed up. This means frequent software updates can make it difficult to access backup at times. Software programs like Windows Server Backup tools can make this process easier and allow backups to be stored in one place for easier tracking and access.

5.Security

Virtual systems could be vulnerable when users don't keep them secure and apply best practices for passwords or downloads. Security then becomes a problem for virtualization, but the isolation of each VM by the system can mitigate security risks and prevent systems from getting breached or compromised.

5.3.2 ESX and ESXi Security

According to the latest statistics, VMware holds more than 75% of the global server virtualization market, which makes the company the undisputed leader in the field, with its competitors lagging far behind. VMware hypervisor provides you with a way to virtualize even the most resource-intensive applications while still staying within your budget. If you are just getting started with VMware software, you may have come across the seemingly unending ESX vs. ESXi discussion. These are two types of VMware hypervisor architecture, designed for “bare-metal” installation, which is directly on top of the physical server (without running an operating system

What Does ESXi Stand for and How Did It All Begin?

If you are already somewhat familiar with the VMware product line, you may have heard that ESXi, unlike ESX, is available free of cost. This has led to the common misconception that ESX servers provide a more efficient and feature-rich solution, compared to ESXi servers. This notion, however, is not entirely accurate.

ESX is the predecessor of ESXi. The last VMware release to include both ESX and ESXi hypervisor architectures is vSphere 4.1 (“vSphere”). Upon its release in August 2010, ESXi became the replacement for ESX. VMware announced the transition away from ESX, its classic hypervisor architecture, to ESXi, a more lightweight solution.

The primary difference between ESX and ESXi is that ESX is based on a Linux-based console OS, while ESXi offers a menu for server configuration and operates independently from any general-purpose OS. For your reference, the name ESX is an abbreviation of Elastic Sky X, while the newly-added letter “i” in ESXi stands for “integrated.” As an aside, you may be interested to know that at the early development stage in 2004, ESXi was internally known as “VMvisor” (“VMware Hypervisor”), and became “ESXi” only three years later. Since version 5, released in July 2011, only ESXi has continued.

ESX vs. ESXi: Key Differences

Overall, the functionality of ESX and ESXi hypervisors is effectively the same. The key difference lies in architecture and operations management. If only to shorten the VMware version comparison to a few words, ESXi architecture is superior in terms of security, reliability, and management. Additionally, as mentioned above, ESXi is not dependent on an operating system. VMware strongly recommends their users currently running the classic ESX architecture to migrate to ESXi. According to VMware documentation, this migration is required for users to upgrade beyond the 4.1 version and maximize the benefits from their hypervisor.

To protect an ESXi host against an unauthorized intrusion and misuse, VMware imposes constraints on several parameters, settings, and activities. You can loosen the constraints to meet your configuration needs. If you do, make sure that you are working in a trusted environment and take other security measures.

Built-In Security Features

Risks to the hosts are mitigated as follows:

- ESXi Shell and SSH interfaces are disabled by default. Keep these interfaces disabled unless you are performing troubleshooting or support activities. For day-to-day activities, use the vSphere Client, where activity is subject to role-based access control and modern access control methods.

- Only a limited number of firewall ports are open by default. You can explicitly open additional firewall ports that are associated with specific services.

- ESXi runs only services that are essential to managing its functions. The distribution is limited to the features required to run ESXi.

- By default, all ports that are not required for management access to the host are closed. Open ports if you need additional services.

- By default, weak ciphers are disabled and communications from clients are secured by SSL. The exact algorithms used for securing the channel depend on the SSL handshake. Default certificates created on ESXi use PKCS#1 SHA-256 with RSA encryption as the signature algorithm.

- An internal web service is used by ESXi to support access by Web clients. The service has been modified to run only functions that a Web client requires for administration and monitoring. As a result, ESXi is not vulnerable to web service security issues reported in broader use.

- VMware monitors all security alerts that can affect ESXi security and issues a security patch if needed. You can subscribe to the VMware Security Advisories and Security Alerts mailing list to receive security alerts

- Insecure services such as FTP and Telnet are not installed, and the ports for these services are closed by default.

- To protect hosts from loading drivers and applications that are not cryptographically signed, use UEFI Secure boot. Enabling Secure Boot is done at the system BIOS. No additional configuration changes are required on the ESXi host, for example, to disk partitions.

- If your ESXi host has a TPM 2.0 chip, enable and configure the chip in the system BIOS. Working together with Secure Boot, TPM 2.0 provides enhanced security and trust assurance rooted in hardware.

Additional Security Measures

Consider the following recommendations when evaluating host security and administration.

Limit access

If you enable access to the Direct Console User Interface (DCUI), the ESXi Shell, or SSH, enforce strict access security policies.

The ESXi Shell has privileged access to certain parts of the host. Provide only trusted users with ESXi Shell login access.

Do not access managed hosts directly

Use the vSphere Client to administer ESXi hosts that are managed by a vCenter Server. Do not access managed hosts directly with the VMware Host Client, and do not change managed hosts from the DCUI.

If you manage hosts with a scripting interface or API, do not target the host directly. Instead, target the vCenter Server system that manages the host and specify the host name.

Use DCUI only for troubleshooting

Access the host from the DCUI or the ESXi Shell as the root user only for troubleshooting. To administer your ESXi hosts, use one of the GUI clients, or one of the VMware CLIs or APIs. See ESXCLI Concepts and

Use only VMware sources to upgrade ESXi components

The host runs several third-party packages to support management interfaces or tasks that you must perform. VMware only supports upgrades to these packages that come from a VMware source. If you use a download or patch from another source, you might compromise management interface security or functions. Check third-party vendor sites and the VMware knowledge base for security alerts.

5.3.3 ESX file system security- storage considerations, backup and recovery

VMware developed its own high performance cluster file system called VMware Virtual Machine File System or VMFS. VMFS provides a file system which has been optimized for storage virtualization for virtual machines through the use of distributed locking. A virtual machine stored on a VMFS partition always appears to the virtual machine as a mounted SCSI disk. The virtual disk or *.vmdk file hides the physical storage layer from the virtual machine's operating system. VMFS versions 1 and 2 were flat file systems, and typically only housed .vmdk files. The VMFS 3 file system now allows for a directory structure. As a result, VMFS 3 file systems can contain all of the configuration and disk files for a given virtual machine. The VMFS file system is one of the things that set VMware so far ahead of its competitors. Conventional file systems will allow one server to have a read/ write access or lock to a given file at any given time. VMware's VMFS is a file system which will allow multiple nodes or multiple VMware ESX servers to read and write to the same LUN or VMFS partition concurrently.

Now that we know about VMFS, let's take a look at the different storage options that are made available.

Direct-attached storage

Direct-attached storage (DAS) is storage that is, as the name implies, directly attached to a computer or server. DAS is usually the first step taken when working with storage. A good example would be a company with two VMware ESX Servers directly attached to a disk array. This configuration is a good starting point, but it typically doesn't scale very well.

Network-attached storage

Network-attached storage (NAS) is a type of storage that is shared over the network at a file system level. This option is considered an entry-level or low-cost option with a moderate performance rating. VMware ESX will connect over the network to a specialized storage device. This device can be in the form of an appliance or a computer that uses Network File System (NFS).

The VMkernel is used to connect to a NAS device via the VMkernel port and supports NFS Version 3 carried over TCP/IP only. From the standpoint of the VMware ESX servers, the NFS volumes are treated the same way VMware ESX would treat iSCSI or Fibre Channel storage. You are able to VMotion guests from one host to the next, create virtual machines, boot virtual machines as well as mount ISO images as CD-ROMs when presented to the virtual machines.

When configuring access to standard Unix/Linux-based NFS devices, some configuration changes will need to be defined. The directory /etc/exports will define the systems that are allowed to access the shared directory. And there are a few options in this file that you should be aware of.

1. Name the directory to be shared.

2. Define the subnets that will be allowed access to the share.

3. Allow both "read" and "write" permissions to the volume.

4. no_root_squash -- The root user (UID = 0) by default is given the least amount of access to the volume. This option will turn off this behavior, giving the VMkernel the access it needs to connect as UID 0.

5. sync -- All file writes MUST be committed to the disk before the client write request is actually completed.

Windows Server 2003 R2 also natively provides NFS sharing when the Windows Services for Unix (SFU) service is installed and configured. Out of the box, Windows Server 2003 R2 has this ability, but it can also be run on Windows Server 2003 (non-R2), and Windows 2000 Server after downloading SFU from Microsoft's Website.

1. After storage has been allocated, the folders are presented similarly as NFS targets.
2. Because there is no common authentication method between VMware ESX and a Microsoft Windows server, the /etc/passwd file must be copied to the Windows server, and mappings must be made to tie an account on the ESX server to a Windows account with appropriate access rights.

5.3.4 Virtualization System Vulnerabilities

Virtualization has eased many aspects of IT management but has also complicated the task of cyber security.The nature of virtualization introduces a new threat matrix, and administrators need to address the resulting vulnerabilities in their enterprise environments.

Critical Virtualization Vulnerabilities

Some attacks against virtual machine, or VM, environments are variations of common threats such as denial of service. Others are still largely theoretical but likely approaching as buzz and means increase. Keep an eye on these critical weaknesses:

1. **VM sprawl:** VMs are easy to deploy, and many organizations view them as hardware-like tools that don’t merit formal policies.This has led to VM sprawl, which is the unplanned proliferation of VMs.Attackers can take advantage of poorly monitored resources.More deployments also mean more failure points, so sprawl can cause problems even if no malice is involved.

2. **Hyperjacking:** Hyperjacking takes control of the hypervisor to gain access to the VMs and their data. It is typically launched against type 2 hypervisors that run over a host OS although type 1 attacks are theoretically possible. In reality, hyperjackings are rare due to the difficulty of directly accessing hypervisors.However, hyperjacking is considered a real-world threat, and administrators should take the offensive and plan for it.

3. **VM escape:** A guest OS escapes from its VM encapsulation to interact directly with the hypervisor.This gives the attacker access to all VMs and, if guest privileges are high enough, the host machine as well. Although few if any instances are known, experts consider VM escape to be the most serious threat to VM security.

4. **Denial of service:** These attacks exploit many hypervisor platforms and range from flooding a network with traffic to sophisticated leveraging of a host’s own resources.The availability of botnets continues to make it easier for attackers to carry out campaigns against

specific servers and applications with the goal of derailing the target's online services.

5. **Incorrect VM isolation:** To remain secure and correctly share resources,VMs must be isolated from each other.Poor control over VM deployments can lead to isolation breaches in which VMs communicate.Attackers can exploit this virtual drawbridge to gain access to multiple guests and possibly the host.

6. **Unsecured VM migration:** This occurs when a VM is migrated to a new host, and security policies and configuration are not updated to reflect the change.Potentially, the host and other guests could become more vulnerable.Attackers have an advantage in that administrators are likely unaware of having introduced weaknesses and will not be on alert.

7. **Host and guest vulnerabilities:** Host and guest interactions can magnify system vulnerabilities at several points.Their operating systems, particularly Windows, are likely to have multiple weaknesses.Like other systems, they are subject to vulnerabilities in email, Web browsing, and network protocols.However, virtual linkages and the co-hosting of different data sets make a serious attack on a virtual environment particularly damaging.

How to Mitigate Risk

Fortunately, security engineers can take several steps to minimize risk.The first task is to accurately characterize all deployed virtualization and any active security measures beyond built-in hypervisor controls on VMs.Security controls should be compared against industry standards to determine gaps.Coverage should include anti-virus, intrusion detection, and active vulnerability scanning.Additionally, consider these action steps:

VM traffic monitoring:The ability to monitor VM backbone network traffic is critical.Conventional methods will not detect VM traffic because it is controlled by internal soft switches.However, hypervisors have effective monitoring tools that should be enabled and tested.

Administrative control:Secure access can become compromised due to VM sprawl and other issues.

- Ensure that authentication procedures, identity management, and logging are ironclad.

- Customer security:Outside of the VM, make sure protection is in place for customer-facing interfaces such as websites.

- VM segregation:In addition to normal isolation, strengthen VM security through functional segregation.For example, consider creating separate security zones for desktops and servers.The goal is to minimize intersection points to the extent feasible.

5.4 SECURITY MANAGEMENT STANDARDS

5.4.1. SaaS

Software-as-a-Service (SaaS) is a software licensing model in which access to the software is provided on a subscription basis, with the software being located on external servers rather than on servers located in-house.

Software-as-a-Service is typically accessed through a web browser, with users logging into the system using a username and password. Instead of each user having to install the software on their computer, the user is able to access the program via the Internet.

- Software-as-a-Service (SaaS) is a software licensing model, which allows access to software a subscription basis using external servers.
- SaaS allows each user to access programs via the Internet, instead of having to install the software on the user's computer.
- SaaS has many business applications, including file sharing, email, calendars, customer retention management, and human resources.
- SaaS is easy to implement, easy to update and debug, and can be less expensive (or at least have lower up-front costs) since users pay for SaaS as they go instead of purchasing multiple software licenses for multiple computers.
- Drawbacks to the adoption of SaaS center around data security, speed of delivery, and lack of control.

Understanding Software-as-a-Service (SaaS)

The rise of Software-as-a-Service (SaaS) coincides with the rise of cloud-based computing. Cloud computing is the process of offering technology services through the Internet, which often includes data storage, networking, and servers. Before SaaS was available, companies looking to update the software on their computers had to purchase compact disks containing the updates and download them onto their systems.

For large organizations, updating software was a time-consuming endeavor. Over time, software updates became available for download through the Internet, with companies purchasing additional licenses rather than additional disks. However, a copy of the software still needed to be installed on all devices that needed access to it.

What Is Software-as-a-Service (SaaS)?

Software-as-a-Service (SaaS) is a software licensing model in which access to the software is provided on a subscription basis, with the software being located on external servers rather than on servers located in-house.

Software-as-a-Service is typically accessed through a web browser, with users logging into the system using a username and password. Instead of each user having to install the software on their computer, the user is able to access the program via the Internet.

KEY TAKEAWAYS

- Software-as-a-Service (SaaS) is a software licensing model, which allows access to software a subscription basis using external servers.
- SaaS allows each user to access programs via the Internet, instead of having to install the software on the user's computer.
- SaaS has many business applications, including file sharing, email, calendars, customer retention management, and human resources.
- SaaS is easy to implement, easy to update and debug, and can be less expensive (or at least have lower up-front costs) since users pay for SaaS as they go instead of purchasing multiple software licenses for multiple computers.
- Drawbacks to the adoption of SaaS center around data security, speed of delivery, and lack of control.

Understanding Software-as-a-Service (SaaS)

The rise of Software-as-a-Service (SaaS) coincides with the rise of cloud-based computing. Cloud computing is the process of offering technology services through the Internet, which often includes data storage, networking, and servers. Before SaaS was available, companies looking to update the software on their computers had to purchase compact disks containing the updates and download them onto their systems.

For large organizations, updating software was a time-consuming endeavor. Over time, software updates became available for download through the Internet, with companies purchasing additional licenses rather than additional disks. However, a copy of the software still needed to be installed on all devices that needed access to it.

With SaaS, users don’t need to install or update any software. Instead, users can log in through the Internet or web browser and connect to the service provider’s network to access the particular service.

Advantages and Disadvantages of SaaS

Advantages

SaaS offers a variety of advantages over traditional software licensing models. Because the software does not live on the licensing company’s servers, there is less demand for the company to invest in new hardware.

It is easy to implement, easy to update and debug, and can be less expensive (or at least have lower up-front costs) since users pay for SaaS as they go instead of purchasing multiple software licenses for multiple computers.

SaaS has numerous applications, including:

- Email services
- Auditing functions
- Automating sign-up for products and services
- Managing documents, including file sharing and document collaboration
- Shared company calendars, which can be used for scheduling events
- Customer relationship management (CRM) systems, which are essentially a database of client and prospect information. SaaS-based CRMs can be used to hold company contact information, business activity, products purchased as well as track leads.

Types of software that have migrated to a SaaS model are often focused on enterprise-level services, such as human resources. These types of tasks are often collaborative in nature, requiring employees from various departments to share, edit, and publish material while not necessarily in the same office.

Disadvantages

Drawbacks to the adoption of SaaS centeraround data security and speed of delivery. Because data is stored on external servers, companies have to be sure that it is safe and cannot be accessed by unauthorized parties.

Slow Internet connections can reduce performance, especially if the cloud servers are being accessed from far-off distances. Internal networks tend to be faster than Internet connections. Due to its remote nature, SaaS solutions also suffer from a loss of control and a lack of customization.

Examples of SaaS

Google Docs

One of the simplest real-world examples of SaaS is Google Docs, Google's free online word processor.

In order to use Google Docs, all you need to do is log in on a web browser for instant access. Google Docs allows you to write, edit, and even collaborate with others wherever you happen to be.

Google Docs was launched in October 2012.

Dropbox

Dropbox is another simple example of SaaS in real life. Dropbox is a cloud storage service that lets businesses store, share, and collaborate on files and data. For example, users are able to back up and sync photos, videos, and other files to the cloud and access them from any device, no matter where they are.

Dropbox was founded in 2007.

5.4.2 PaaS

PaaS, or Platform-as-a-Service, is a cloud computing model that provides customers a complete cloud platform—hardware, software, and infrastructure—for developing, running, and managing applications without the cost, complexity, and inflexibility that often comes with building and maintaining that platform on-premises.

The PaaS provider hosts everything—servers, networks, storage, operating system software, databases, development tools—at their data center. Typically customers can pay a fixed fee to provide a specified amount of resources for a specified number of users, or they can choose 'pay-as-you-go' pricing to pay only for the resources they use. Either option enables PaaS customers to build, test, deploy run, update and scale applications more quickly and inexpensively they could if they had to build out and manage their own on-premises platform.

Every leading cloud service provider—including Amazon Web Services (AWS), Google Cloud, IBM Cloud and Microsoft Azure—has its own PaaS offering. Popular PaaS solutions are also available as open source projects (e.g. Apache Stratos, Cloud Foundry) or from software ventors (e.g. Red Hat OpenShift and Salesforce Heroku).

Benefits of PaaS

The most commonly-cited benefits of PaaS, compared to an on-premises platform, include:

- **Faster time to market.** With PaaS, there's no need to purchase and install the hardware and software you use to build and maintain your application development platform—and no need for development teams to wait while you do this. You simply tap into the cloud service provider's PaaS to begin provisioning resources and developing immediately.

- **Affordable access to a wider variety of resources.** PaaS platforms typically offer access to a wider range of choices up and down the application stack— including operating systems, middleware, databases and development tools—than most organizations can practically or affordably maintain themselves.

- **More freedom to experiment, with less risk.** PaaS also lets you try or test new operating systems, languages and other tools without having to make substantial investments in them, or in the infrastructure required to run them.

- **Easy, cost-effective scalability.** With an on-premises platform, scaling is always expensive, often wasteful and sometimes inadequate: You have to purchase additional compute, storage and networking capacity in anticipation of traffic spikes; much of that capacity sits idle during low-traffic periods, and none of it can be increased in time to accommodate unanticipated surges. With PaaS, you can purchase additional capacity, and start using it immediately, whenever you need it.

- **Greater flexibility for development teams.** PaaS services provide a shared software development environment that allows development and operations teams access to all the tools they need, from any location with an internet connection.

- **Lower costs overall.** Clearly PaaS reduces costs by enabling an organization to avoid capital equipment expense associated with building and scaling an application platform. But PaaS also can also reduce or eliminate software licensing costs. And by handling patches, updates and other administrative tasks, PaaS can reduce your overall application management costs.

How PaaS works

In general, PaaS solutions have three main parts:

- Cloud infrastructure including virtual machines (VMs), operating system software, storage, networking, firewalls

- Software for building, deploying and managing applications

- A graphic user interface, or GUI, where development or DevOps teams can do all their work throughout the entire application lifecycle

Because PaaS delivers all standard development tools through the GUI online interface, developers can log in from anywhere to collaborate on projects, test new applications, or roll out completed products. Applications are designed and developed right in the PaaS using middleware. With streamlined workflows, multiple development and operations teams can work on the same project simultaneously.

PaaS providers manage the bulk of your cloud computing services, such as servers, runtime and virtualization. As a PaaS customer, your company maintains management of applications and data.

Use cases for PaaS

By providing an integrated and ready-to-use platform—and by enabling organizations to offload infrastructure management to the cloud provider and focus on building, deploying and managing applications—PaaS can ease or advance a number of IT initiatives, including:

- **API development and management**: Because of its built-in frameworks, PaaS makes it much simpler for teams to develop, run, manage and secure APIs (application programming interfaces) for sharing data and functionality between applications.
- **Internet of Things (IoT)**: Out of the box, PaaS can support a range of programming languages (Java, Python, Swift, etc.), tools and application environments used for IoT application development and real-time processing of data generated by IoT devices.
- **Agile development and DevOps:** PaaS can provide fully-configured environments for automating the software application lifecycle including integration, delivery, security, testing and deployment.
- **Cloud migration and cloud-native development:** With its ready-to-use tools and integration capabilities, PaaS can simplify migration of existing applications to the cloud—particularly via replatforming (moving an application to the cloud with modifications that take better advantage of cloud scalability, load balancing and other capabilities) or refactoring (re-architecting some or all of an application using microservices, containers and other cloud-native technologies).
- **Hybrid cloud strategy:** Hybrid cloud integrates public cloud services, private cloud services and on-premises infrastructure and provides orchestration, management and application portability across all three. The result is a unified and flexible distributed computing environment, where an organization can run and scale its traditional (legacy) or cloud-native workloads on the most appropriate computing model. The right PaaS solution allows developers to build once, then deploy and mange anywhere in a hybrid cloud environment.

5.4.3. IaaS

Infrastructure-as-a-Service, commonly referred to as simply "IaaS," is a form of cloud computing that delivers fundamental compute, network, and storage resources to consumers on-demand, over the internet, and on a pay-as-you-go basis. IaaS enables end users to scale and shrink resources on an as-needed basis, reducing the need for high, up-front capital expenditures or unnecessary "owned" infrastructure, especially in the case of "spiky" workloads. In contrast to PaaS and SaaS (even newer computing models like containers and serverless), IaaS provides the lowest-level control of resources in the cloud.

IaaS emerged as a popular computing model in the early 2010s, and since that time, it has become the standard abstraction model for many types of workloads. However, with the advent of new technologies, such as containers and serverless, and the related rise of the microservices application pattern, IaaS remains foundational but is in a more crowded field than ever.

IaaS platform and architecture

IaaS is made up of a collection of physical and virtualized resources that provide consumers with the basic building blocks needed to run applications and workloads in the cloud.

- **Physical data centers:** IaaS providers will manage large data centers, typically around the world, that contain the physical machines required to power the various layers of abstraction on top of them and that are made available to end users over the web. In most IaaS models, end users do not interact directly with the physical infrastructure, but it is provided as a service to them.

- **Compute:** IaaS is typically understood as virtualized compute resources, so for the purposes of this article, we will define IaaS compute as a virtual machine. Providers manage the hypervisors and end users can then programmatically provision virtual "instances" with desired amounts of compute and memory (and sometimes storage). Most providers offer both CPUs and GPUs for different types of workloads. Cloud compute also typically comes paired with supporting services like auto scaling and load balancing that provide the scale and performance characteristics that make cloud desirable in the first place.

- **Network:** Networking in the cloud is a form of Software Defined Networking in which traditional networking hardware, such as routers and switches, are made available programmatically, typically through APIs. More advanced networking use cases involve the construction of multi-zone regions and virtual private clouds, both of which will be discussed in more detail later.

- **Storage:** The three primary types of cloud storage are block storage, file storage, and object storage. Block and file storage are common in traditional data centers but can often struggle with scale, performance and distributed characteristics of cloud. Thus, of the three, object storage has thus become the most common mode of storage in the cloud given that it is highly distributed (and thus resilient), it leverages commodity hardware, data can be accessed easily over HTTP, and scale is not only essentially limitless but performance scales linearly as the cluster grows.

Advantages

Taken together, there are many reasons why someone would see cloud infrastructure as a potential fit:

- **Pay-as-you-Go:** Unlike traditional IT, IaaS does not require any upfront, capital expenditures, and end users are only billed for what they use.

- **Speed:** With IaaS, users can provision small or vast amounts of resources in a matter of minutes, testing new ideas quickly or scaling proven ones even quicker.

- **Availability:** Through things like multizone regions, the availability and resiliency of cloud applications can exceed traditional approaches.
- **Scale:** With seemingly limitless capacity and the ability to scale resources either automatically or with some supervision, it's simple to go from one instance of an application or workload to many.
- **Latency and performance:** Given the broad geographic footprint of most IaaS providers, it's easy to put apps and services closers to your users, reducing latency and improving performance.

5.5 AVAILABILITY MANAGEMENT

Availability is the heart of IT service management: it has the greatest responsibility in determining IT service value. It is one of three Information Security pillars under the C.I.A. approach. That's why it is understandable when customers like Andy cause a commotion if availability isn't treated with the care it should have—particularly if the service provider is slow and unclear in communicating the incident and resolution efforts.

According to ITIL® 4, availability is the ability of an IT service or other configuration item to perform its agreed function when required. So, if you can't log in to Facebook or download your emails or access your Salesforce dashboard, your immediate reaction is to deem that service unavailable.

The purpose of availability management is to ensure that services deliver agreed levels of availability to meet the needs of customers and users. The more critical a service is to the customer, the more the company should invest in its availability. We gain insights regarding the bare minimum of what comprises availability management from the ISO/IEC 20000 standard:

- Assessing and documenting risks to service availability at regular intervals
- Determining and documenting service availability requirements and targets, by considering relevant business requirements, service requirements, SLAs, and risks
- Monitoring and recording service availability results and comparing to targets
- Investigating and addressing instances of unplanned non-availability

Availability management works hand-in-hand with other practices such as architecture, change and configuration, release and deployment, and incident and problem management in order to ensure that elements such as capacity, continuity, and security are designed, built, deployed and managed effectively across the life of the service and its underlying infrastructure and components. A holistic view is required as there are countless availability risks in the ITSM domain, such as expired certificates, poorly planned configuration changes, human error, and vendor-related failures, among others.

Monitoring and measurement of availability must consider both the component view (through events and alerts) as well as the customer view (based on complaints and usage patterns). The success of availability management at a service level will be measured by two main metrics:

- Mean time to restore service (MTRS): How quickly your company addresses non-availability, e.g. 4 hours
- Mean time between failures (MTBF): The frequency of non-availability, e.g. twice a year

The focus of availability management has shifted from designing systems that are fault tolerant (addressing MTBF) towards designing systems that recover quickly. This has brought forward concepts such as the antifragile software movement that thrive on volatility and surprise. Techniques such as auto scaling, microservices, and chaos engineering are now quite prevalent in this area.

The Availability Manager role

While the job title Availability Manager isn't one that stands out in today's age (though organizations do still recruit for this role), the role of managing availability is part and parcel of ITSM environments, particularly those of an operational nature.

Interestingly, the European e-competence framework does not list 'Availability' in any title of its 40 reference dimensions or in the 30 European ICT Professional Role Profiles. A quick search, however, reveals that availability knowledge is required in several roles and activities:

- Architecture design
- Problem management
- Information security strategy development
- Information security management
- The data administrator role
- The DevOps expert role

Whether you're a solution architect, software developer, systems administrator, or service desk support specialist, availability management will always be critical to your KPIs or OKRs. An excellent example is the site reliability engineer (SRE): availability is among the role's top elements as it is essential to protecting, providing, and progressing software and systems.

Availability manager tasks and responsibilities

To get an idea of expectations for your Availability Manager, SFIA 7 defines three availability management responsibility levels, categorized under Delivery and Operation (sub-category: Service Design). These are examples of higher responsibility, so an availability manager for these levels would be in leadership and/or have significant expertise:

Availability management: Level 4

- Contributes to the availability management process and its operation and performs defined availability management tasks.
- Analyzes service and component availability, reliability, maintainability and serviceability.
- Ensures that services and components meet and continue to meet all agreed performance targets and service levels.
- Implements arrangements for disaster recovery and documents recovery procedures.
- Conducts testing of recovery procedures.

Availability management: Level 5

- Provides advice, assistance, and leadership associated with the planning, design, and improvement of service and component availability, including the investigation of all breaches of availability targets and service non-availability, with the instigation of remedial activities.
- Plans arrangements for disaster recovery together with supporting processes and manages the testing of such plans.

Availability management: Level 6

Sets policy and develops strategies, plans, and processes for the design, monitoring, measurement, maintenance, reporting and continuous improvement of service and component availability, including the development and implementation of new availability techniques and methods.

5.6 ACCESS CONTROL

Access Control in cloud security is a system with which a company can regulate and monitor permissions, or access to their business data by formulating various policies suited chosen by the company. Access control in cloud security helps companies gain macro-level visibility into their data and user behavior, which a cloud app may not be able to offer, given their on-demand services and mobility.

Today, data is the most valuable asset of a company, safeguarding it is the next thing to do! Access Control in cloud computing gives companies the control to restrict unauthorized user access and, at the same time, give enough access for smooth functioning at work.

CloudCodes Access Control in cloud security lets companies formulate policies to restrict access through specific IP addresses, browsers, devices, and during specified time shifts. Here's an in-depth view of our Access Control in cloud computing solution.

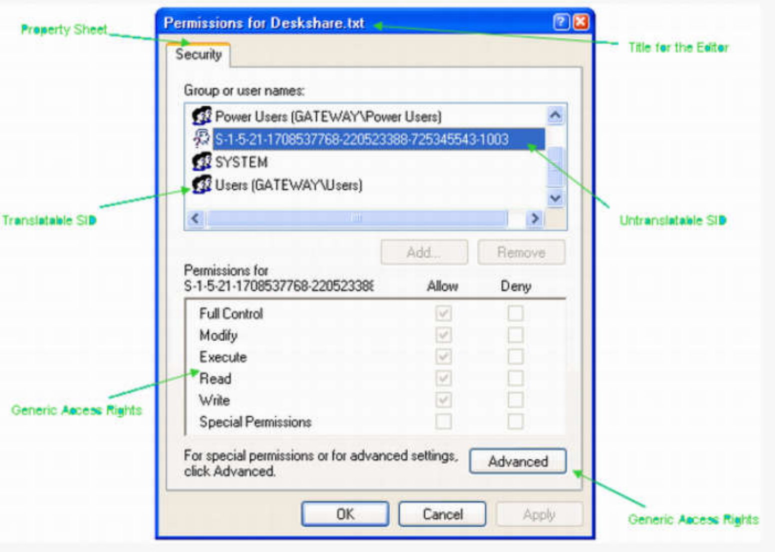
5.5.1 Access Control Models

1. Discretionary Access Control (DAC) –

DAC is a type of access control system that assigns access rights based on rules specified by users. The principle behind DAC is that subjects can determine who has access to their objects. The DAC model takes advantage of using access control lists (ACLs) and capability tables. Capability tables contain rows with ‘subject’ and columns containing ‘object’. The security kernel within the operating system checks the tables to determine if access is allowed. Sometimes a subject/program may only have access to read a file; the security kernel makes sure no unauthorized changes occur.

Implementation –

This popular model is utilized by some of the most popular operating systems, like Microsoft Windows file systems.



Permissions for Deskshare.txt

Group or user names:

Power Users (GATEWAY\Power Users)
S-1-5-21-1708537768-220523388-725345543-1003
SYSTEM
Users (GATEWAY\Users)

Permissions for S-1-5-21-1708537768-22052338€	Allow	Deny
Full Control	☑	□
Modify	☑	□
Execute	☑	□
Read	☑	□
Write	☑	□
Special Permissions	□	□

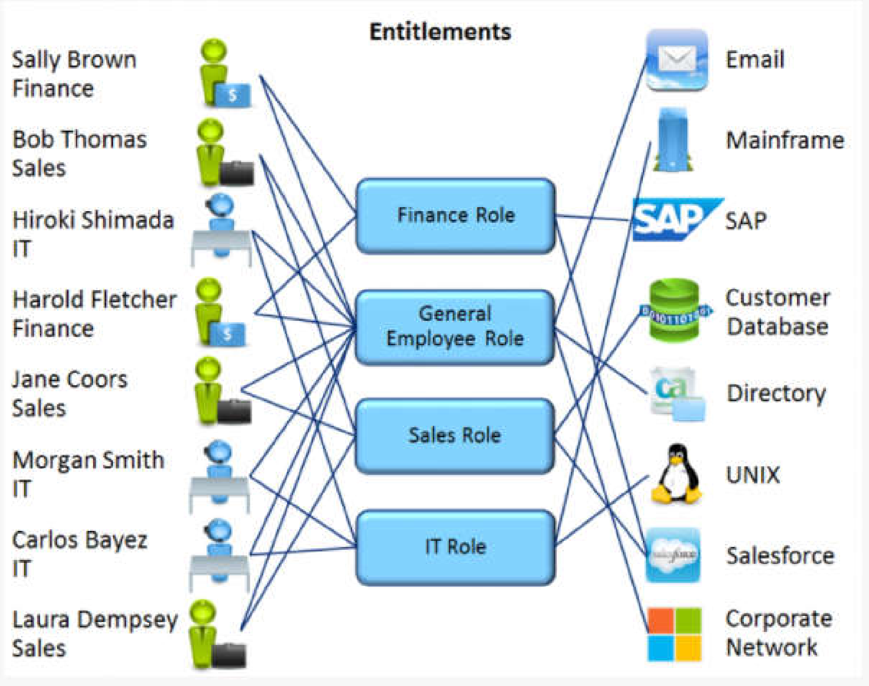
For special permissions or for advanced settings, click Advanced.

2. Role-Based Access Control (RBAC) –

RBAC, also known as a non-discretionary access control, is used when system administrators need to assign rights based on organizational roles instead of individual user accounts within an organization. It presents an opportunity for the organization to address the principle of 'least privilege'. This gives an individual only the access needed to do their job, since access is connected to their job.

Implementation-

Windows and Linux environments use something similar by creating 'Groups'. Each group has individual file permissions and each user is assigned to groups based on their work role. RBAC assigns access based on roles. This is different from groups since users can belong to multiple groups but should only be assigned to one role. Example roles are: accountants, developer, among others. An accountant would only gain access to resources that an accountant would need on the system. This requires the organization to constantly review the role definitions and have a process to modify roles to segregate duties. If not, role creep can occur. Role creep is when an individual is transferred to another job/group and their access from their previous job stays with them.

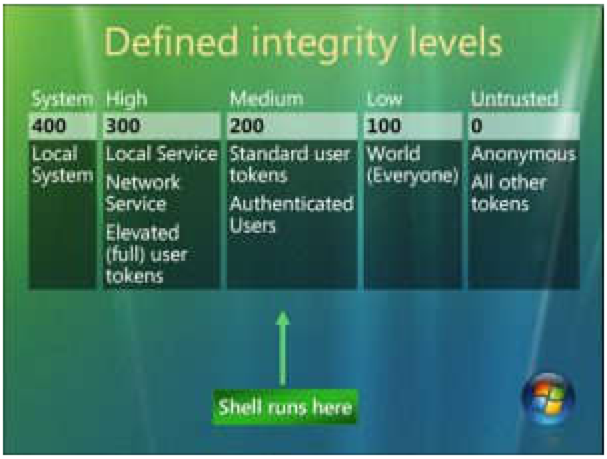


3. Mandatory Access Control (MAC) –

Considered the strictest of all levels of access control systems. The design and implementation of MAC is commonly used by the government. It uses a hierarchical approach to control access to files/resources. Under a MAC environment, access to resource objects is controlled by the settings defined by a system administrator. This means access to resource objects is controlled by the operating system based on what the system administrator configured in the settings. It is not possible for users to change access control of a resource. MAC uses "security labels" to assign resource objects on a system. There are two pieces of information connected to these security labels: classification (high, medium, low) and category (specific department or project – provides "need to know"). Each user account is also assigned classification and category properties. This system provides users access to an object if both properties match. If a user has high classification but is not part of the category of the object, then the user cannot access the object. MAC is the most secure access control but requires a considerable amount of planning and requires a high system management due to the constant updating of objects and account labels.

Implementation-

Other than the government's implementation of MAC, Windows Vista-8 used a variant of MAC with what they called, Mandatory Integrity Control (MIC). This type of MAC system added integrity levels (IL) to process/files running in the login session. The IL represented the level of trust the object would have. Subjects were assigned an IL level, which was assigned to their access token. IL levels in MIC were: low, medium, high, and system. Under this system, access to an object was prohibited unless the user had the same level of trust, or higher than the object. Windows limited the user to not being able to write or delete files with a higher IL. It first compared IL levels, then moved on to checking the ACLs to make sure the correct permissions are in place. This system took advantage of the Windows DAC system ACLs and combined it with integrity levels to create a MAC environment.



Defined integrity levels

System	High	Medium	Low	Untrusted
400	300	200	100	0
Local System	Local Service Network Service Elevated (full) user tokens	Standard user tokens Authenticated Users	World (Everyone)	Anonymous All other tokens

Shell runs here

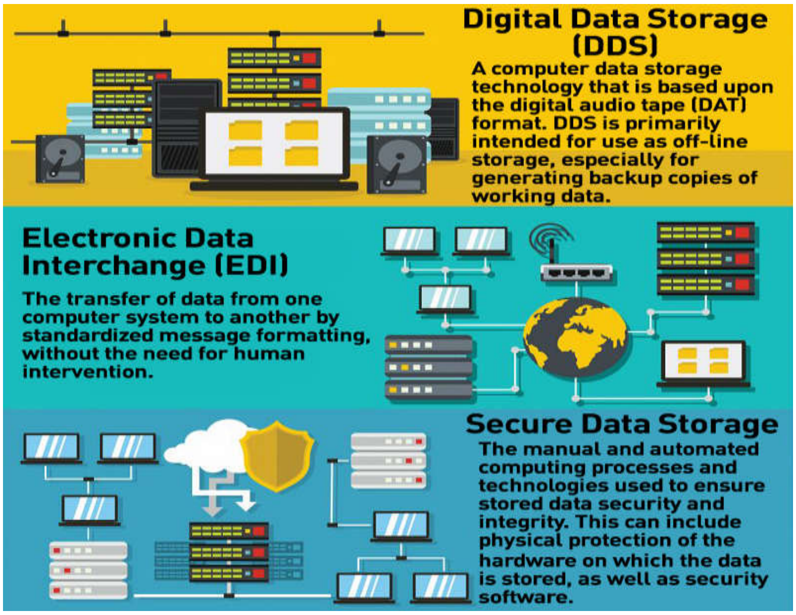
5.7 DATA SECURITY AND STORAGE IN CLOUD

Data storage security involves protecting storage resources and the data stored on them – both on-premises and in external data centers and the cloud – from accidental or deliberate damage or destruction and from unauthorized users and uses. It's an area that is of critical importance to enterprises because the majority of data breaches are ultimately caused by a failure in data storage security.

Secure Data Storage:

Secure Data Storage collectively refers to the manual and automated computing processes and technologies used to ensure stored data security and integrity. This can include physical protection of the hardware on which the data is stored, as well as security software.

Secure data storage applies to data at rest stored in computer/server hard disks, portable devices – like external hard drives or USB drives – as well as online/cloud, network-based storage area network (SAN) or network attached storage (NAS) systems.



A computer data storage technology that is based upon the digital audio tape (DAT) format. DDS is primarily intended for use as off-line storage, especially for generating backup copies of working data.

The transfer of data from one computer system to another by standardized message formatting, without the need for human intervention.

The manual and automated computing processes and technologies used to ensure stored data security and integrity. This can include physical protection of the hardware on which the data is stored, as well as security software.

How Secure Data Storage is Achieved:

- Data encryption
- Access control mechanism at each data storage device/software
- Protection against viruses, worms and other data corruption threats
- Physical/manned storage device and infrastructure security
- Enforcement and implementation of layered/tiered storage security architecture

Secure data storage is essential for organizations which deal with sensitive data, both in order to avoid data theft, as well as to ensure uninterrupted operations.

Storage Vulnerabilities:

Another huge driver of interest in data storage security is the vulnerabilities inherent in storage systems. They include the following:

- **Lack of encryption** — While some high-end NAS and SAN devices include automatic encryption, plenty of products on the market do not include these capabilities. That means organizations need to install separate software or an encryption appliance in order to make sure that their data is encrypted.

- **Cloud storage** — A growing number of enterprises are choosing to store some or all of their data in the cloud. Although some argue that cloud storage is more secure than on-premises storage, the cloud adds complexity to storage environments and often requires storage personnel to learn new tools and implement new procedures in order to ensure that data is adequately secured.

- **Incomplete data destruction** — When data is deleted from a hard drive or other storage media, it may leave behind traces that could allow unauthorized individuals to recover that information. It's up to storage administrators and managers to ensure that any data erased from storage is overwritten so that it cannot be recovered.

- **Lack of physical security** — Some organizations don't pay enough attention to the physical security of their storage devices. In some cases they fail to consider that an insider, like an employee or a member of a cleaning crew, might be able to access physical storage devices and extract data, bypassing all the carefully planned network-based security measures.

Data Storage Security Principles:

At the highest level, data storage security seeks to ensure "CIA" – confidentiality, integrity, and availability.

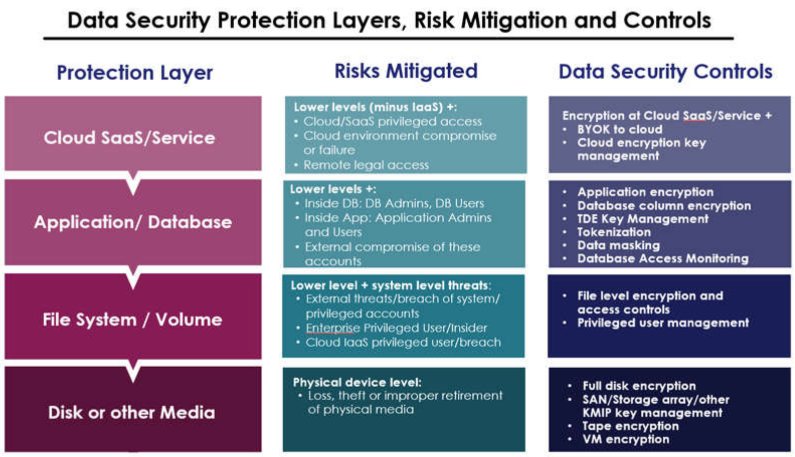
- **Confidentiality:** Keeping data confidential by ensuring that it cannot be accessed either over a network or locally by unauthorized people is a key storage security principle for preventing data breaches.

- **Integrity:** Data integrity in the context of data storage security means ensuring that the data cannot be tampered with or changed.

- **Availability:** In the context of data storage security, availability means minimizing the risk that storage resources are destroyed or made inaccessible either deliberately – say during a DDoS attack – or accidentally, due to a natural disaster, power failure, or mechanical breakdown.

Data Security Best Practices:

In order to respond to these technology trends and deal with the inherent security vulnerabilities in their storage systems, experts recommend that organizations implement the following data security best practices:



Data Security Protection Layers, Risk Mitigation and Controls

Protection Layer	Risks Mitigated	Data Security Controls
Cloud SaaS/Service	Lower levels (minus IaaS) +: • Cloud/SaaS privileged access • Cloud environment compromise or failure • Remote legal access	Encryption at Cloud SaaS/Service + • BYOK to cloud • Cloud encryption key management
Application/ Database	Lower levels +: • Inside DB: DB Admins, DB Users • Inside App: Application Admins and Users • External compromise of these accounts	• Application encryption • Database column encryption • TDE Key Management • Tokenization • Data masking • Database Access Monitoring
File System / Volume	Lower level + system level threats: • External threats/breach of system/ privileged accounts • Enterprise Privileged User/Insider • Cloud IaaS privileged user/breach	• File level encryption and access controls • Privileged user management
Disk or other Media	Physical device level: • Loss, theft or improper retirement of physical media	• Full disk encryption • SAN/Storage array/other KMIP key management • Tape encryption • VM encryption

1. **Data storage security policies** — Enterprises should have written policies specifying the appropriate levels of security for the different types of data that it has. Obviously, public data needs far less security than restricted or confidential data, and the organization needs to have security models, procedures and tools in place to apply appropriate protections. The policies should also include details on the security measures that should be deployed on the storage devices used by the organization.

2. **Access control** — Role-based access control is a must-have for a secure data storage system, and in some cases, multi-factor authentication may be appropriate. Administrators shouldalso be sure to change any default passwords on their storage devices and to enforce the use of strong passwords by users.

3. **Encryption** — Data should be encrypted both while in transit and at rest in the storage systems. Storage administrators also need to have a secure key management systems for tracking their encryption keys.

4. **Data loss prevention** — Many experts say that encryption alone is not enough to provide full data security. They recommend that organizations also deploy data loss prevention (DLP) solutions that can help find and stop any attacks in progress.

5. **Strong network security** — Storage systems don't exist in a vacuum; they should be surrounded by strong network security systems, such as firewalls, anti-malware protection, security gateways, intrusion detection systems and possibly advanced analytics and machine learning based security solutions. These measures should prevent most cyberattackers from ever gaining access to the storage devices.

6. **Strong endpoint security** — Similarly, organizations also need to make sure that they have appropriate security measures in place on the PCs, smartphones and other devices that will be accessing the stored data. These endpoints, particularly mobile devices, can otherwise be a weak point in an organization's cyberdefenses.

7. **Redundancy** — Redundant storage, including RAID technology, not only helps to improve availability and performance, in some cases, it can also help organizations mitigate security incidents.

8. **Backup and recovery** — Some successful malware or ransomware attacks compromise corporate networks so completely that the only way to recover is to restore from backups. Storage managers need to make sure that their backup systems and processes are adequate for these type of events, as well as for disaster recovery purposes. In addition, they need to make sure that backup systems have the same level of data security in place as primary systems.

5.8 SUMMARY

- **Cybersecurity** is the protection of computer systems and networks from information disclosure, theft of or damage to their hardware, software, or electronic data, as well as from the disruption or misdirection of the services they provide.

- Cloud computing provides customers with highly scalable and on-mend computing resources. NIST specified three cloud service models: Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructural as a Service (IaaS), each service models target a specific need of customers.

- Authentication is the process for confirming the identity of the user. The traditional authentication process allows the system to identify the user through a username and then validate their identity through password.

- **Virtualization** plays a very important role in the cloud computing technology, normally in the cloud computing, users share the data present in the clouds like application etc, but actually with the help of virtualization users shares the Infrastructure.

- Virtualization has eased many aspects of IT management but has also complicated the task of cyber security.The nature of virtualization introduces a new threat matrix, and administrators need to address the resulting vulnerabilities in their enterprise environments.

- Availability is the heart of IT service management: it has the greatest responsibility in determining IT service value.

- Access Control in cloud security is a system with which a company can regulate and monitor permissions, or access to their business data by formulating various policies suited chosen by the company.

- Data storage security involves protecting storage resources and the data stored on them – both on-premises and in external data centers and the cloud – from accidental or deliberate damage or destruction and from unauthorized users and uses. It’s an area that is of critical importance to enterprises because the majority of data breaches are ultimately caused by a failure in data storage security.

5.9 QUESTIONS

1. How concepts of Security apply in the cloud?
2. Write a short note on Authentication.
3. What are the alogorithms used for Authentication in Cloud?
4. Write a short note on Extensible Authentication Protocol-CHAP, Lightweight Directory Access Protocol, Single Sign-on (SSO) protocol.
5. What do you mean by Virtualization?
6. What are Virtualization System Security Issues ?
7. Write a short note on ESX and ESXi Security.
8. What are Virtualization System Vulnerabilities.
9. Write a short note on Saas, Paas, Iaas.
10. What do you mean by Availability Management?
11. Write a short note on Access control

5.10 REFERENCE FOR FURTHER READING

- https://about.usps.com/handbooks/as805h/as805h_ch4.htm
- https://www.sumologic.com/glossary/cloud-computing-security/
- https://www.ijser.org/researchpaper/The-Cloud-Computing-Security-Secure-User-Authentication.pdf
- http://www.iosrjournals.org/iosr-jce/papers/conf.15013/Volume%204/7.%2030-35.pdf
- http://www.techadvisory.org/2019/05/what-are-the-common-challenges-of-virtualization/
- https://www.nakivo.com/blog/vmware-esx-vs-esxi-key-differences-overview/
- https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.security.doc/GUID-B39474AF-6778-499A-B8AB-E973BE6D4899.html
- https://www.vmware.com/pdf/esx2_security.pdf
- https://searchitchannel.techtarget.com/tip/VMware-ESX-essentials-Virtual-Machine-File-System

- https://pentestlab.blog/2013/02/25/common-virtualization-vulnerabilities-and-how-to-mitigate-risks/
- https://www.investopedia.com/terms/s/software-as-a-service-saas.asp
- https://www.ibm.com/cloud/learn/paas
- https://www.bmc.com/blogs/availability-management-introduction/
- https://westoahu.hawaii.edu/cyber/best-practices/best-practices-weekly-summaries/access-control/

❖❖❖❖

6

MOBILE SECURITY

Unit Structure

6.0 OBJECTIVES

1. To understand the basics of Mobile security
2. To understand Mobile application security
3. To understand the concepts which maily deals with the protection of mobile devices.
4. To understand the various possible plans taken into the considerations to protect the sensitive data or information which is stored and transmitted by various devices such laptops, smartyphones etc.

6.1 INTRODUCTION

Mobile security or mobile device security is nothing but ensuring protection of laptops, smartphones, and tablets from threats which is linked with wireless computing or communications. It is nothing but a strategy, framework, and software which is used to protect any devices that moves with users including smartphones, phones.

6.2 MOBILE SYSTEM ARCHITECTURES [1]

The mobile application architecture is a collection of patterns and techniques used to construct the whole structure of the mobile application. It is the app's backbone, determining how it functions. A mobile app architecture pattern includes everything in the app, including the UI/UX, platform, technology stack, data storage, and so on. Within an application, the design of the mobile app architecture typically includes numerous layers:

1. Presentation Layer
2. Business Layer
3. Data Layer

The presentation layer consists primarily of User Interface components. The definition of the client's profile is a vital stage in building this layer so that all of the visual elements and their layout please your users. Of course, it all boils down to investigating a set of application receivers and then turning the results into an effective UI while keeping the correct User Experience in mind.

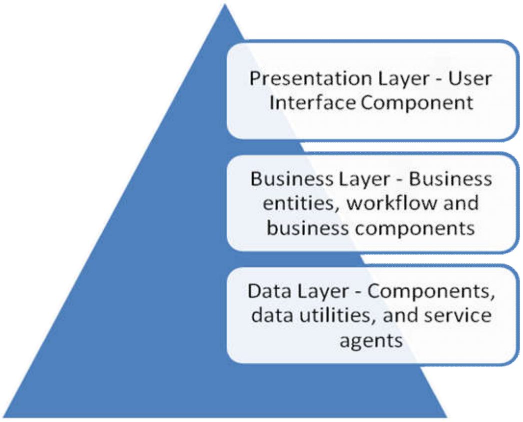

Presentation Layer - User Interface Component
Business Layer - Business entities, workflow and business components
Data Layer - Components, data utilities, and service agents

Fig 1: Mobile System Architecture

Business layer consists of business entities, workflows, business components, and all technical. This layer contains all of the mobile application development tasks. Complex business procedures and policies are also possible. There is only the application's façade, which includes the core process, components, and entities, i.e. everything connected to the app's logic and business.

Components, data utilities, and service agents comprise the data layer.

There are mainly two types of mobile architecture namely Android and IOS mobile architecture.

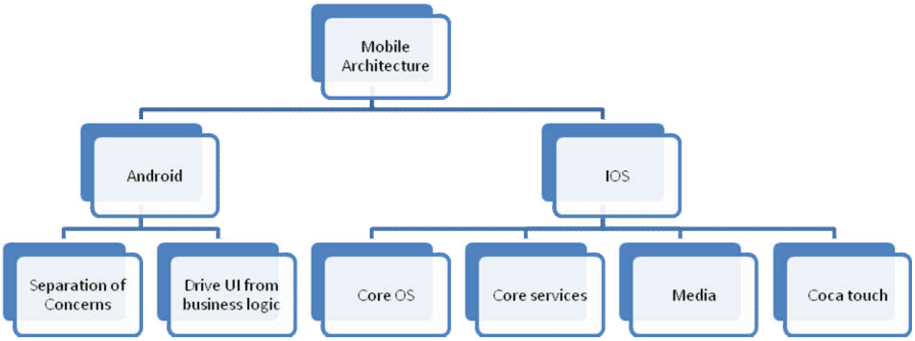


Fig 2: Types of architectures with principles

Activities, fragments, services, content providers, and broadcast receivers are all common app components of an Android app. The Android OS then utilizes this file to determine how to incorporate the app into the overall user experience of the device. Because a well-written Android app comprises several components and users frequently engage with multiple applications in a short period of time, apps must adapt to various types of user-driven workflows and tasks.

The iOS application's architecture is layered. It includes an intermediary layer between applications and hardware, preventing them from communicating directly. In iOS, the lowest levels provide essential services, while the upper layers create the user interface. The architecture can be separated into four components by default: Core Operating System, Core Services, Media, and Coca Touch.

6.3 OVERVIEW OF MOBILE CELLULAR SYSTEMS[2]

Cellular networks are the foundational technology for mobile phones, personal communication systems, wireless networking, and other devices. The technology is being developed to replace high power transmitter/receiver systems in mobile radio telephones. For data transmission, cellular networks use lesser power, shorter range, and more transmitters.

The following are the characteristics of cellular systems:

1. Provide extremely high capacity in a narrow spectrum.
2. Reuse of radio channels across cells.
3. Use a fixed number of channels to serve an arbitrary large number of people by reusing the channel across the coverage region.
4. Communication is always between the mobile and the base station (rather than directly between the mobiles).
5. Each cellular base station is given a set of radio channels within a small geographical area known as a cell.
6. Different channel groups are assigned to neighboring cells.
7. By restricting the coverage area to within the cell's boundary, the channel groups can be reused to cover other cells.
8. Maintain interference levels at acceptable limits.
9. Reusing frequencies or planned frequencies.
10. Wireless Cellular Network Organization.

6.4 GSM AND UMTS SECURITY & ATTACKS [3]

1. **Universal Mobile Telecommunications System (UMTS)**: UMTS is an abbreviation for Universal Mobile Telecommunications System, which was built using 3GPP specifications. Its network is divided into three major components: UE (User Equipment), Radio Access Network (RAN), and Core Network. Based on different releases from the 3GPP community, many technologies are classified as UMTS. It is also known as 3G.

2. **GSM (Global System for Mobile Communication)**: GSM is an abbreviation for Global System for Mobile Communication. It is the most widely utilized mobile communication technology on the planet. It employs time division multiple access (TDMA) to transport the digitized and reduced data along a channel with two distinct streams of client data, each in its own time slot.

1. GSM security[4]

The modern cellular telecommunications system with the highest level of security is GSM. The security measures used by GSM are specified. By preserving call confidentiality and subscriber anonymity, GSM upholds end-to-end security.

To protect the user's privacy, temporary identification codes are provided to the subscriber number. By using encryption methods and frequency hopping, which can be enabled through digital systems and signaling, the privacy of the communication is maintained.

Using a challenge-response method, the GSM network verifies the subscriber's identification. The MS receives a 128-bit Random Number (RAND). Using the unique subscriber authentication key (Ki) and the authentication algorithm (A3), the MS calculates the 32-bit Signed Response (SRES) based on the encryption of the RAND. The GSM network repeats the calculation after receiving the SRES from the subscriber to confirm the subscriber's identification.

Since it is stored in the subscriber's SIM as well as the AUC, HLR, and VLR databases, the individual subscriber authentication key (Ki) is never communicated over the radio channel. If the calculated value and the received SRES match, the MS has successfully authenticated and may proceed. If the values do not match, the connection is cut off and the MS is informed that the authentication failed.

The SIM performs the calculation of the signed response. Because sensitive subscriber data, like the IMSI or the unique subscriber authentication key (Ki), is never taken out of the SIM during the authentication process, it offers increased security.

The 64-bit ciphering key (Kc) is generated by the ciphering key generating algorithm (A8), which is present in the SIM. This key is calculated by combining the ciphering key generation algorithm (A8) with the unique subscriber authentication key (Ki) and the same random number (RAND) used in the authentication procedure.

The ability to alter the ciphering key in GSM adds an extra layer of security, making the system more impenetrable to eavesdroppers. If necessary, the ciphering key may be modified on a regular basis. The SIM internally calculates the ciphering key (Kc), just like it does during the authentication procedure. As a result, the SIM never divulges private information like the unique subscriber authentication key (Ki).

Using the ciphering algorithm A5, encrypted voice and data communications between the MS and the network are made possible. The GSM network's ciphering mode request instruction starts encrypted communication. When the mobile station receives this command, it starts encrypting and decrypting data using the ciphering key (Kc) and ciphering algorithm (A5).

Privacy Regarding Subscriber Identity

Temporary Mobile Subscriber identification (TMSI) is used to protect subscriber identification confidentiality. The TMSI is provided to the mobile station after the authentication and encryption processes have been completed. The mobile station reacts after receipt. The TMSI is effective in the region where it was issued. The Location Area Identification (LAI), in addition to the TMSI, is required for communications outside the location area.

2. UMTS security

There are four security feature groups in the UMTS specification:

1. **Network access security** is the collection of security tools that guarantee users' safe access to 3G services, protecting them specifically from assaults on the (radio) access link;
2. **Network domain security** is the collection of security tools that allow nodes in the provider domain to safely communicate signaling data and defend the wireline network from assaults;
3. Security measures for securing access to mobile stations are known as **user domain security.**
4. Applications in the user and provider domains can securely exchange messages thanks to a set of security characteristics called **application domain security**.

Attacks(GSM and UMTS) [5] –

The security mechanisms utilized in GSM networks are examined in this study[5], along with network weaknesses such as SIM and SMS attacks, encryption attack or cryptography attacks, and signaling attacks. Following attacks are referred from this paper[5] only. They have referred following attacks from Lord:2003 paper.

1. **SIM-based attacks**

The GSM SIM was initially created to be as hard to tamper with, copy, or otherwise compromise as possible. The SIM has become less secure as a result of numerous problems that have been uncovered over time. The COMP 128 technique can be broken to create SIM clones, which can take up to 8–15 hours and call for physical access to the SIM. However, it has been shown that a device known as Dejan Karavic's SIM Scan is frequently in use. Although 500 randomly chosen inputs should have been adequate, IBM researchers used 1000 randomly chosen inputs instead, which cuts the time it takes to clone a SIM to only minutes or at most seconds.

2. **SMS attacks**

Text messaging, also referred to as Short Message Service (SMS), is a major source of income for any operator. People who choose to conduct business communications, reveal passwords or secret codes for banking operations, or receive system reports typically believe it due to a lack of education.The SMS-SUBMIT format is used when sending an SMS from a mobile station to the service center (SC). The SC then sends a message in SMS-DELIVER format to the recipient's mobile device.

Although the SMS-SUBMIT and SMS-DELIVER message structures differ, they both follow a common standard, and the SMS-DELIVER that follows an SMS-SUBMIT is typically predictable. A report can be requested by the network to confirm the message was sent, as well as by the message sender. These reports' structure is likewise quite predictable. It is therefore conceivable to send some SMS messages that do not notify the end user but yet provide a delivery record.

3. **Cryptography Attacks**

It has been reported that the A5/1 or A5/2 encryption technique is used to encrypt data on the GSM network. A5/0 has no encryption and is used in nations where it is politically difficult to offer cryptographic gear.

Former Soviet Union nations and various Middle Eastern nations serve as examples of this. Mathematicians and cryptography experts have investigated A5/1, the supposedly stronger of the two algorithms in use, and found that it is extremely vulnerable to cryptanalytic attacks.

There are two attacks: the Random Subgraph Attack and the Baised Birthday Attack. While the second assault only needs two seconds of data and a few minutes of processing time, the first attack needs two minutes of data and one minute of processing (Lord: 2003). There are numerous trade-off considerations for each of these attacks, but three of them can be summed succinctly.The concepts utilized in these two assaults are not only relevant to this stream cipher but also to other stream ciphers, resulting in novel security measures.

4. Signaling attacks

Transmissions between the MS and the BTS over the air interface are secure according to the A5 algorithms. They can limit real-time encryption cracking and over-the-air call interception to a certain extent. Following the BTS, traffic is transferred within the operator's network in plain text (www.gsmclone.net). This means that the attacker will be able to listen to everything broadcast, including the actual phone call as well as RAND, SRES, and Kc, if he is able to gain access to the operator's signaling network. If the attacker has access to the GSM signaling network's SS7 signaling protocol, it is entirely unsecure. In addition, the attacker could gain access to the HLR to obtain the Keys, but this is less likely given the high level of protection associated with HLR.

6.5 VULNERABILITIES IN CELLULAR SERVICES [6]

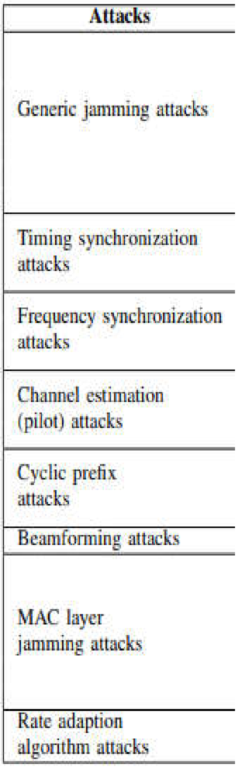
There are six security flaws in the cellular network—

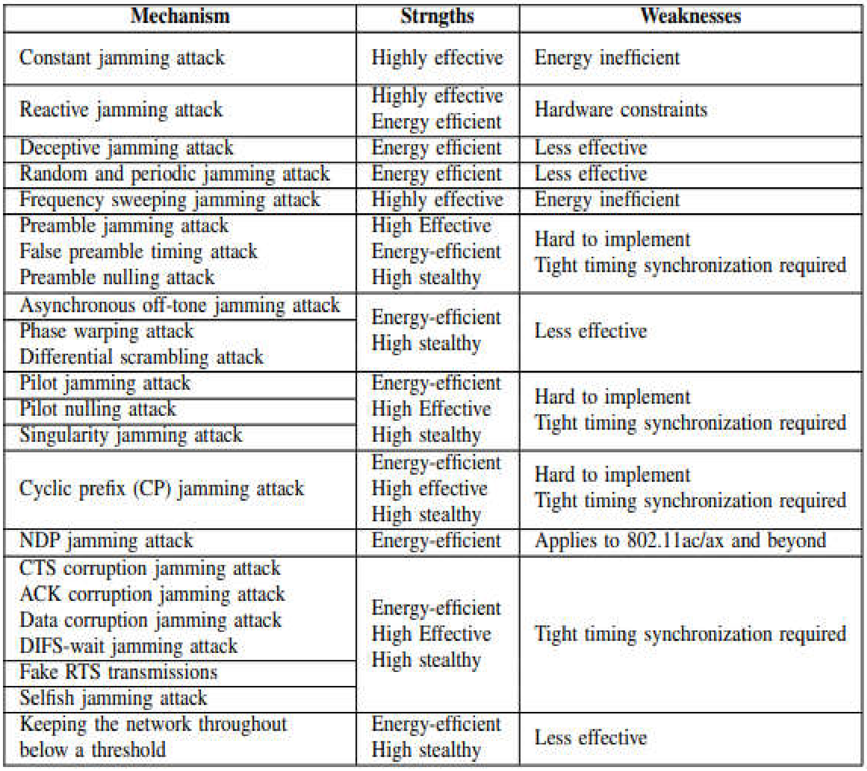
1. (MIB, SIB) Insecure broadcast messages
2. Measurement reports without verification
3. Insufficient cross-validation at the planning stage
4. Initiation of a random-access channel (RACH) without verification
5. A recovery mechanism is missing, and
6. Difficulty separating network attacks from failures

6.6 CELLULAR JAMMING ATTACKS & MITIGATION [7]

Due to the fact that they handle even more data traffic than cellular networks, WLANs are becoming more and more crucial. The protection of WLANs from jamming attacks is crucial given the prevalence of wireless applications in smart settings such as smart hospitals, smart homes, and smart buildings.[7]

Table 1: Wi-Fi Jamming attacks[7]





Attacks	Mechanism	Strngths	Weaknesses
Generic jamming attacks	Constant jamming attack	Highly effective	Energy inefficient
	Reactive jamming attack	Highly effective Energy efficient	Hardware constraints
	Deceptive jamming attack	Energy efficient	Less effective
	Random and periodic jamming attack	Energy efficient	Less effective
	Frequency sweeping jamming attack	Highly effective	Energy inefficient
Timing synchronization attacks	Preamble jamming attack False preamble timing attack Preamble nulling attack	High Effective Energy-efficient High stealthy	Hard to implement Tight timing synchronization required
Frequency synchronization attacks	Asynchronous off-tone jamming attack Phase warping attack Differential scrambling attack	Energy-efficient High stealthy	Less effective
Channel estimation (pilot) attacks	Pilot jamming attack Pilot nulling attack Singularity jamming attack	Energy-efficient High Effective High stealthy	Hard to implement Tight timing synchronization required
Cyclic prefix attacks	Cyclic prefix (CP) jamming attack	Energy-efficient High effective High stealthy	Hard to implement Tight timing synchronization required
Beamforming attacks	NDP jamming attack	Energy-efficient	Applies to 802.11ac/ax and beyond
MAC layer jamming attacks	CTS corruption jamming attack ACK corruption jamming attack Data corruption jamming attack DIFS-wait jamming attack Fake RTS transmissions Selfish jamming attack	Energy-efficient High Effective High stealthy	Tight timing synchronization required
Rate adaption algorithm attacks	Keeping the network throughout below a threshold	Energy-efficient High stealthy	Less effective

6.7 SECURITY IN CELLULAR VOIP SERVICES[8]

Following are the 12 security indicators in CellularVoIP Services

- Secure user credentials with a strong password and two-factor authentication.
- Perform regular call log reviews for unusual call activity.
- Disable international calling / enable geo-fencing.
- Outsource to a SaaS provider for VoIP calls.
- Update firmware on VoIP phones.
- Use a router with a firewall.
- Limit physical access to networking equipment.
- Restrict user access to parts of the phone system.
- Ensure data encryption through your VoIP provider.
- Educate users on VoIP security best practices.
- Prevent ghost calls on IP phones.
- Implement intrusion prevention systems.

6.8 MOBILE APPLICATION SECURITY[9]

Mobile Security is the need of hour; organizations, institutions and individuals are today actively engaged with the mobile and similar devices and all such devices are great threats due to many reasons.

In today's age of information technology, mobile security is crucial. Mobile computing is extremely close to becoming there. In other words, it is also referred to as the security of mobile-based technology, such as smart phones. Attackers typically link mobile security to smartphones, computers, and other devices. This typically includes Bluetooth, WIFI, short message service (SMS), and multimedia messaging service (MMS). However, a small number of specialists also issue warnings on operating system security due to the possibility that attackers may make use of various objects via browsers, OS, or malicious software. It is important to keep in mind that downloading apps might occasionally compromise smartphone security. Applications for privacy and integrity should be included with every smartphone or electronic device.

According to network expert the major target of the attackers are Data: As smart phone or electronic devices contains different kind of sensitive or virtual information such as –credit card no., authentication indication, audio, visual content, call log etc. So this is the prime target.

Identity: With an electronic device the owner can be indentified easily and hare attackers may use this identity for different purposes.

As far as modern threads are concern with mobile security; concerned with different objects such as

1. Boot nets
2. Spyware
3. Malicious link
4. Malicious applications

There is different attacking system for mobile security and that may cause in the following

1. Attack based on SMS and MMS

Research in Higher Education, Learning and Administration

IQAC 2019

ISBN No.: 978-81-941751-0-0

SIMS Pandeshwar & Srinivas University Mukka Page 115

2. Attack based on different kind of network like GSM network and WIFI based network

3. Web browser

4. Operating system

5. Hardware and vulnerabilities

6. Insecure software etc.

In mobile security SMS is also a weak point sometime. It causes in the mobile system having binary SMS system. It leads the denial of service attack. We can see such witness in SIMENS S55 model having Chinese Character. Similarly in earlier days few Nokia phones are also unable to recognize denial of service attack. It is important to note that distributed denial of service is also an important attack to the mobile and the telecommunication system.

In mobile security another focus attacking place is GSM network. The GSM encryption belongs to A5 algorithm and their vulnerabilities is an important concern. We can see this kind of attack in some of the Europeans countries. Gradually A5/3 and A5/4 algorithm have been popular against this kind of attack. After the development of 2G GSM we can see the vulnerabilities. The hackers in recent past can also break the GSM algorithm.

As far as the WIFI is concerned in recent past the attackers can get information of a smart phone by find out the vulnerabilities. The security of wireless network previously secured by WEP (Wired equivalent privacy algorithm) keys but the weakness of WEP altered by WIFI

Protected Access (WPA) and WPA3 algorithm. The protocol Temporal Key Integrity (TKIP) has been introduced to allow the migration of WPA2 and WPA3.

In the recent past security is also an important concern for Bluetooth system. With Bluetooth one can easily break the vulnerabilities. The attackers are required to connect to port for accessing or controlling the device or mobile. In Bluetooth system attackers send a file and if users download the file then the system may be corrupted such as CABIR (SYMBIAN).

6.9 SUMMARY

In this chapter, we learnt the GSM, UMTS, vulnerabilities, jamming attacks, mobile application security and attacks in detail. Mobile Security as a whole necessitates various defense mechanisms, however there are still some problems that make security difficult to implement. Operating Systems are one of these few crucial ones. It is important to keep in mind that some operating systems are single tasking, hence they cannot work with a firewall or antivirus program.

Another essential issue to consider is energy independence. It is important to remember that for security reasons, network usage shouldn't be too high. In addition to technological defenses, it is crucial that consumers are interested in and informed of security-related issues. Additionally, a few things—rich operating systems, secure operating systems, secure elements, and secure applications—are necessary.

6.10 REFERENCES FOR READING (REFERRED WEBSITES)

1. https://binarapps.com/mobile-architecture-what-are-the-types/

2. https://www.tutorialspoint.com/wireless_communication/wireless_communication_cellular_networks.htm#:~:text=Cellular%20network%20is%20an%20underlying%20technology%20for%20mobile,shorter%20range%20and%20more%20transmitters%20for%20data%20transmission

3. https://www.geeksforgeeks.org/difference-between-umts-and-gsm/

4. https://www.tutorialspoint.com/gsm/gsm_security.htm

5. https://www.ajol.info/index.php/tim/article/view/27226

6. https://thehackernews.com/2021/12/new-mobile-networkvulnerabilities.html#:~:text=1%20Insecure%20broadcast%20messages%20%28MIB%2C%20SIB%29%202%20Unverified,6%20Difficulty%20of%20distinguishing%20network%20failures%20from%20attacks

7. H. Pirayesh and H. Zeng, "Jamming Attacks and Anti-Jamming Strategies in Wireless Networks: A Comprehensive Survey," in IEEE Communications Surveys & Tutorials, vol. 24, no. 2, pp. 767-809, Secondquarter 2022, doi: 10.1109/COMST.2022.3159185.

8. https://telzio.com/blog/voip-security#:~:text=VoIP%20Security%3A%2012%20Best%20Practices%20for%20VoIP%20Phone,to%20parts%20of%20the%20phone%20system.%20More%20items

9. https://www.researchgate.net/publication/336845625_MOBILE_APPLICATIONS_SECURITY_AN_OVERVIEW_AND_CURRENT_TREND

❖❖❖❖

7

SECURE WIRELESS NETWORK

Unit Structure:

7.0 OBJECTIVES

1. To understand the basics of wireless network
2. To understand wireless network attacks
3. To understand how to minimize the risks of wireless network.
4. To understand how to prevent the unwanted users from accessing a particular wireless network.

7.1 INTRODUCTION

Wireless security is the prevention of unauthorized access or damage to computers or data using wireless networks. It is used for standard networks designing, implementing, and ensuring the security on a wireless network. It is basically used for the devices and networks that are connected in a wireless environment.

7.2 OVERVIEW OF WIRELESS NETWORKS

Wireless networks are computer networks that are not wired together. The majority of the time, radio waves are used for communication between network nodes. They enable network connections for devices as they are moving throughout the network's coverage area.

Fig 1: Wireless Network

Types of Wireless Networks

Wireless LANs − Connects two or more network devices using wireless distribution techniques.

Wireless MANs − Connects two or more wireless LANs spreading over a metropolitan area.

Wireless WANs − Connects large areas comprising LANs, MANs and personal networks.

Due to the lack of wires and cables, it offers workspaces that are clutter-free.

As there is no requirement for connecting devices to one another, it promotes the mobility of network devices attached to the system.Since there is no need to put out wires, accessing network devices from any area that is covered by the network or a Wi-Fi hotspot becomes convenient.Wireless networks are simpler to install and configure.Since new devices don't require wiring to the existing configuration, they can be linked to it with ease. The amount of equipment that can be added to or withdrawn from the system can also vary greatly because they are not constrained by the cable capacity. Because of this, wireless networks are immensely scalable.Wireless networks don't or only use a few cables. This lowers the expense of the setup and equipment.

7.3 SCANNING AND ENUMERATING 802.11 NETWORKS

What is 802.11 - The Institute of Electrical and Electronics Engineers (IEEE) is responsible for maintaining the 802.11 standard, which describes a link layer wireless protocol. When people hear 802.11, they frequently assume Wi-Fi, although the two are not nearly the same. Wi-Fi and 802.11 have become incredibly popular in recent years, and every new laptop has a built-in Wi-Fi adapter. When the initial 802.11 standard was ratified in 1997, transmission speeds could reach a maximum of 2 Mbps. This version of the standard permitted the use of Direct Sequence Spread

Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS) as two separate physical ways for encoding information. But because these two encoding methods are incompatible, there was a lot of uncertainty in the market.

The IEEE published 802.11b, an update to the original 802.11 standard, in 1999. The maximum transmission speed was extended to a significantly faster 11 Mbps by the 802.11b standard, which employed DSSS. 802.11a, which allowed 802.11 to operate in the unlicensed 5-GHz National Information Infrastructure (UNII) band instead of the congested 2.4-GHz Industrial, Scientific, and Medical (ISM) band, was also introduced in 1999.

Comparative analysis of Wi-Fi and 802.11

The Wi-Fi Alliance is in charge of the 802.11 subset that makes up Wi-Fi. Nearly all of the major wireless equipment makers felt they needed a smaller, more agile group dedicated to ensuring interoperability across vendors because the 802.11 standard is so vast and the procedure required to change the standard can take some time (it's governed by a committee). As a result, the Wi-Fi Alliance was founded. The Wi-Fi Alliance guarantees that all items with the Wi-Fi-certified emblem are compatible. In this way, Wi-Fi Alliance establishes the "right thing" to do in the event that any issue in the 802.11 standard arises. Additionally, it permits suppliers to put critical portions of draft standards—standards that have not yet been approved—into use. Wi-Fi Protected Access (WPA) is an illustration of this.

Choose operating system - Device drivers, which are linked to a particular operating system, are necessary for this hardware to communicate with the operating system. Various wireless hacking programs also only work on specific operating systems. When taken together, these dependencies highlight how crucial it is to choose an operating system.

Windows

The fact that Windows is likely already installed on your laptop is a benefit. Another benefit is that Windows users have access to NetStumbler, a scanning application that is simple to set up and use. Although NetStumbler will be explored in depth in the chapter's section on tools, it's crucial to keep in mind that it's an active scanner.

Linux

The obvious choice for wireless hacking is Linux. The majority of wireless tools were developed with Linux in mind, and it has the most active group of driver developers. Drivers that support monitor mode are the rule rather than the exception on Linux. Additionally, since drivers are open source, it is simple to patch or alter them to carry out more sophisticated assaults. Configuring and installing unique kernel drivers and tools can be difficult if you are new to Linux. Fortunately, a number of bootable CD distributions, including Knoppix-STD, Auditor, and PHLAK, were created with security in mind.

OS X

OS X is a peculiar creature. Although the operating system's core is open, several of its subsystems are not. Even while I think the OS X device driver subsystem is incredibly elegant, it isn't nearly as well known as the Linux or any BSD driver subsystems. This indicates that there aren't many individuals out there trying to hack OS X device drivers. Additionally, very few vendors offer any kind of OS X drivers at all, and if they do, they are frequently missing functionality like monitor mode. Michael (Mick) Rossberg, fortunately for OS X users globally, is an extremely competent and driven individual when it comes creating OS X drivers.

7.4 ATTACKING 802.11 NETWORKS

Wireless network security has a rather murky history, which is really not all that surprising given how frequently purportedly secure methods are compromised. However, 802.11 was designed to be unique. The numerous inventive and unrelated methods that WEP was cracked, however, set a record for the quantity of band-aid fixes that had to be hurriedly implemented. New methods, many of which were directly related to the band-aid fixes, were discovered not too long after the band-aids were used. This served as a wake-up call to the IEEE, which later produced 802.11i (also known as WPA2). Experts in the sector created 802.11i, which fixes the majority of the issues that have been identified in the intervening years.

Several categories can be used to group wireless network defenses. The first category—"totally ineffective," sometimes known as "security through obscurity"—is easy to get around for anyone who is sincerely interested in doing so.

The following defense style could be categorized as "annoying." WEP and a WPA-PSK password created using a dictionary fall into this category. An attacker can discover any static WEP key with enough time.

A network that requires genuine effort and some amount of talent to infiltrate is considered to be in the third category of defense once "annoying" security measures have been passed. Most networks aren't as secure as this. These networks employ WPA/WPA2 that has been properly configured. Chapter 7 goes into great detail on the methods used to attack properly configured WPA/WPA2 networks.

Last but not least, there are tools that can be used to attack wireless networks in ways that are not directly linked to wireless networking, including obtaining the WEP/WPA key from a Windows laptop without attacking it through the wireless network. This chapter discusses attacks in the order listed.

Advanced Attacks Against WEP

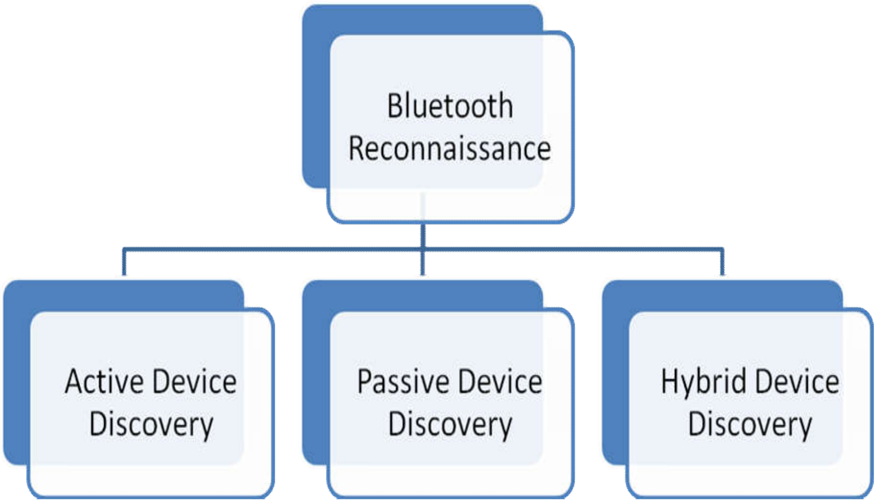
RC4 Encryption Primer - RC4 encryption works by generating a stream of random bytes. The random bytes generated are then XOR'd with the plaintext packet, and the result is called ciphertext. Before the random bytes are generated, RC4 must be initialized with a secret key. If two users both use the same secret key, they will generate the same random bytes. The user who receives the message can XOR the random bytes out of the encrypted message and re-create the original.

Inductive Chosen Plaintext Attack - If an attacker knows X bytes of the RC4 keystream generated by the secret WEP key, she can get X + 1 bytes by guessing. Consider that an attacker watches a shared key authentication exchange and, therefore, knows 128 bytes of RC4 output for the given IV.

The Fragmentation Attack - The attack allows for recovery of the WEP key using the statistical attacks mentioned previously but much more quickly. The basis for this attack is several optimizations related to creating traffic on the local network.

7.5 BLUETOOTH SCANNING AND RECONNAISSANCE

The Bluetooth specification defines 79 channels across the 2.4-GHz ISM band, each 1-MHz wide. Devices hop across these channels at a rate of 1600 times per second (every 625 microseconds). This channel-hopping technique is known as Frequency Hopping Spread Spectrum (FHSS), and in current Bluetooth implementations, the user can get 3-Mbps of bandwidth with a maximum intended distance of approximately 100 meters. FHSS provides robustness against noisy channels by rapidly moving throughout the available RF spectrum. Any set of devices wanting to communicate using Bluetooth needs to be on the same channel at the same time, as shown in the illustration. Devices that are hopping in a coordinated fashion can communicate with each other, forming a Bluetooth piconet, the basic network model used for two or more Bluetooth devices. Every piconet has a single master and between one and seven slave devices. Communication in a piconet is strictly between a slave and a master. The channel-hopping sequence utilized by a piconet is pseudorandom and can only be generated with the address and clock of the master device.



In the reconnaissance phase of a Bluetooth attack, we'll examine the process of identifyingvictim Bluetooth devices in the area through active discovery and passive discovery, using visual inspection and hybrid discovery. The goal of the discovery process is to identify the presence of Bluetooth devices, revealing each device's 48-bit MAC address or Bluetooth Device Address (BD_ADDR).

Once you have discovered a device, you can start to enumerate the services on the device, identifying potential exploit targets. You can also fingerprint the remote device and leverage Bluetooth sniffing tools to gain access to data from the piconet.

Active Device Discovery

The first step in Bluetooth reconnaissance scanning is to simply ask for information about devices within range. Known as inquiry scanning in the Bluetooth specification, a device can actively transmit inquiry scan messages on a set of frequencies, listening for responses. If a target Bluetooth device is configured in discoverable mode, it will return the inquiry scan message with an inquiry response and reveal its BD_ADDR, timing information (known as the device clock or CLK), and device class information (e.g., the device type such as phone, wearable device, toy, computer, and so on).

Windows Discovery with BlueScanner

BlueScanner is a free tool from Aruba Networks for Bluetooth scanning on Windows XP, Vista, and Windows 7 systems and is shown here in action, BlueScanner uses the Microsoft Windows Bluetooth drivers to identify and enumerate available devices, characterizing them by name, BD_ADDR, and available services. As an analysis tool, BlueScanner is unique due to the simple feature of applying a location label in the scan results, allowing you to identify any free-form string to describe the devices being discovered.

Double-clicking an entry in BlueScanner will open the Bluetooth Device Information dialog, which displays the device name and BD_ADDR information as well as detailed service information. Location information can also be changed for the specific device from this dialog.

In the device summary view on the left, BlueScanner will identify the number of devices organized by location, type (phone, headset, laptop), and services. Clicking any individual service will display only the devices running the selected service, making it easy to identify the devices to target with the Object Exchange (OBEX) Push Server, for example.

BlueScanner retains the logging information from past scans in a file calledbluescanner.dat in the same directory where the program executable is installed. This file is a standard ASCII file, delimited by carriage return and linefeed characters. Using standard Windows or Unix/Linux text-handling tools, such as findstr.exe, grep, and awk, it is possible to cull data from this file for additional reporting needs.

Passive Device Discovery

The Bluetooth specification doesn't require that two devices wishing to communicate gothrough the inquiry scan exchange. As a consequence, if you determine a device's address through some outside technique (such as reading it in the documentation), the device has to treat your connection the same as if you had discovered it actively.

Hybrid Discovery

When active device discovery and visual inspection don't work for identifying Bluetoothdevices, several hybrid discovery mechanisms are also possible.

When a device manufacturer produces a product with multiple interfaces, it must assign each interface a MAC address. Commonly, the multiple MAC addresses on a single device are relative to each other, similar to the first 5.5 bytes with the last nibble increased by one (for example, 00:21:5c:7e:70:c3 and 00:21:5c:7e:70:c4). This behavior has been used by wireless intrusion detection system (WIDS) vendors to detect a rogue AP on your network connecting through a NAT interface, by observing commonalities between IEEE 802.11 BSSID (AP MAC address) and the NAT MAC address observed onthe wired network. We can use similar logic to identify the Bluetooth interface on products such as the iPhone.

Starting with the iPhone 3G, Apple issues MAC addresses to the Wi-Fi and the Bluetooth interfaces in a one-off fashion where the Bluetooth BD_ADDR is always one address less than the Wi-Fi MAC address. You can observe this behavior on your iPhoneby tapping Settings | General | About.

7.6 BLUETOOTH EAVESDROPPING

First, Bluetooth is based on Frequency-Hopping Spread Spectrum (FHSS), where the transmitter and the receiver share knowledge of a pattern of frequencies used for exchanging data. For every piconet, the frequency pattern is unique, based on the BD_ADDR of the Bluetooth master device. Frequency hopping at a rate of 1600 hops per second (under normal conditions), the Bluetooth devices transmit and receive data for a short period of time (known as a slot) before changing to the next frequency. Under most circumstances, knowing the BD_ADDR of the piconet master is necessary to follow along with the other devices.

Second, just knowing the BD_ADDR isn't enough to frequency hop with the other devices in the piconet. In addition to knowing the frequency-hopping pattern, the sniffer must also know where in the frequency-hopping pattern the devices are at any given time. The Bluetooth specification uses another piece of information, known as the master clock or CLK, to keep track of timing for the device's location within the channel set. This value has no relationship to the time of day; rather, it is a 28-bit value incremented by oneevery 312.5 microseconds.

Finally, Bluetooth interfaces are simply not designed for the task of passive sniffing. Unlike Wi-Fi monitor-mode access, Bluetooth interfaces do not include the native ability to sniff and report network activity at the baseband layer. You can sniff local traffic at the HCI layer using Linux tools such as hcidump, but this type of sniffing does not reveal lower-layer information or activity, requires an active connection to the piconet, and only shows activity to and from the local system.

Open-Source Bluetooth Sniffing

As an alternative to the costly commercial tools designed for Bluetooth sniffing, the open-source gr-bluetooth project also can be used to capture and assess Bluetooth activity. As an open-source tool, gr-bluetooth is tremendously useful because developers are free to extend the tool's functionality as they see fit, unlike the rigid and limited usefulness of the FTS4BT product.

The gr-bluetooth project is designed to take advantage of the Universal Software Radio Peripheral (USRP) for Bluetooth traffic analysis.

In this chapter, we examined different techniques an attacker can use to observe and eavesdrop on Bluetooth traffic through traffic sniffing. Unlike IEEE 802.11, Bluetooth has several inherent physical layer characteristics through the use of frequency hopping spread spectrum that make sniffing difficult. Both commercial and open-source tools overcome these challenges to varying degrees of success, cost, and complexity.Once an attacker has established a toolkit enabling her to eavesdrop on Bluetoothtraffic, the attacker has multiple opportunities to exploit Bluetooth networks, including the ability to extract unencrypted data sent between targets and the ability to eavesdrop on Bluetooth keyboards configured in HID mode.

7.7 ATTACKING & EXPLOITING BLUETOOTH

Many organizations often overlook the security threat posed by Bluetooth devices. While significant effort is spent on deploying and hardening Wi-Fi networks through vulnerability assessments and penetration tests or ethical hacking engagements, very little is done in the field of Bluetooth security.Part of the reason why few organizations spend any resources on evaluating their Bluetooth threat is a common risk misconception: “We are indifferent about Bluetooth security because it doesn’t threaten our critical assets.” Even when organizations recognize the threat Bluetooth poses, very few people have the developed skills and expertise to implement a Bluetooth penetration test successfully or to ethically hack a given Bluetooth device.

Pin Attacks

Two devices may pair to derive a 128-bit link key that is used to authenticate the identity of the claimant device and encrypt all traffic. This pairing exchange is protected by a PIN value up to Bluetooth 2.7.Despite the availability of the Secure Simple Pairing (SSP) mechanism introduced in Bluetooth 2.1, most Bluetooth users still use the legacy pairing mechanism with PIN authentication for the initial pairing exchange. The pairing process is a point of significant

vulnerability between the devices where an attacker who can observe the pairing exchange can mount an offline brute-force attack against the PIN selection. After the pairing process is complete, subsequent connections leverage the stored 128-bit link key for authentication and key derivation, which is currently impractical to attack.In order to crack the PIN information, the attacker must discover the following pieces of information:

- IN_RAND, sent from the initiator to the responder
- Two COMB_KEY values, sent from the initiator and the responder devices
- AU_RAND, sent from the authentication claimant
- Signed Response (SRES), sent from the authentication verifier

Since the Bluetooth authentication mechanism performs mutual-authentication (the slave authenticates to the master, and vice-versa), the attacker has two opportunities to identify the AU_RAND and SRES values; either exchange is sufficient, but identifying the device performing authentication (master or slave BD_ADDR) is significant. In addition, the attacker needs to know both the slave and master BD_ADDRs, which are not transmitted over the air as part of the pairing exchange.

BTCrack

BTCrack is a Bluetooth PIN cracking tool for Windows clients written by Thierry Zoller. This tool is easy to use, though we've given it a relatively low simplicity score, due to the challenges in capturing the pairing data needed to crack the PIN.

7.8 ZIGBEE SECURITY & ATTACKS

The ZigBee specification includes features designed to protect the confidentiality and integrity of wireless communications using AES encryption and device and data authentication using a network key. To satisfy the varying security needs of ZigBee devices, two operational security modes have been defined:

- Standard security mode Formerly known as residential security mode, standard security mode provides authentication of ZigBee nodes using a single shared key where the Trust Center authorizes devices through the use of an Access Control List (ACL). This mode is less resource-intensive for devices, since each device on the network is not required to maintain a list of all device authentication credentials.
- High security mode Formerly known as commercial security mode, high security mode requires that a single device in the ZigBee network, known as the Trust Center, keep track of all the encryption and authentication keys used on the network, enforcing policies for network authentication and key updates. The Trust Center device must have sufficient resources to keep track of the authentication credentials used on the network and represents a single point of failure for the entire ZigBee network, since, if it fails, no devices will be permitted to join the network.

Zig Bee Attacks

To date, little work has been published about attacking and exploiting ZigBee. A limited number of papers have pointed out vulnerabilities inherent in IEEE 802.15.4 or ZigBee, but no tools have been widely published to exploit these vulnerabilities or otherwise assess the security of ZigBee technology. Seeing the lack of tools and techniques for evaluating the security of ZigBee networks, this author set to work in the development of an attack tool suite designed to help people evaluate the security of ZigBee implementations.

KillerBee is a Python-based framework for manipulating ZigBee and IEEE 802.15.4 networks available at http://killerbee.googlecode.com. Written and tested on Linux systems, the project is free and open-source with the goal of simplifying common attack tasks while empowering other Python tools for use in exploring ZigBee security. KillerBee includes a handful of specific attack tools developed using this framework, both for practical attacks and to demonstrate the use of the framework.

Building a KillerBee Toolkit

In order to start using the KillerBee toolkit to its full capabilities, a few steps are necessary

for building your toolkit, including the following hardware and software:

- Atmel RZ Raven USB Stick (hardware)
- Atmel JTAGICE mkII On-Chip Programmer (hardware)
- Atmel 100-mm to 50-mm JTAG standoff adapter (hardware)
- 50-mm male-to-male header (hardware)
- AVR Studio for Windows (software, free)
- KillerBee Firmware for the RZUSBSTICK (software, free)
- A Windows host for programming the RZ Raven USB Stick (one-time operation)

Eavesdropping Attacks

Because a significant number of ZigBee networks do not employ encryption, eavesdropping attacks are very useful for an attacker. Even in the cases when the ZigBee network does use encryption, an attacker can make use of unencrypted ZigBee frame information, such as the MAC header, to identify the presence of ZigBee networks and other important characteristics, such as the configuration of the network, node addresses, and the PAN ID.

A handful of tools provide the ability to capture ZigBee network traffic, ranging from inexpensive to tremendously expensive, though we'll provide some assistance in maximizing your investment.

Replay Attacks

The concept of a replay attack is simple: using observed data, retransmit the frames as if the original sender were transmitting them again. The effect of a replay attack will depend largely on the content of the data being replayed and the nature of the protocol in use.For example, in a network used for electronic banking, if an attacker can implement a replay attack and re-send a bank transfer, then the funding of the original transfer could be doubled, tripled, or quadrupled depending on the number of times the attacker replays the data. In the world of ZigBee devices, a replay attack is similar with a decidedly different impact.

Encryption Attacks

Encryption key distribution, rotation, revocation, and management in a ZigBee network is a challenge to address securely. As few ZigBee devices have a Man-Machine Interface (MMI), administrators have limited opportunity to purchase a product and configure a key locally before provisioning the device.

Defending Against a Hardware Attack

In this attack, we highlighted steps for stealing a ZigBee device and attacking the hardware to recover encryption key material. From a physical security perspective, you can protect ZigBee devices against theft through classic monitoring and theft-deterrent techniques, including video monitoring, security guards, hardware locks, and device tethers. These systems generally do not mix well with ZigBee, however, where a device may be outside in an unprotected area or, in some cases, in the hands of the consumer who is meant to use the system such as in retail locations for automated checkout and payment.

7.9 SUMMARY

Wireless networks are computer networks that are not wired together. The numerous inventive and unrelated methods that WEP was cracked, however, set a record for the quantity of band-aid fixes that had to be hurriedly implemented. The Bluetooth specification defines 79 channels across the 2.4-GHz ISM band, each 1-MHz wide. Devices hop across these channels at a rate of 1600 times per second (every 625 microseconds).

ZigBee is a quickly growing, low-speed, and extremely low-power utilization protocol, servicing multiple industry verticals such as healthcare, home automation, smart-grid systems, and security systems. While ZigBee includes mechanisms to protect data confidentiality, frequently citing the use of AES as the miracle defense against attacks, the vulnerabilities in ZigBee stem from the limited functionality of inexpensive devices, which challenge defending against eavesdropping attacks, sequence enforcement (enabling replay attacks with zbreplay), and key provisioning (enabling key compromise with zbdsniff).

7.10 REFERENCES FOR READING

1. ADVANCED ATTACKS AGAINST WEP (luskinserver.no-ip.org) - http://luskinserver.no-ip.org/DOCS-TECH/Hacking/Hacking%20Exposed/Hacking%20Exposed%20Wireless/final/bbl0045.html

2. http://luskinserver.no-ip.org/DOCS-TECH/Hacking/Hacking%20Exposed/Hacking%20Exposed%20Wireless/final/bbl0022.html

3. https://null-byte.wonderhowto.com/how-to/bt-recon-snoop-bluetooth-devices-using-kali-linux-0165049/

4. Hacking Exposed Wireless Book

❖❖❖❖