IAT-2 2025

i) Explain IPSec protocol in detail. Also, write about the applications and advantages of IPSec

IPSec (Internet Protocol Security) is a **suite of protocols** used to **secure Internet Protocol (IP) communications** by **authenticating and encrypting** each IP packet in a data stream.

It works at the **network layer (Layer 3)** of the OSI model and is used to **secure VPNs, protect data in transit**, and ensure privacy, integrity, and authenticity.

🔧 Key Components of IPSec

1. Protocols Used:

- **AH (Authentication Header)**
  - Provides **data integrity and authentication**
  - Does **not** provide encryption (no confidentiality)
- **ESP (Encapsulating Security Payload)**
  - Provides **encryption**, **integrity**, and **authentication**
  - Can work with or without encryption

2. Security Modes:

- **Transport Mode:**
  - Encrypts only the **payload (data)** of the IP packet
  - Used for **end-to-end** communication (host to host)
- **Tunnel Mode:**
  - Encrypts the **entire IP packet** and adds a new IP header
  - Used for **network-to-network** or **host-to-network** communication (like in VPNs)

3. Key Management Protocols:

- **IKE (Internet Key Exchange):**
  - Used for **negotiating security associations (SAs)** and **managing keys**
  - Ensures secure key exchange between communicating parties

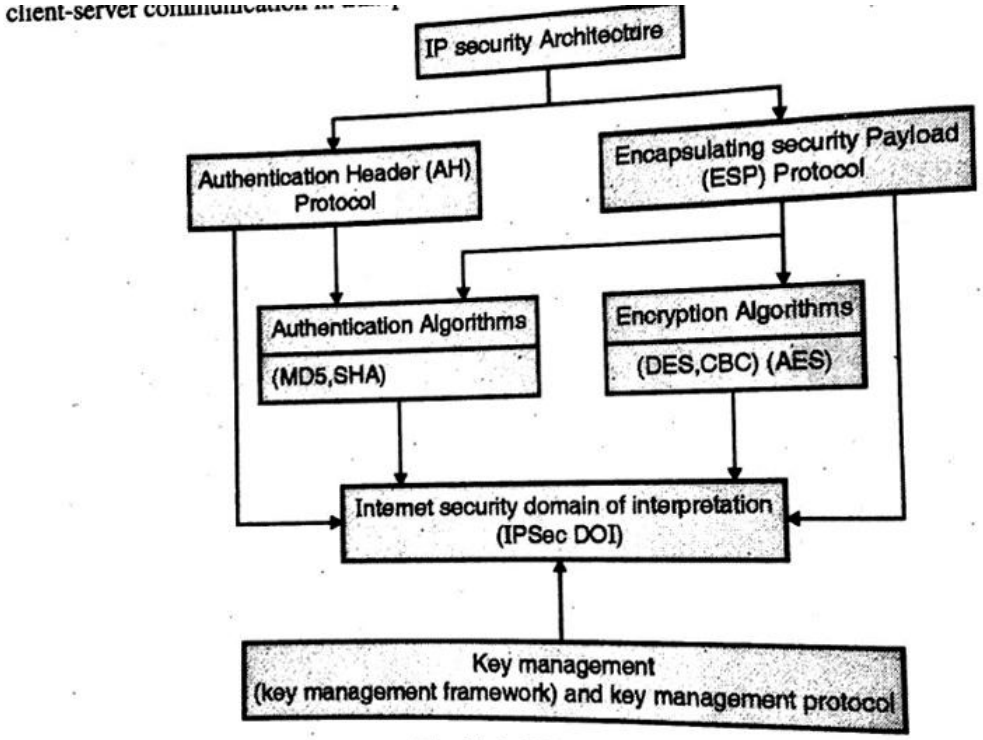


Fig.11.1: IPSec architecture

📌 Applications of IPSec

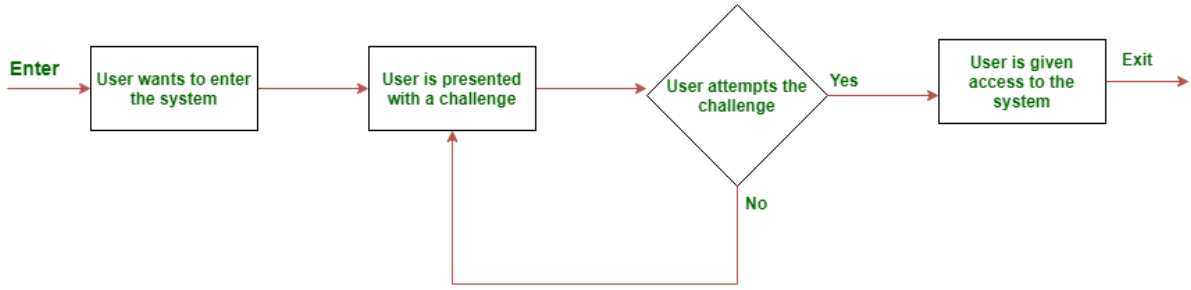
1. **Virtual Private Networks (VPNs)**:
    - Used to create secure site-to-site or remote access VPNs.
    - Ensures secure communication over public networks.
2. **Secure Remote Access**:
    - Employees can securely access enterprise resources from outside.
3. **Data Protection in Enterprise Networks**:
    - Safeguards sensitive data transferred within corporate networks.
4. **Secure Communication between Branch Offices**:
    - Ensures safe data exchange between geographically separated locations.
5. **Cloud Communication**:
    - Secures data transfer between on-premises data centers and cloud.

✅ Advantages of IPSec

1. **Strong Security**:
    - Offers end-to-end data encryption and strong authentication.
2. **Transparent to Applications**:
    - Works at the network layer, so applications don't need modification.
3. **Flexible**:
    - Can be used in both IPv4 and IPv6 networks.
4. **Resistant to Attacks**:
    - Protects against IP spoofing, replay attacks, and packet tampering.
5. **Scalable**:
    - Suitable for small to large networks; can be implemented in routers, firewalls, etc.
6. **Standardized Protocol**:
    - Widely supported and standardized by IETF.

ii) Describe challenge-response based authentication

Ans: They are a group of protocols in which one side presents a challenge (to be answered) and the other side must present a correct answer (to be checked/validated) to the challenge in order to get authenticated.



Types of Challenges:

- *Static*: Fixed questions with consistent answers, such as security questions.
- *Dynamic*: Randomly generated challenges requiring real-time computation, enhancing security.

Common Implementations:

- **CAPTCHA**: Distinguishes humans from bots by presenting tasks like identifying objects in images.
- **CHAP (Challenge-Handshake Authentication Protocol)**: Used in network protocols to periodically verify identities during a session.

iii) Give detailed explanation on entity authentication with its types and example.

Entity authentication is the process of verifying the identity of a user, device, or system that wants to access a network, system, or service. The goal is to ensure that the entity trying to gain access is actually who or what it claims to be.

Entity authentication is a **critical component of security** in computer systems, networks, and communications. It protects against **unauthorized access, impersonation, and various attacks** like spoofing, man-in-the-middle, and replay attacks.

Types of Entity Authentication

Entity authentication methods can broadly be classified into the following types:

1. Knowledge-Based Authentication ("Something you know")

This method relies on information that only the entity should know.

Examples:

- Passwords
- PINs
- Answers to security questions

2. Possession-Based Authentication ("Something you have")

This method is based on something the user possesses.

Examples:

- Security tokens
- Smart cards
- Mobile phones (for OTPs)

3. Inherence-Based Authentication ("Something you are")

This involves biometric authentication based on unique physical characteristics.

Examples:

- Fingerprint scans
- Facial recognition

- Iris/retina scans
- Voice recognition

4. Location-Based Authentication ("Somewhere you are")

This uses the geographic location of the entity to verify identity.

Examples:

- IP address or GPS location
- Authentication only allowed within a building or country

5. Behavioral-Based Authentication ("Something you do")

This type of authentication relies on unique behavioral patterns.

Examples:

- Typing rhythm (keystroke dynamics)
- Mouse movements
- Gait recognition (in advanced systems)

Two-Factor and Multi-Factor Authentication (2FA & MFA)

Entity authentication becomes **stronger** when **two or more types** of authentication factors are combined.

Example of 2FA:

- ATM: Insert your card (**something you have**) + enter PIN (**something you know**)

Example of MFA:

- Login to a website with password (**something you know**) + OTP sent to phone (**something you have**) + fingerprint (**something you are**)

iv) Describe various types of firewalls with example.

v) Discuss various attacks on digital signatures and the methods by which they can be overcome?

Ans: **Types of Attacks**

1. **Key-Only Attack**: The attacker only has access to the public key and attempts to forge a signature.
2. **Known Message Attack**: The attacker has access to a set of messages and their corresponding signatures, which they use to analyze and potentially forge new signatures.
3. **Chosen Message Attack**:
    - **Generic Chosen Message Attack**: The attacker selects messages before knowing the public key and obtains valid signatures for these messages.
    - **Directed Chosen Message Attack**: The attacker selects messages after knowing the public key but before obtaining signatures.
4. **Universal Signature Forgery (USF)**: The attacker manipulates the signature object by adding invalid content or removing references to the signature object.

Methods to Overcome Attacks

1. **Strong Cryptographic Algorithms**: Use robust algorithms like RSA or ECC to make it computationally infeasible for attackers to forge signatures.
2. **Secure Key Management**: Protect private keys with strong encryption and store them securely to prevent unauthorized access.
3. **Public Key Infrastructure (PKI)**: Implement PKI to authenticate users and devices, ensuring secure digital interactions.
4. **Regular Updates**: Update cryptographic protocols and software to address vulnerabilities and enhance security.
5. **Intrusion Detection Systems**: Deploy systems to monitor and detect unauthorized activities related to digital signatures.

vi) What are the different components of an Intrusion Detection System?

Ans: An Intrusion Detection System (IDS) is designed to monitor network or system activities for malicious actions or policy violations. The key components of an IDS typically include:

1. **Sensors:** Monitor network traffic and system activities for suspicious behavior.
2. **Analyzers:** Evaluate collected data against known threat signatures or abnormal patterns.
3. **Database:** Stores event information and IDS configurations.
4. **Alert System:** Notifies administrators of potential security breaches.
5. **User Interface:** Allows administrators to configure and control the IDS.

These components collaborate to detect and alert administrators to potential intrusions, enhancing organizational security.

vii) Explain Boot Sector Virus in detail.

A **boot sector virus** is a type of computer virus that infects the **boot sector** of a storage device like a hard disk, floppy disk, or USB drive. The boot sector is a special part of the disk that contains the code needed to start (boot) the operating system. By targeting this area, the virus can gain control of the system early in the boot process—**before the operating system even loads**.

This makes boot sector viruses particularly **dangerous** and **hard to detect or remove**, especially in older systems.

How Does a Boot Sector Virus Work?

1. When a system is powered on, the **BIOS** (Basic Input/Output System) looks for the **boot sector** on the disk to load the OS.
2. If the boot sector is infected, the virus is executed **before** the operating system.
3. The virus may load itself into memory and remain **resident**.
4. Once in memory, it can:
    a. Infect other disks inserted into the computer.

b. Load the original boot sector code to avoid detection.
c. Spread to other systems via infected storage media.

🔁 Spreading Mechanism

Boot sector viruses spread by copying themselves into the **boot sector of clean disks** when inserted into an infected system. If someone then boots another computer from that disk, the virus spreads to the new system.

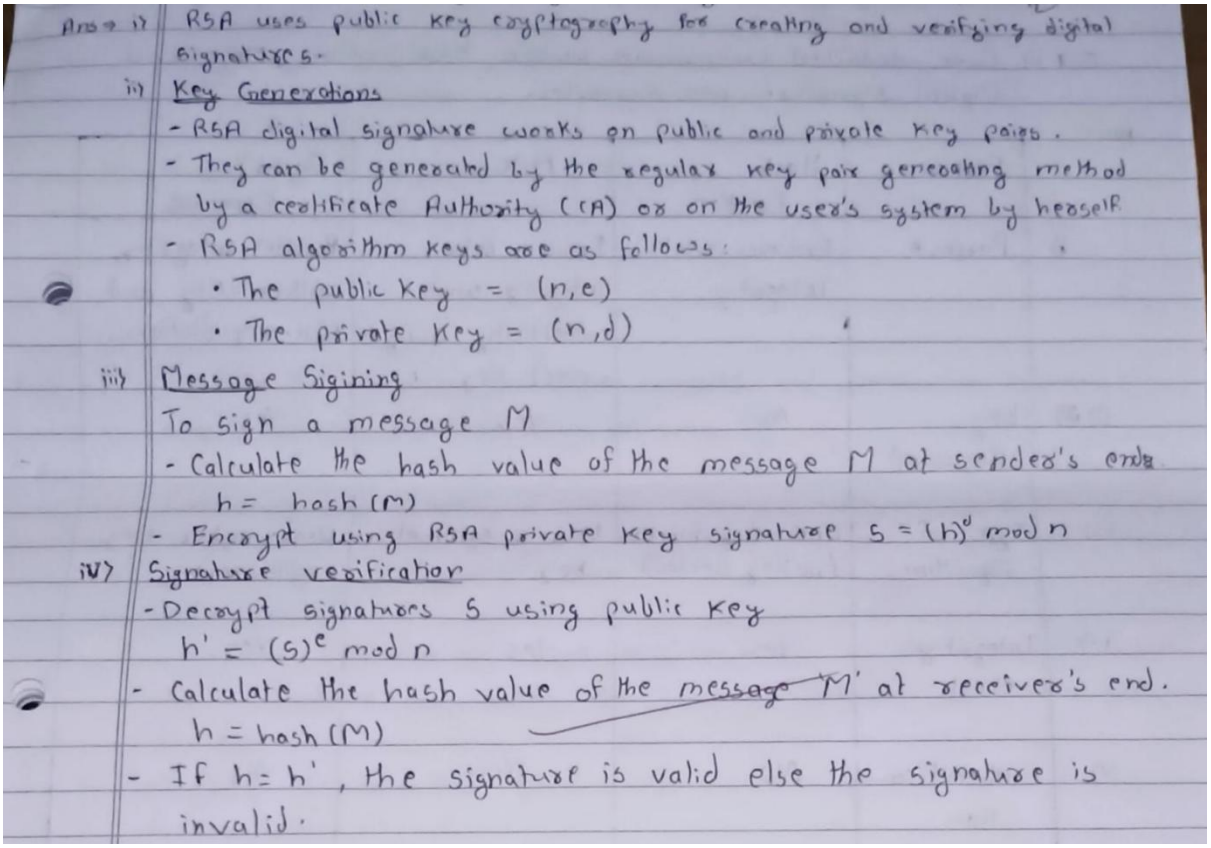
📌 Examples of Boot Sector Viruses

1. **Brain** (1986) – The **first PC boot sector virus**; originated in Pakistan and spread through infected floppy disks.
2. **Stoned** – Displayed the message "Your PC is now Stoned!" on infected systems.
3. **Michelangelo** – Activated on March 6 (Michelangelo's birthday), and attempted to overwrite hard drive data.

viii) What is the purpose of a digital signature in a digital certificate, and how does the RSA-based signature algorithm work?

Ans: A digital signature in a digital certificate serves a crucial purpose:

1) Authentication: The digital signature verifies that the certificate was issued by a trusted Certificate Authority (CA). This ensures that the entity presenting the certificate is legitimate.
2) Integrity: It ensures that the contents of the digital certificate haven't been tampered with after it was issued. If any part of the certificate is altered, the digital signature becomes invalid.
3) Non-repudiation: The signature provides proof of origin, making it difficult for the entity to deny that the certificate belongs to them.



Ans ⇒ i) RSA uses public key cryptography for creating and verifying digital signatures.

ii) Key Generations

- RSA digital signature works on public and private key pairs.
- They can be generated by the regular key pair generating method by a certificate Authority (CA) or on the user's system by herself
- RSA algorithm keys are as follows:
  - The public key = (n, e)
  - The private key = (n, d)

iii) Message Signing

To sign a message M

- Calculate the hash value of the message M at sender's end
  $h = hash(M)$
- Encrypt using RSA private key signature $S = (h)^d \bmod n$

iv) Signature verification

- Decrypt signatures S using public key
  $h' = (S)^e \bmod n$
- Calculate the hash value of the ~~message~~ M' at receiver's end.
  $h = hash(M)$
- If $h = h'$, the signature is valid else the signature is invalid.

ix) Explain the handshake protocol in SSL?

3. SSL Handshake Protocol

✍ ***Definition :*** *The cryptographic parameters of the session state are produced by the SSL handshake protocol.*

- When an SSL client and the server first start communicating, they need to agree upon certain parameters. There are also several steps that need to be carried out to establish a secure session. At a high level, the following four steps are carried out.

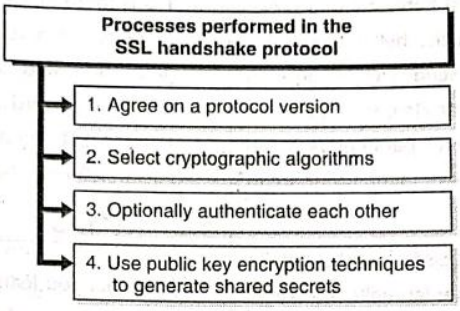


Fig. 6.2.5

- The steps can be detailed in the following simplistic handshake process diagram.

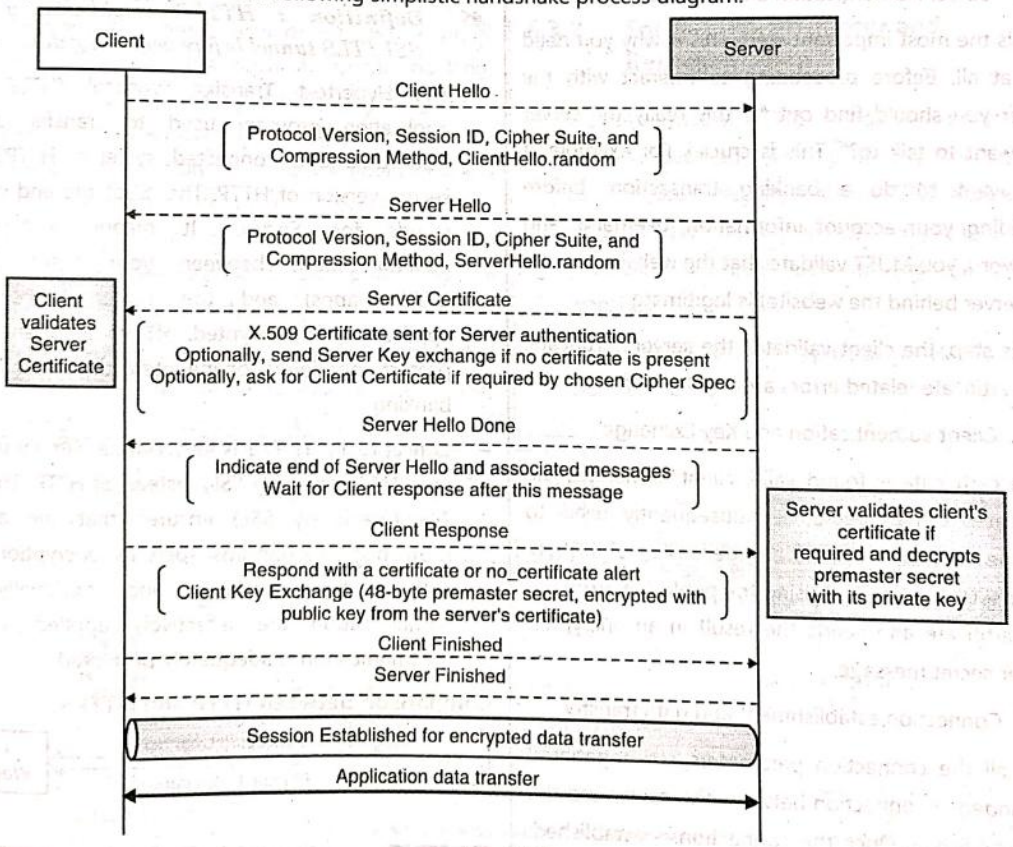


Fig. 6.2.6

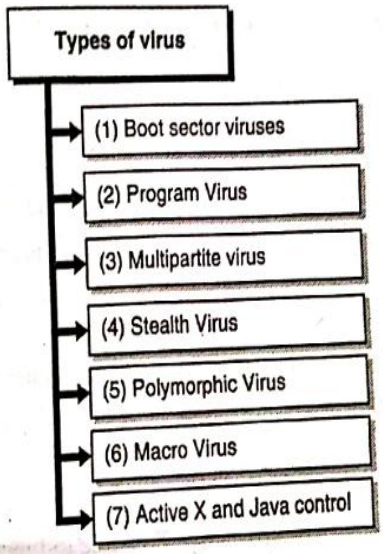
🧠 Real-World Example

When you open a website like https://www.bank.com:

- Your browser performs an **SSL handshake** with the server.
- After successful validation, you'll see a 🔒 **padlock icon** in the address bar, indicating a secure connection.

x) What are the different types of viruses and worms? How do they propagate?

Types of Virus



13.1.4(B) Types of Computer Worms

1. **E-mail worms**

   It spreads through infected email message of any infected websites.

2. **Instant messaging worms**

   It spreads by sending link to contact list of instant messaging application.

3. **Internet worm**

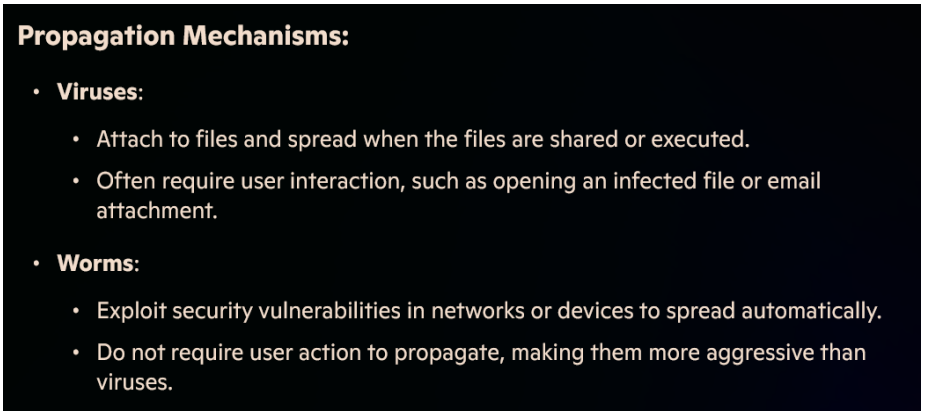
   It scans all network resources which are available and system. If it found vulnerable, i will take advantage and gain access.

4. **IRC (Internet Relay Chat) worms**

   It places a copy of itself through link in infected websites.

5. **File sharing Network worms**

   It places a copy of itself in a folder which is sharable and spread via P2P network.



xi) Discuss the need for Email Security. How does PGP achieve confidentiality and authentication in emails?

Email is one of the most commonly used communication tools, but it is **inherently insecure**. Messages can be intercepted, altered, or impersonated during transmission. Hence, **email security** is essential to:

🔐 Why Email Security is Needed:

1. **Confidentiality** – Prevent others from reading your emails (protect sensitive info).
2. **Authentication** – Ensure the email is actually from the sender it claims to be.
3. **Integrity** – Ensure the message was not changed during transmission.
4. **Non-repudiation** – Prevent sender from denying that they sent the email.
5. **Protection from attacks** – Block spam, phishing, malware, and spoofing.

➜ 1. PGP Authentication

1. Ramesh has (private/public) key pair *(Rd/Re)* and he wants to send a digitally signed message *m* to Suresh.
2. Ramesh hashes the message using SHA-1 to obtain *SHA(m)*.
3. Ramesh encrypts the hash using his private key *Rd* to obtain ciphertext *c* given by

   $$c = \text{encrypt}_{Rd}(\text{SHA}(m))$$

4. Ramesh sends the pair *(m,c)to Suresh*
5. Suresh receives *(m,c)* and decrypts *c* using Ramesh's public key *Re* to obtain signature S

   $$S = \text{decrypt}_{Re}(c)$$

6. He computes the hash of *m* using SHA-1 and if this hash value is equal to S then the message is authenticated.

   Suresh is sure that the message is correct and that came from Ramesh. Furthermore Ramesh cannot later deny sending the message since only Ramesh has access to his private key *Rd* which works with respective public key *Rd*.

➜ 2. PGP Confidentiality

1. Ramesh wishes to send Suresh a confidential message *m*.
2. Ramesh generates a random session key *k* for a symmetric cryptosystem.
3. Ramesh encrypts *k* using Suresh's public key *Be* to get.

$$k' = \text{encrypt}_{Be}(k)$$

4. Ramesh encrypts the message *m* with the session key *k* to get ciphertext *c*

   $$c = \text{encrypt}_{k}(m)$$

5. Ramesh sends Suresh the values *(k',c)*
6. Suresh receives the values *(k',c)* and decrypts *k'* using his private key B_d to obtain *k*.

   $$k = \text{decrypt}_{Bd}(k')$$

7. Suresh uses the session key *k* to decrypt the ciphertext *c* and recover the message *m*

   $$m = \text{decrypt}_{k}(c)$$

Public and symmetric key cryptosystems are combined in this way to provide security for key exchange and then efficiency for encryption. The session key *k* is used only to encrypt message *m* and is not stored for any length of time.

... and Confidentiality

xii) Explain the different types of SQL injections? Discuss mitigation techniques for SQL injection.

SQL Injection is a web security vulnerability that allows attackers to **interfere with the queries** an application sends to its database. It can lead to data theft, modification, or even complete system compromise.

🔍 Types of SQL Injection

1. Classic (In-band) SQL Injection

a. Error-based SQLi

- Uses database error messages to reveal information.
- Example: `' OR 1=1 --`

b. Union-based SQLi

- Uses UNION SQL operator to combine results from multiple queries.
- Example: `' UNION SELECT username, password FROM users --`

2. Blind SQL Injection

No visible error messages, attacker infers responses based on app behavior.

a. Boolean-based

- Sends queries that return TRUE or FALSE, observing responses.
- Example: `' AND 1=1 --` (true), `' AND 1=2 --` (false)

b. Time-based

- Uses `SLEEP()` or `WAITFOR DELAY` to measure time-based responses.
- Example: `' IF(1=1, SLEEP(5), 0) --`

3. Out-of-Band SQL Injection

- Sends data to an attacker's server (e.g., via DNS or HTTP requests).
- Used when in-band and blind SQLi don't work.

4. Second-order SQL Injection

- Malicious input is stored in the database and executed later in a different query context.
- Example: Attacker registers with a payload in their username; later, an admin panel query is exploited.

Mitigation Techniques for SQL Injection

☑ 1. Use Prepared Statements (Parameterized Queries)

- Prevents direct injection by separating SQL logic from data.

☑ 2. Use Stored Procedures

- Helps limit the execution of dynamic queries.

☑ 3. Input Validation & Sanitization

- Allow only valid data (whitelisting) for user inputs.
- Reject or escape dangerous characters.

☑ 4. Limit Database Privileges

- The web application should connect with **limited access**, not as a root user.

☑ 5. Error Handling

- Do not expose detailed database errors to users.
- Use generic error messages.

☑ 6. Use Web Application Firewalls (WAF)

- Detect and block known SQL injection patterns.

☑ 7. Regular Security Testing

- Perform code reviews, penetration testing, and use automated vulnerability scanners.

xiii) Explain ARP spoofing in detail with example.

Ans:

ARP Spoofing is a type of cyber attack where hackers intercept communications between network devices by sending falsified ARP (Address Resolution Protocol) messages.

This allows them to associate their MAC address with the IP address of another device on the network, effectively redirecting traffic meant for that device to themselves.

How ARP Spoofing Works

a) Scanning: Hackers use ARP spoofing tools to scan the IP and MAC addresses of hosts on the network.

b) Selection and Launching: They select their target and send ARP packets over the local network containing the hacker's MAC address and the target's IP address.
c) Accessing: Once the ARP cache on the host is corrupted, data meant for the victim is sent to the hacker instead. This allows hackers to steal data or launch other attacks.

Types of ARP Spoofing Attacks

- Man-in-the-Middle (MitM): Hackers intercept communications between devices to steal or modify information.
- Session Hijacking: Hackers extract session IDs or gain unauthorized access to private systems and data.
- Denial-of-Service (DoS): Hackers overload the target's network by linking their MAC address with multiple IP addresses, causing network congestion[1].

Preventive Measures

- Cryptographic Network Protocols: Using encrypted communication protocols like TLS, HTTPS, and SSH can reduce the risk of ARP spoofing.
- Packet Filtering: Packet filters can protect the network from malicious packets and suspicious IP addresses.
- Virtual Private Network (VPN): VPNs are highly effective in preventing ARP spoofing attacks.
- ARP Spoofing Detection Software: These tools help in detecting ARP spoofing by inspecting and certifying data before transmission

xiv) Explain DOS and DDOS attack? Explain how it is launched.

Denial of Service (DoS) Attack:

1. **Definition**: A DoS attack is a malicious attempt to disrupt the normal functioning of a server, service, or network by overwhelming it with a flood of illegitimate requests. This

makes the system slow, unresponsive, or completely inaccessible to legitimate users.

2. **How It Works**:
    - The attacker uses a single computer to send a massive number of requests to the target.
    - These requests consume the target's resources (e.g., bandwidth, memory, or CPU), causing it to crash or become unavailable.
3. **Types of DoS Attacks**:
    - **Buffer Overflow Attack**: Exploits vulnerabilities in memory allocation to crash the system.
    - **Ping Flood**: Overwhelms the target with ICMP (ping) packets.
    - **SYN Flood**: Exploits the TCP handshake process to exhaust server resources.
4. **Launching a DoS Attack:**
    - The attacker uses a single machine to send a flood of requests to the target.
    - Tools like LOIC (Low Orbit Ion Cannon) or HOIC (High Orbit Ion Cannon) are often used.

Distributed Denial of Service (DDoS) Attack:

1. **Definition**: A DDoS attack is a more advanced form of DoS attack, where multiple compromised devices (a botnet) are used to flood the target with traffic.
2. **How It Works**:
    - The attacker infects numerous devices (computers, IoT devices, etc.) with malware, turning them into bots.
    - These bots are controlled remotely to send a coordinated flood of requests to the target.
    - The sheer volume of traffic overwhelms the target, making it difficult to distinguish between legitimate and malicious requests.
3. **Types of DDoS Attacks**:

- **Volume-Based Attacks**: Flood the target with massive amounts of data to saturate bandwidth.
- **Protocol Attacks**: Exploit weaknesses in network protocols (e.g., SYN floods).
- **Application Layer Attacks**: Target specific applications or services, such as HTTP or DNS.

4. **Launching a DDoS Attack:**

   - The attacker creates a botnet by infecting devices with malware.
   - The botnet is then used to send a coordinated flood of traffic to the target.
   - Tools like Mirai or specialized DDoS services (illegal) are used to orchestrate the attack.

xv) How is security achieved in transport and tunnel modes of IPSec? What are security associations?

xvi) What is ICMP flood attack? Explain in detail.

Ans: An ICMP flood attack, also known as a ping flood, is a type of Denial of Service (DoS) attack. It aims to overwhelm a target system by sending a massive number of ICMP Echo Request packets (ping requests).

Mechanism:

- The attacker sends a large volume of ICMP Echo Request packets to the target system.
- The target system responds to each request with an ICMP Echo Reply packet.
- This process consumes significant network bandwidth and system resources, leading to degraded performance or complete unavailability.

Impact:

- The attack can disrupt legitimate network traffic, making the system inaccessible to users.
- It can also cause network congestion and slowdowns.

Types of ICMP Flood Attacks:

1) Targeted Local Disclosed Ping Flood:
    - Targets a specific computer within a local network.
    - Requires the attacker to know the IP address of the target.
    - Can result in the target computer being knocked offline.
2) Router Disclosed Ping Flood:
    - Targets routers to disrupt communication between devices on a network.
    - Overwhelms the router with ICMP requests, affecting all connected devices.
3) Blind Ping Flood:
    - Uses external tools to discover the IP address of the target before launching the attack.

Mitigation Strategies:

1) Disable ICMP Functionality: Prevent the system from processing ICMP requests.
2) Rate Limiting: Limit the number of ICMP packets the system can process.
3) Firewall Rules: Block excessive ICMP traffic at the network perimeter.
4) Network Monitoring: Detect and respond to unusual traffic patterns.

xvii) Discuss layer wise TCP/IP vulnerabilities.

xviii) What is a buffer overflow in C? Why gets () is bad?

Ans:

1. Buffer overflows

Q. 13.1.4 What is buffer overflow in software security ? (Ref. sec. 13.1.2(1))

- Attacker can insert malicious data values / instruction codes into overflow space.
- Array bound checking is not performed by C compiler, pointer limits cannot be defined as well.
- **Example :** int B[15];
- Here the array bound is (0 to 14). i.e. B[0].....B[14].

- If anything inserted after that bound then the adjacent data is overwritten.
- Attacker can overwrite users data, changes users instruction, overwrite OS data, changes OS instructions. Thus can get complete control of a program or OS. This is also known as aliasing.

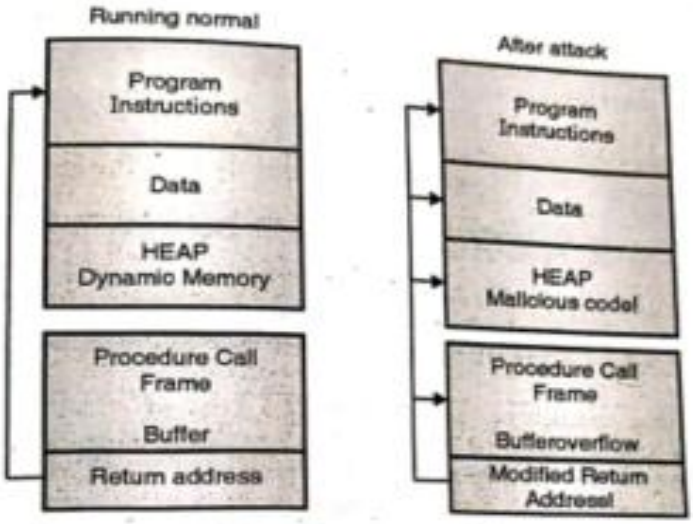


Fig. 13.1.1 : Buffer overflow attack

- As shown in Fig. 13.1.1 attacker changes the return address and thus can transfer the control of the program.

Why gets() is Bad:

1. **No Bounds Checking**: It reads input until a newline or EOF, but doesn't limit input size, leading to buffer overflows.
2. **Security Risk**: Makes programs highly vulnerable to attacks (e.g., arbitrary code execution).

3. **Deprecation**: Removed in modern C standards like C11.
4. **Alternatives**: Use fgets() or scanf() with specified limits for safer input handling.

xix) Differentiate between SQL injection and buffer overflow.

Ans:

SQL Injection vs Buffer Overflow

Aspect	SQL Injection	Buffer Overflow
Definition	Injecting malicious SQL code into queries	Writing more data than a buffer can hold, overflowing into adjacent memory
Target	Databases through web applications	Application memory (stack/heap)
Main Goal	Unauthorized access, data theft, or manipulation	Crashing programs or executing arbitrary code
Attack Vector	User input fields (login forms, URLs, search bars)	Inputs causing buffer overflow (e.g., large strings in C/C++)
Languages Affected	SQL	C, C++, Assembly (low-level languages)
Vulnerability Cause	Improper input validation in SQL queries	Lack of bounds checking in memory operations
Example Attack	`' OR '1'='1 --` in login field	Entering 300 characters in a buffer designed for 100
Security Impact	Data leakage, unauthorized database access	Program crash, system compromise, code execution
Mitigation Techniques	Use prepared statements, input sanitization	Use safe functions, bounds checking, memory-safe languages