UNIT - VI System Security

System Security

- Intruders , Viruses and Relate Threats ,Firewall Design Principles.
- Comprehensive Examples Using Available Software Platforms/Case Tools, Configuration Management

System Security refers to the protection of computer systems, networks, and data from **unauthorized access, attacks, damage, and misuse**.

Objectives of System Security

1. **Confidentiality** – Information is accessible only to authorized users
2. **Integrity** – Data remains accurate and is not altered without permission
3. **Availability** – Systems and data are available when needed

Vulnerability

Vulnerability is a **weakness or flaw** in a system, network, software, or process that can be **exploited by attackers** to gain unauthorized access, cause damage, or steal information.

A vulnerability is a security weakness in a system that can be exploited by a threat or attacker.

Vulnerabilities may result from the following **elements**:

- **Hackers**
  Hackers exploit system weaknesses such as poor passwords, software bugs, or misconfigurations to gain unauthorized access.

- **Radiation**
  Electromagnetic radiation or environmental factors can cause data corruption or hardware damage, leading to system vulnerabilities.

- **Malfunctioning**
  Hardware or software failures, bugs, or improper functioning of system components can create security loopholes.

- **Information on the Network**
  Sensitive information transmitted over unsecured networks can be intercepted, modified, or misused by attackers.

Internet Vulnerabilities are **weaknesses in internet-connected systems or networks** that can be exploited by attackers to steal data, disrupt services, or gain unauthorized access.

Common Internet Vulnerabilities

1. **Weak Passwords** – Easy-to-guess or reused passwords
2. **Unpatched Software** – Outdated browsers, OS, or applications
3. **Unsecured Wi-Fi Networks** – Open or poorly encrypted networks
4. **Phishing Attacks** – Fake emails or websites stealing credentials
5. **Malware** – Viruses, spyware, ransomware spread via the internet

INTRUDERS

Intruders are individuals or programs that **attempt to gain unauthorized access** to a computer system or network with the intention of **stealing data, causing damage, or misusing resources**.

Types of Intruders

1. **Masquerader** – An intruder who pretends to be an authorized user
2. **Misfeasor** – A legitimate user who misuses granted privileges
3. **Clandestine User** – An intruder who gains supervisory control and hides activities

Need for Intrusion Monitoring and Detection

Intrusion monitoring and detection are necessary to **identify unauthorized access and security attacks** on a system or network at an early stage.

Need / Importance

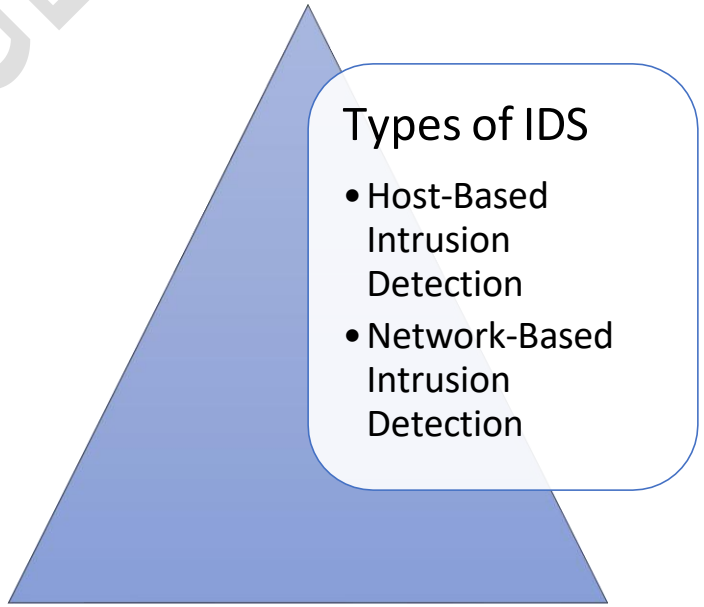
- **Detect Unauthorized Access**
  Helps identify intruders who try to access the system illegally.
- **Prevent Data Loss**
  Protects sensitive and confidential information from theft or misuse.
- **Early Warning of Attacks**
  Detects attacks like hacking, malware, and insider misuse in real time.
- **System Integrity Protection**
  Ensures data and system resources are not modified or damaged.
- **Reduce Damage**
  Quick detection helps limit the impact of security breaches.

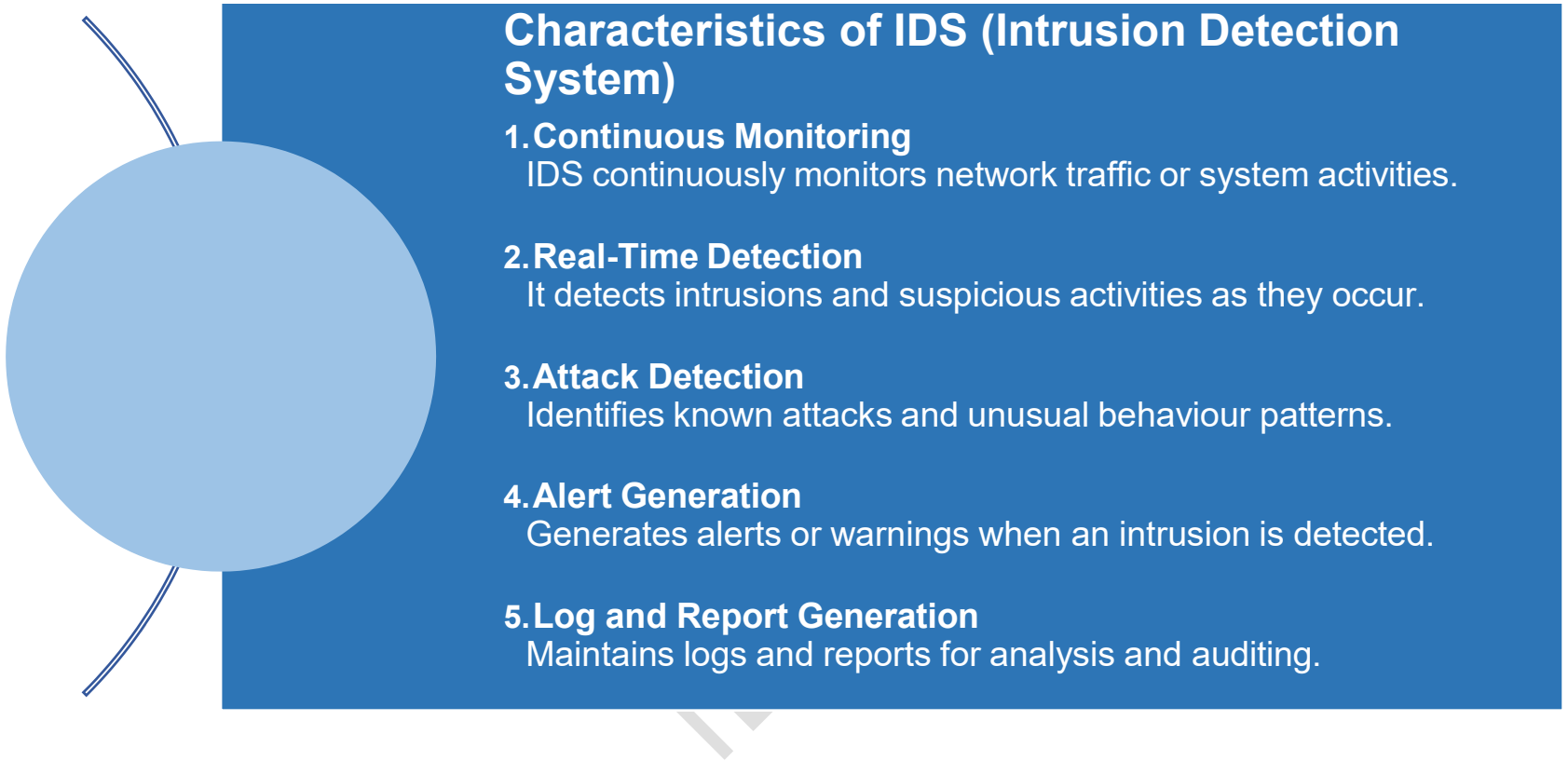
IDS (Intrusion Detection System)

An **Intrusion Detection System (IDS)** is a security mechanism that **monitors network traffic or system activities** to detect **unauthorized access, attacks, or policy violations** and generates alerts.

Types of IDS

1. **Network-based IDS (NIDS)**
Monitors network traffic for suspicious activities.
2. **Host-based IDS (HIDS)**
Monitors system files, logs, and activities on a single host.




Characteristics of IDS (Intrusion Detection System)
1. Continuous Monitoring
IDS continuously monitors network traffic or system activities.
2. Real-Time Detection
It detects intrusions and suspicious activities as they occur.
3. Attack Detection
Identifies known attacks and unusual behaviour patterns.
4. Alert Generation
Generates alerts or warnings when an intrusion is detected.
5. Log and Report Generation
Maintains logs and reports for analysis and auditing.

Audit Records in Intrusion Detection

A **fundamental tool for intrusion detection** is the **audit record**. Audit records provide detailed information about system activities and help in identifying unauthorized or suspicious behavior.

Two plans are used in audit records:

1. Native Audit Records
These are **standard audit logs** generated by the operating system or applications.
•Record normal system and user activities
•Include login/logout, file access, and system events
•Used for general monitoring and security analysis

2. Detection-Specific Audit Records
These are **specially designed audit records** created specifically for intrusion detection.
•Collect only security-relevant data
•Focus on suspicious or abnormal activities
•Reduce log size and improve detection efficiency

Approaches of Intrusion Detection

- Statistical Anomaly Detection
  - Threshold Detection
  - Profile-Based
- Rule-Based Detection
  - Anomaly Detection
  - Penetration Identification

Approaches of Intrusion Detection

- Intrusion detection techniques are broadly classified into the following approaches:

1.Statistical Anomaly Detection

- This approach detects intrusions by comparing current system behavior with normal behavior.

i Threshold Detection

- Uses predefined thresholds for system activities
- Raises an alert when activity exceeds the normal limit
- Example: Excessive failed login attempts

ii Profile-Based Detection

- Maintains profiles of normal user or system behavior
- Compares current activity with stored profiles
- Deviations indicate possible intrusion

2. Rule-Based Detection

This approach uses predefined rules to detect intrusions.

- **1 Anomaly Detection**
  - Detects activities that violate defined rules
  - Identifies abnormal behavior based on rule violations
- **2 Penetration Identification**
  - Identifies known attack patterns and intrusion techniques
  - Matches activities with known penetration signatures

Virus and Related Threats

A **computer virus** is a malicious program that attaches itself to a legitimate program or file and spreads when the infected program is executed. Its main purpose is to **damage data, disrupt system operations, or steal information**.

Types of vulnerabilities

- Malware
- cyber crime
- Cyber terrorism
- spoofing
- phishing and identity
- sniffing
- Denial of service (DOS)

1. Malware

Malware is malicious software designed to damage, disrupt, or gain unauthorized access to systems.

Spyware

- Secretly collects user information
- Monitors activities like keystrokes and browsing
- Sends data to attackers without user knowledge

Adware

- Displays unwanted advertisements
- Slows down system performance
- May track user behavior

2. Cyber Crime
•Illegal activities performed using computers or the internet
•Examples: hacking, online fraud, data theft

3. Cyber Terrorism
•Use of the internet to carry out large-scale attacks
•Targets critical infrastructure like banking, power, and defense systems

4. Spoofing
•Pretending to be a trusted source
•Used to steal sensitive information
•Examples: email spoofing, IP spoofing

5. Phishing and Identity Theft
•Fake emails or websites trick users
•Steals passwords, bank details, and personal data

6. Sniffing
•Unauthorized capturing of network data
•Used to steal login credentials and confidential information

7. Denial of Service (DoS)
•Attacks that make services unavailable
•Overloads the server or network with traffic

Types of Malicious Software (Malware)

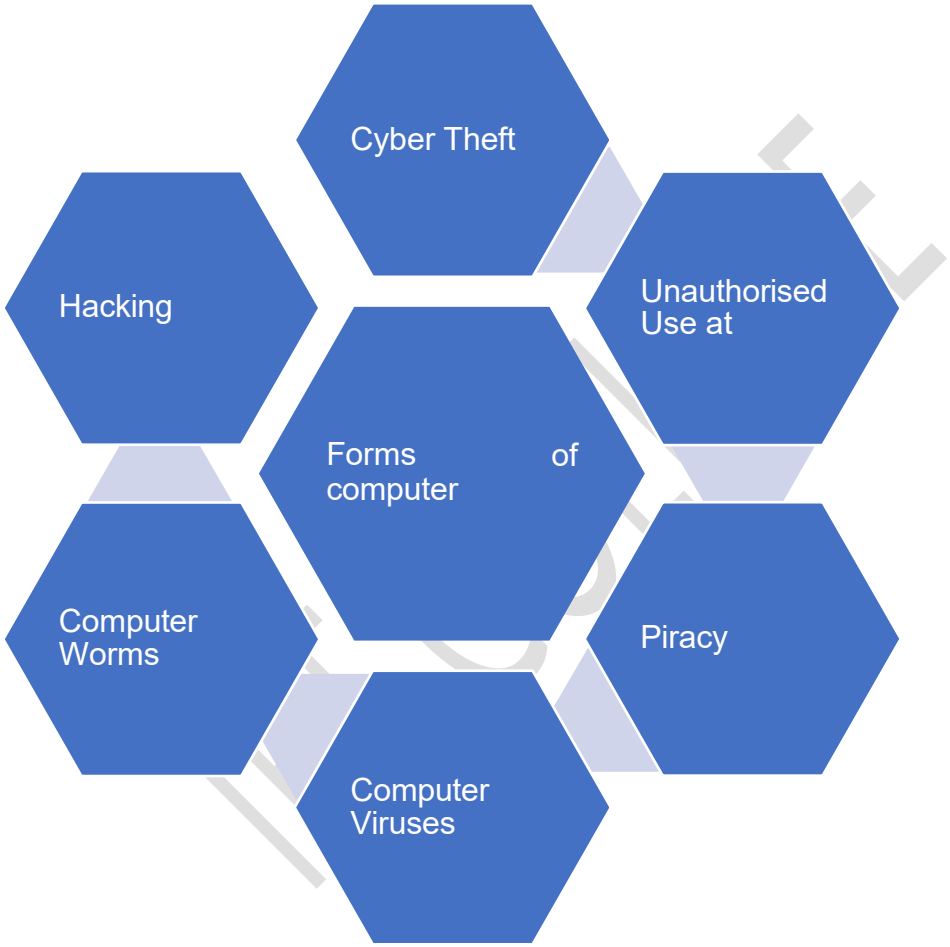
- **Virus**
  - Attaches to files and programs
  - Spreads when infected files are executed
- **Worm**
  - Self-replicates without user action
  - Spreads through networks
- **Trojan Horse**
  - Appears as legitimate software
  - Creates backdoors for attackers
- **Spyware**
  - Secretly monitors user activities
  - Steals personal and confidential information

5. **Adware**
- Displays unwanted advertisements
- May track user behavior

6. **Ransomware**
- Encrypts data and demands ransom
- Blocks access to files or systems

7. **Rootkit**
- Hides malicious activities
- Gives attackers administrative control

8. **Keylogger**
- Records keystrokes
- Steals passwords and sensitive data


Cyber Theft
Hacking
Unauthorised Use at
Forms of computer
Computer Worms
Piracy
Computer Viruses

Forms of Computer Crimes

1. Cyber Theft
Cyber theft involves **stealing money, data, or digital resources** using computers or the internet.
Example: Stealing bank details and transferring money online.

2. Unauthorized Use
Using a computer system, network, or data **without permission**.
Example: Using someone else's login credentials to access their email or office system.

3. Hacking
Hacking is **gaining unauthorized access** to computer systems or networks.
Example: Breaking into a company server to steal confidential data.

4.Piracy

Piracy is the **illegal copying, distribution, or use of software or digital content**.

Types of Piracy

1. Software Piracy

•Illegal copying or sharing of software

•Example: Using cracked or unlicensed software

2. Piracy of Intellectual Property

•Unauthorized use of copyrighted material

•Example: Movies, music, books, or designs copied without permission

Computer Worms

- Computer worms are **self-replicating malicious programs** that spread without user interaction.

Types of Computer Worms

- **Email Worms** – Spread through email attachments
- **Instant Messaging Worms** – Spread via chat messages
- **Internet Worms** – Exploit network vulnerabilities
- **File-Sharing Network Worms** – Spread through shared files

Computer Viruses
Computer viruses are malicious programs that **attach themselves to files or programs**.

Types of Computer Viruses

1. **Macro Viruses** – Infect documents like Word or Excel files
2. **Encrypted Viruses** – Hide code using encryption
3. **Boot Sector Viruses** – Infect system boot sector
4. **Script Viruses** – Written in scripting languages like JavaScript
5. **Polymorphic Viruses** – Change their code to avoid detection

Difference Between Virus and Worm

Basis	Virus	Worm
Definition	A virus is a malicious program that attaches itself to a file or program	A worm is a standalone malicious program
User Action	Requires user action to spread	Spreads automatically without user action
Dependency	Needs a host file to function	Does not need a host file
Speed of Spread	Spreads slowly	Spreads very fast
Network Use	Limited network impact	Actively spreads through networks
Example	Macro virus, boot sector virus	Email worm, internet worm

Spoofing

Spoofing is a cyber attack in which an attacker **pretends to be a trusted source** to gain unauthorized access, steal data, or spread malware.

Types of Spoofing

- IP Spoofing
- DNS Spoofing
- Email Spoofing
- Website Spoofing

1. IP Spoofing

In IP spoofing, the attacker **fakes the source IP address** to appear as a trusted system.

- Used to bypass security systems

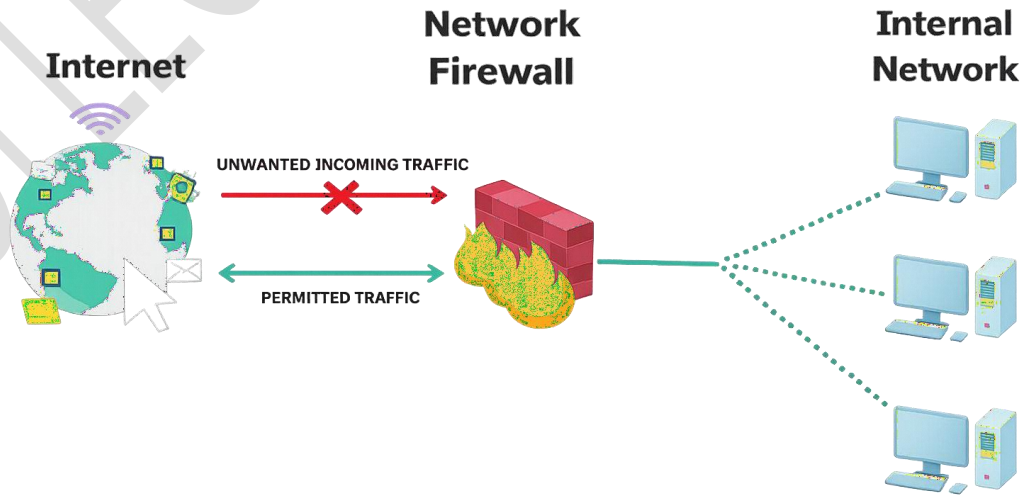
2. DNS Spoofing

In DNS spoofing, the attacker **corrupts DNS records** to redirect users to fake websites.

- Used for phishing and data theft

Firewall Design Principles

Firewall design principles define how a firewall should be **planned and implemented** to protect a network effectively.



Firewall Techniques

1. **Service Control**
Controls which network services (HTTP, FTP, SMTP) are allowed or blocked.

2. **Direction Control**
Controls the direction of traffic (inbound or outbound).

3. **User Control**
Controls access based on user identity.

4. **Behavior Control**
Controls how services are used (e.g., restrict file uploads).

Types of firewalls

- packet-filtering firewall
- Application level firewall
- Screened-host firewall
- Stateful inspection firewall
- Circuit- level firewall

Types of Firewalls

1. Packet-Filtering Firewall

•Filters packets based on IP address, port number, and protocol.

•Works at **Network Layer**.

•Fast but less secure.

2. Application-Level Firewall (Proxy Firewall)

•Filters traffic at the **application layer**.

•Examines application data (HTTP, FTP, SMTP).

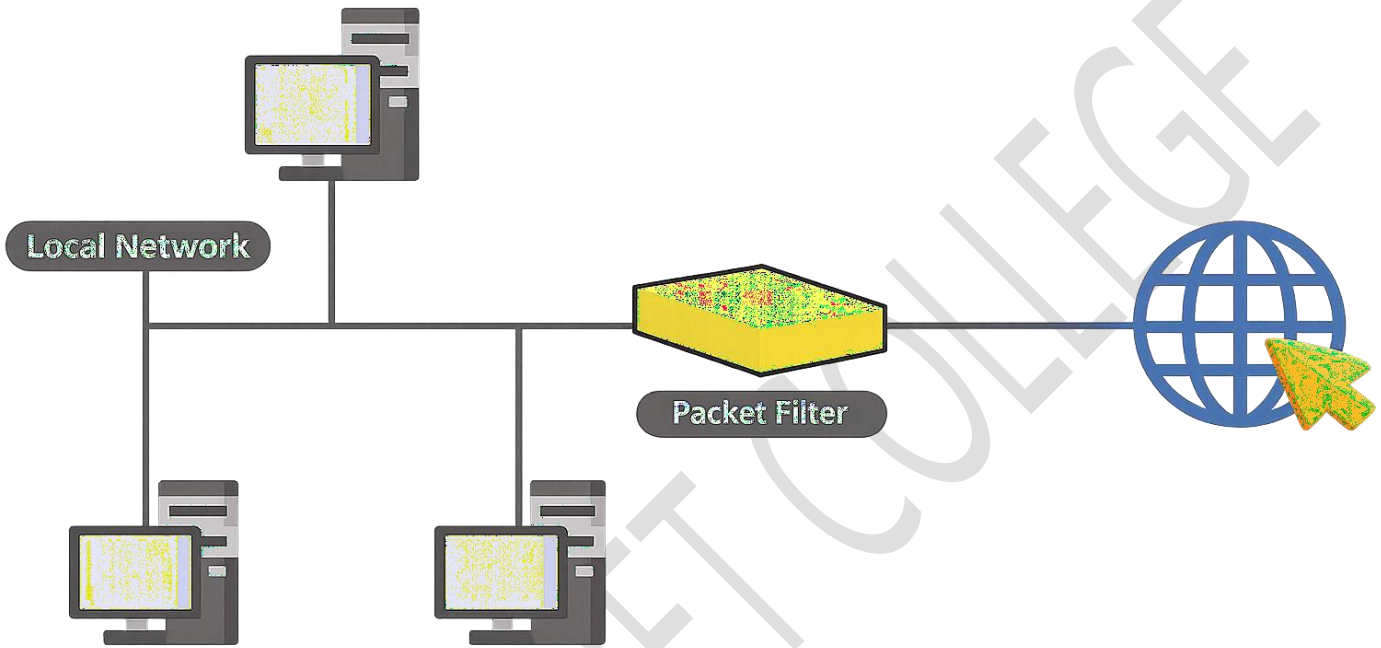
•High security, slower performance.

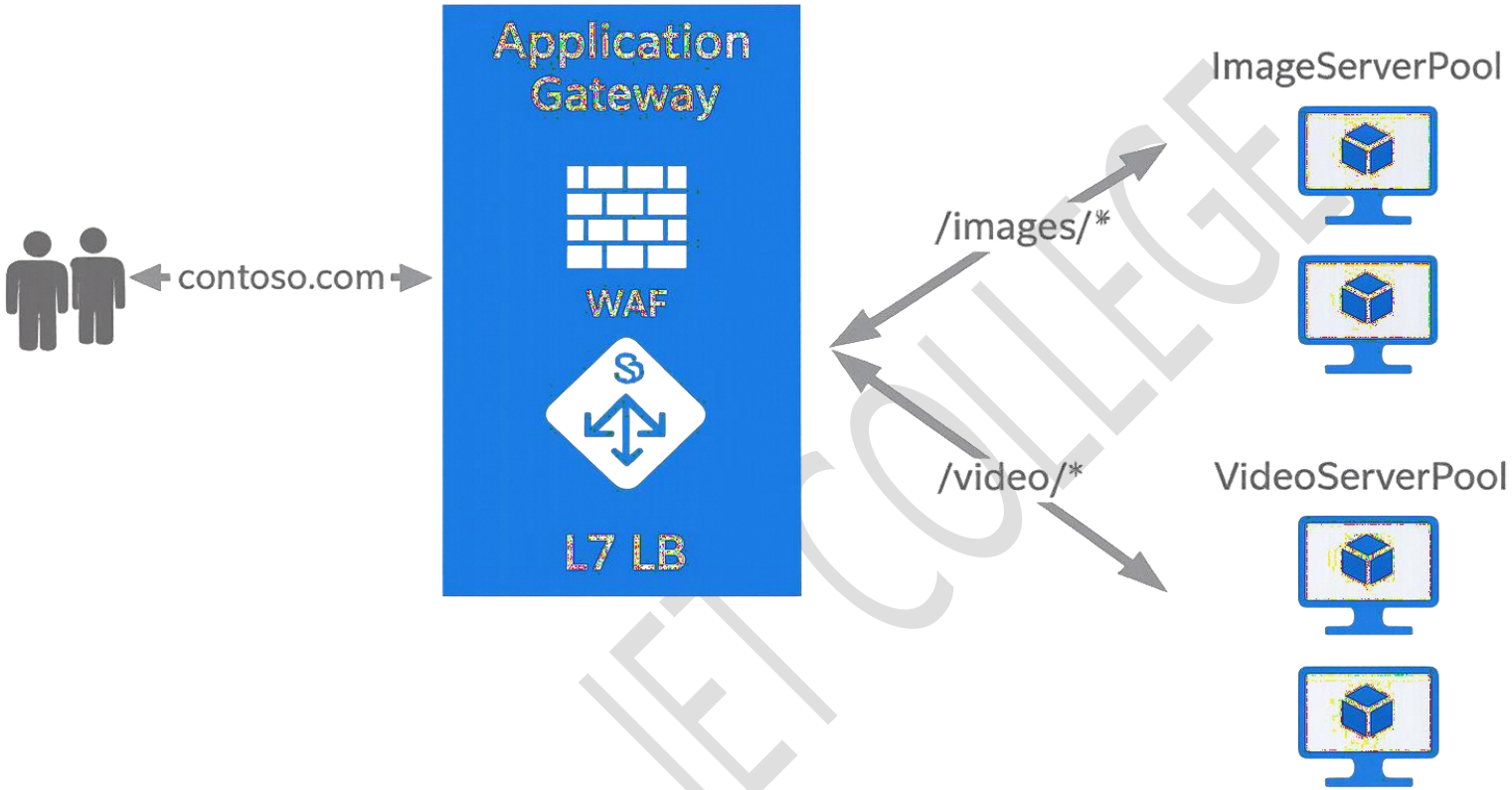
3. Screened-Host Firewall
•Combines **packet-filtering router** and **bastion host**.
•Router filters traffic; bastion host provides secure access.
•Better security than packet filtering alone.

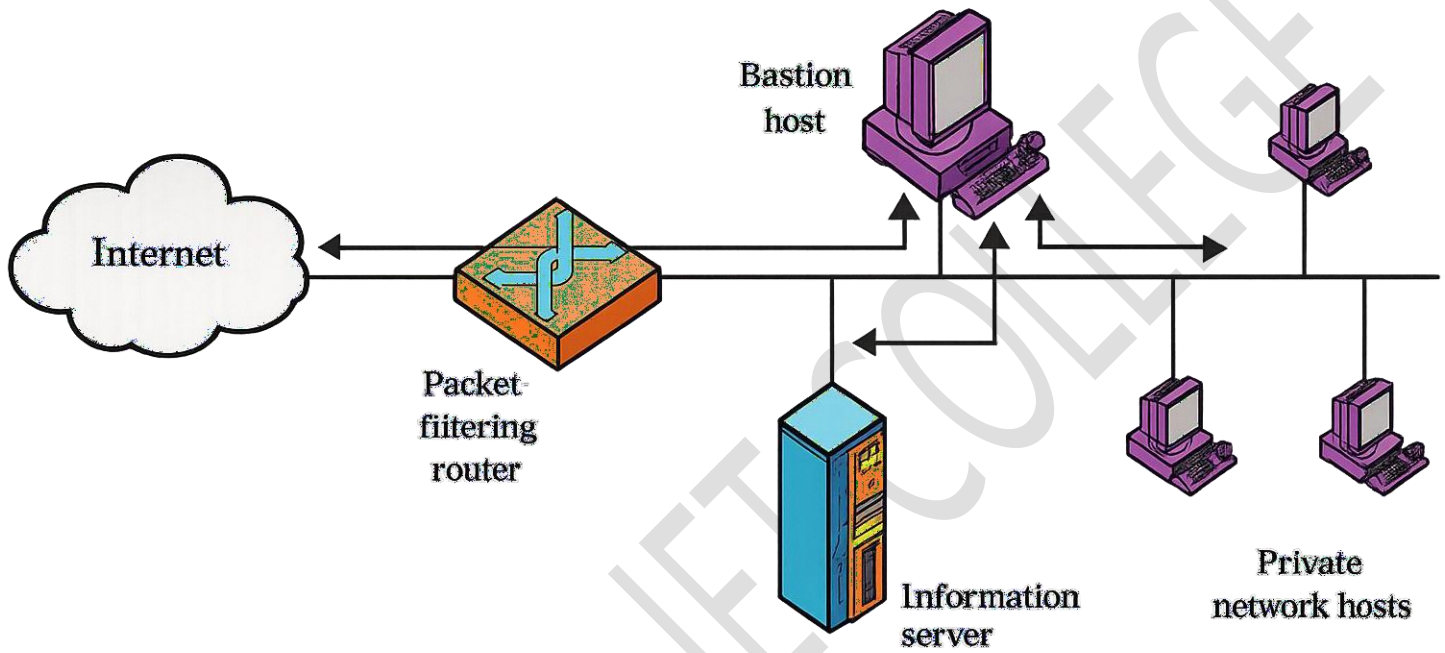
4. Stateful Inspection Firewall
•Monitors the **state of active connections**.
•Allows packets only if they belong to a valid session.
•More secure than packet filtering.

5. Circuit-Level Firewall
•Monitors TCP handshakes and session setup.
•Does not inspect packet content.
•Works at **Session Layer**.

Packet Filtering Firewall




Application Gateway
WAF
L7 LB
contoso.com
/images/*
/video/*
ImageServerPool
VideoServerPool


Bastion
host
Internet
Packet
fiitering
router
Information
server
Private
network hosts

Benefits of Firewalls

1. **Network Security**
   - Protects internal network from unauthorized access.

2. **Access Control**
   - Allows or blocks traffic based on rules (IP, port, protocol).

3. **Prevention of Attacks**
   - Helps block malware, hacking attempts, and intrusions.

4. **Traffic Monitoring**
   - Monitors incoming and outgoing network traffic.

5. **Data Protection**
   - Prevents sensitive data from leaking outside the network.

Limitations of Firewalls

1. **Cannot Stop Internal Attacks**
    - Firewalls cannot protect against attacks from inside the network.

2. **Limited Application-Level Protection**
    - Basic firewalls cannot inspect application data deeply.

3. **Configuration Errors**
    - Incorrect rules can create security loopholes.

4. **Performance Overhead**
    - May slow down network traffic.

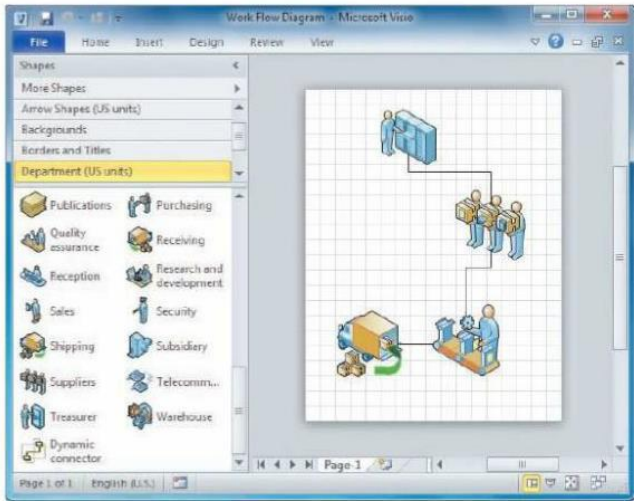
5. **Cost and Maintenance**
    - Advanced firewalls are expensive and need skilled management.

System Development Tools

System development tools help in **planning, designing, developing, testing, and maintaining** an information system.

Systems Development Tools

- Modeling
  - Business model
  - Requirements model
  - Data model
  - Object model
  - Network model
  - Process model



1. Modelling Tools

Modelling tools are used to **represent a system graphically** before actual development.

🔹 Purpose

- Understand system requirements
- Visualize system structure and flow
- Reduce errors before coding

🔹 Common Modelling Techniques

- **Data Flow Diagrams (DFD)**
- **Entity Relationship Diagrams (ERD)**
- **Flowcharts**

2. Prototyping Tools

Meaning:

Prototyping tools are used to **build a working model (prototype)** of the system.

❑ Purpose

- Get early user feedback
- Clarify user requirements
- Reduce development risks

❑ Types of Prototyping

- **Throwaway Prototype** – discarded after requirements are clear
- **Evolutionary Prototype** – continuously improved
- **Rapid Prototyping** – quick design for user review

3. Computer-Aided Software Engineering (CASE) Tools

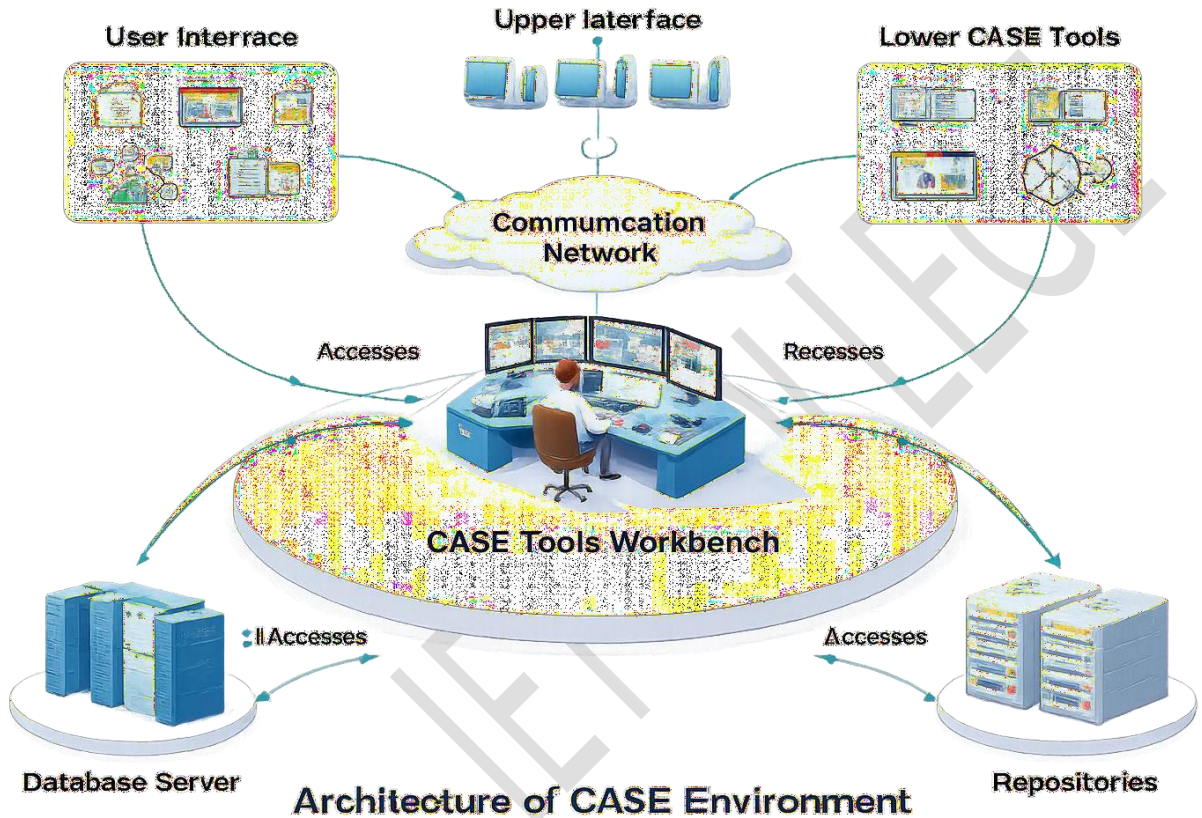
CASE tools are **software tools that automate system development activities**.

- **Functions**

•System analysis and design
•Code generation
•Documentation
•Testing and maintenance
•Reduces development time
•Better documentation

Characteristics of CASE Tools

1. **Automation**
   - Automates analysis, design, coding, testing, and documentation.
2. **Standardization**
   - Uses standard methods, notations, and templates.
3. **Central Repository**
   - Stores all system data, diagrams, and documents in one place.
4. **Consistency Checking**
   - Ensures consistency among models and documents.
5. **Productivity Improvement**
   - Reduces development time and effort.
6. **Better Documentation**
   - Automatically generates updated documentation.



Architecture of CASE Environment

Architecture of CASE Environment

A CASE environment is generally divided into **layers**.

Each layer performs a specific function.

1. User Interface Layer

•Provides interaction between **user and CASE tools**.

•Common graphical interface for all tools.

2. Tools Layer

•Contains the **actual CASE tools**.

•Supports SDLC activities such as:

- Requirement analysis
- System modelling (DFD, ERD, UML)
- Code generation
- Testing tools

•Tools work together using shared data.

3. Object Management Layer (OML)

•Acts as a **bridge** between tools and repository.

•Manages:

- Creation, deletion, and modification of objects
- Version control
- Integrity and consistency

•Ensures tools access data in a controlled manner.

4. Shared Repository Layer

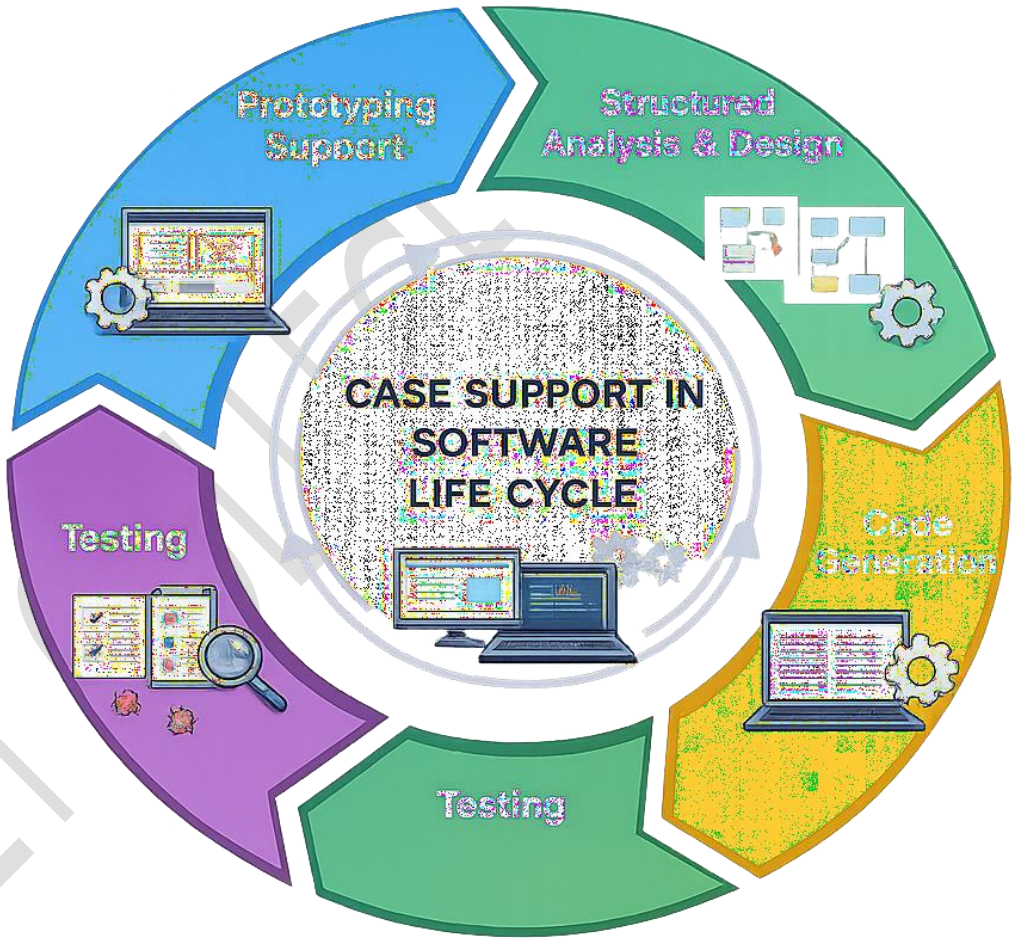
•Central database of the CASE environment.

•Stores:

- Models
- Diagrams
- Data definitions
- Documentation
- Code components

•Allows data sharing among all CASE tools.

CASE Support in Software Life Cycle
CASE (Computer-Aided Software Engineering) tools provide automated support throughout different phases of the software development life cycle (SDLC).



1. Prototyping Support

•Helps in quickly creating **working models (prototypes)** of the system

•Improves **user involvement and requirement clarity**

•Allows early detection of errors and requirement changes

2. Structured Analysis and Design

•Supports **system analysis and design techniques**

•Helps create:

- Data Flow Diagrams (DFD)
- Entity Relationship Diagrams (ERD)

3. Code Generation

•Automatically converts **design models into source code**

•Reduces manual coding effort and human errors

•Ensures standard coding practices

4. Testing Support

•Helps in **test case generation and execution**

•Supports:

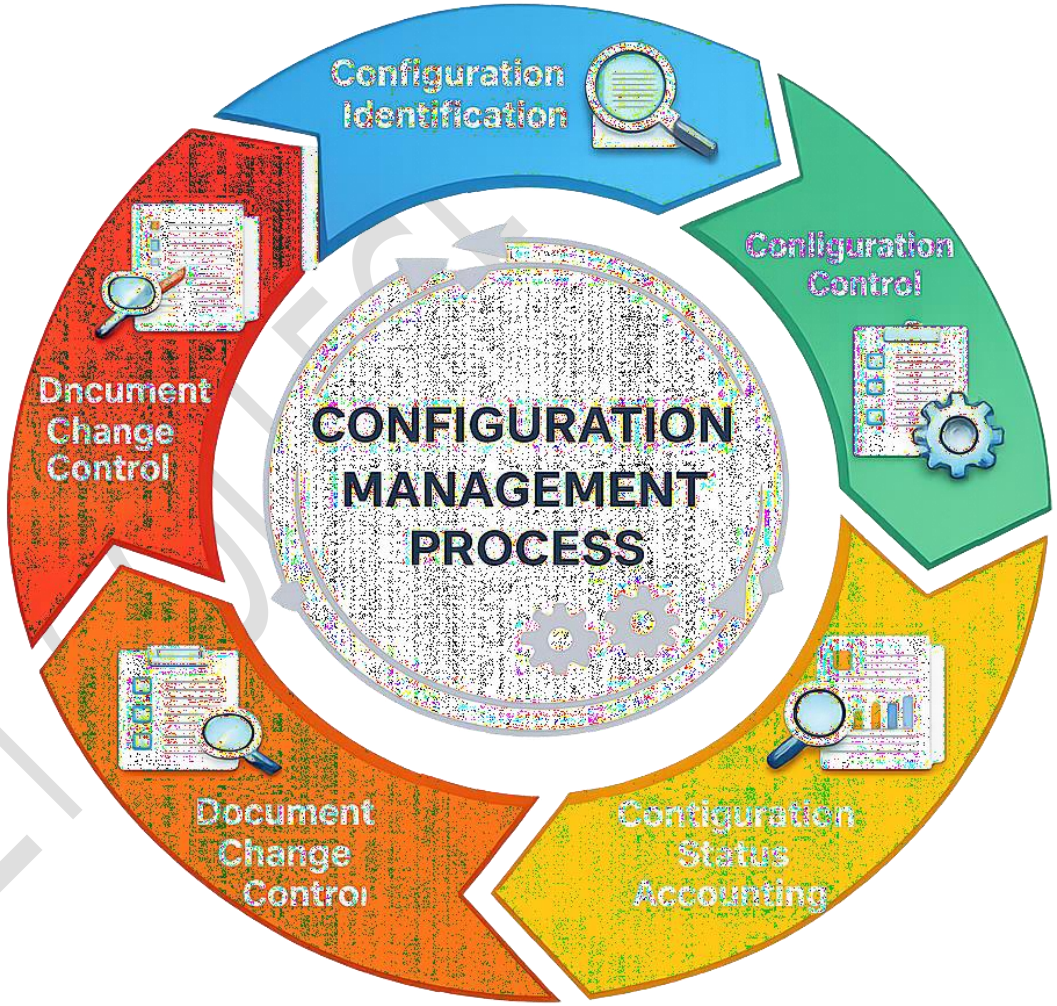
- Unit testing
- Integration testing
- System testing

•Tracks defects and test results

Configuration Management

- **Meaning**

Configuration Management (CM) is the process of **identifying, organizing, controlling, and tracking changes** in software throughout its life cycle.



Key Activities of Configuration Management

1.Configuration Identification

•Identifies **configuration items (CIs)** such as:

- Source code
- Design documents
- Test cases

•Assigns unique names and versions.

2.Configuration Control

•Controls changes to CIs.

•Uses **change request procedures**.

•Ensures only authorized changes are made.

3. Configuration Status Accounting

•Records and reports:

- Current version status
- Change history

•Helps in project tracking.

4. Configuration Auditing

•Verifies:

- Changes are correctly implemented
- Software matches documentation

•Ensures compliance with standards.

Very Short Answer Questions

1. What is system security?
2. Who is an intruder?
3. What is a computer virus?
4. What is malware?
5. What is a firewall?
6. What is intrusion detection?
7. What is configuration management?
8. What is antivirus software?
9. What is a security threat?
10. What is access control?

Short Answer Questions

1. Explain different types of intruders.
2. What are computer viruses? Give two examples.
3. Explain firewall design principles.
4. What is configuration management in security?
5. Explain the role of antivirus software in system security.

Long Answer Questions

1. Explain **intruders and their types** with examples.
2. Describe **viruses and related security threats** in detail.
3. Explain **firewall design principles** and their importance.
4. Discuss **system security tools and software platforms** with examples.
5. Explain **configuration management** and its role in maintaining system security.