Computer Intrusions

Definition:
Computer intrusion is the unauthorized access, use, or disruption of a computer system or network. It is often carried out by hackers or malicious software (malware) to steal data, damage systems, or gain control of resources.

Types of Computer Intrusions:

1. **Hacking:**
   Gaining unauthorized access to computer systems by bypassing security mechanisms.
2. **Malware Infections:**
   Use of malicious software such as viruses, worms, trojans, or ransomware to disrupt or damage systems.
3. **Phishing Attacks:**
   Trick users into revealing sensitive information like usernames, passwords, or credit card details via fake emails or websites.
4. **Denial of Service (DoS) Attacks:**
   Overloading a system or network to make it unavailable to legitimate users.
5. **Man-in-the-Middle Attacks (MITM):**
   Intercepting communications between two parties to steal or manipulate data.

Consequences of Computer Intrusions:

- **Data Theft:** Confidential information such as personal data, banking details, or intellectual property can be stolen.
- **System Damage:** Files or operating systems may be corrupted or deleted.
- **Financial Loss:** Organizations may lose money through fraud, ransom payments, or downtime.
- **Reputation Damage:** Affected companies may lose customer trust.
- **Legal Issues:** Failure to secure data can lead to legal penalties.

Prevention Techniques:

- Use of **strong passwords** and **multi-factor authentication**
- **Firewalls** and **antivirus software**
- **Regular software updates** and **patching vulnerabilities**
- **Employee training** on recognizing phishing and other threats
- **Network monitoring** and **intrusion detection systems (IDS)**

Conclusion:

Computer intrusions pose serious threats to personal, organizational, and national security. Preventive measures, user awareness, and strong cybersecurity practices are essential to safeguard digital infrastructure.

🔒 1. Firewall – Introduction (10 Marks)

Introduction:

A **firewall** is a security system that monitors and controls incoming and outgoing network traffic based on predefined security rules. It acts as a barrier between a trusted internal network and untrusted external networks (like the Internet), allowing or blocking traffic based on rules.

Purpose:

- Prevent unauthorized access.
- Monitor and filter network traffic.
- Provide a line of defense against cyber threats.

Working:

Firewalls inspect packets of data traveling to/from a network. Based on set rules (like IP address, port number, protocol), they decide whether to allow or block traffic.

Historical Context:

Firewalls have evolved from simple packet filters in the 1980s to advanced systems using AI and behavioral analysis today.

🔍 2. Characteristics and Types of Firewalls (10 Marks)

Characteristics of Firewalls:

- **Traffic Control:** Filters traffic based on IP, protocol, ports, etc.
- **Access Rules:** Implements security policies using allow/deny rules.
- **Logging and Alerts:** Keeps logs of traffic and alerts on suspicious activity.
- **Stateful Inspection:** Monitors the state of active connections.
- **Application Awareness:** Some modern firewalls understand application-level traffic.

Types of Firewalls:

1. **Packet-Filtering Firewall:**

- Works at network layer.
- Checks IP addresses, port numbers, and protocols.
- Fast but limited in filtering capabilities.

2. **Stateful Inspection Firewall:**
   - Tracks active connections.
   - More secure than packet filtering.
   - Blocks unwanted traffic based on session state.
3. **Application-Level Firewall (Proxy Firewall):**
   - Works at application layer.
   - Filters specific application traffic like HTTP, FTP.
   - Slower but more secure.
4. **Next-Generation Firewall (NGFW):**
   - Combines traditional firewall with intrusion prevention, application awareness, and deep packet inspection.
   - Uses AI/ML for threat detection.
5. **Software vs. Hardware Firewall:**
   - **Software Firewall** is installed on individual devices.
   - **Hardware Firewall** is a standalone device protecting networks.

✅ **3. Benefits of Firewalls (10 Marks)**

Benefits:

1. **Prevents Unauthorized Access:**
   - Blocks intrusions and hackers from accessing internal networks.
2. **Monitors Network Traffic:**
   - Keeps track of all inbound and outbound traffic.
3. **Protects Against Malware:**
   - Stops malicious data packets from entering the system.
4. **Enforces Security Policies:**
   - Helps organizations control what type of traffic is allowed.
5. **Reduces Risk of Attacks:**
   - Prevents attacks like DoS, brute-force, and port scanning.

6. **Content Filtering:**
    - Blocks access to unwanted or harmful websites.
7. **Customizable Rules:**
    - Rules can be configured to meet specific needs of an organization.
8. **Improves Network Performance:**
    - Can detect and stop bandwidth-hogging applications.

⚠️ 4. Limitations of Firewalls (10 Marks)

Limitations:

1. **Cannot Prevent Insider Threats:**
    - Firewalls don't detect attacks from within the network (e.g., disgruntled employees).
2. **Ineffective Against Social Engineering:**
    - Phishing, baiting, etc., bypass firewalls as they manipulate users directly.
3. **No Protection Against Encrypted Threats:**
    - Firewalls struggle with inspecting encrypted traffic unless equipped with SSL inspection.
4. **Can Be Misconfigured:**
    - Poorly set rules can lead to security holes or blocked legitimate traffic.
5. **Not a Standalone Solution:**
    - Needs to be part of a larger security framework (antivirus, IDS, training, etc.)
6. **Performance Overhead:**
    - Especially for application-level or NGFWs, performance may degrade if resources are limited.
7. **Limited Deep Inspection (in basic firewalls):**
    - Cannot analyze content within packets in basic models.

✅ 1. Trusted Systems (10 Marks)

Definition:

A **trusted system** is a computer system that is designed to enforce a specific security policy and can be relied upon to correctly protect data and resources from unauthorized access or modification.

Key Characteristics:

1. **Security Policy Enforcement:**
    - Strictly enforces rules about who can access what and how.
2. **Authentication and Authorization:**
    - Identifies users and grants permissions accordingly.
3. **Data Integrity and Confidentiality:**
    - Prevents unauthorized changes and protects sensitive information.
4. **Auditing and Logging:**
    - Maintains records of activities for analysis and accountability.
5. **Reliability and Predictability:**
    - Operates in a secure and stable manner over time.

Components:

- **Security Kernel:** The core part that enforces access control and other security policies.
- **Reference Monitor:** Controls access to resources and ensures rules are followed.

Examples:

- Military-grade systems.
- Financial institution servers.
- Secure government infrastructure.

Benefits:

- High level of security.
- Better control over system behavior.
- Trusted for handling sensitive data.

Limitations:

- Expensive to design and maintain.
- May reduce system performance.
- Complex to configure.

🔐 **2. Access Control (10 Marks)**

Definition:

Access control is a security technique that regulates who or what can view or use resources in a computing environment.

Types of Access Control:

1. **Discretionary Access Control (DAC):**
    - Users can grant access to their own files.
    - Example: File sharing permissions in Windows.
2. **Mandatory Access Control (MAC):**
    - Access is based on fixed policies set by the system (e.g., classified information access).
3. **Role-Based Access Control (RBAC):**
    - Access is based on user roles within the organization (e.g., admin, employee).
4. **Attribute-Based Access Control (ABAC):**
    - Uses attributes like user location, time, and device to grant access.

Components:

- **Subjects:** Users or processes requesting access.
- **Objects:** Files, databases, or systems being accessed.
- **Access Rights:** Permissions like read, write, execute.

Benefits:

- Prevents unauthorized access.
- Enhances security and data protection.
- Ensures accountability through user-specific controls.

Challenges:

- Complexity in large systems.
- Risk of privilege escalation if not managed properly.

🛡️ 3. Intrusion Detection (10 Marks)

Definition:

Intrusion Detection is the process of monitoring computer systems and networks for suspicious or malicious activities that could indicate a security breach or attack.

Types of Intrusion Detection Systems (IDS):

1. **Network-based IDS (NIDS):**
    - Monitors traffic across the network for threats.
2. **Host-based IDS (HIDS):**
    - Monitors a specific computer or server for signs of intrusion.

Detection Methods:

1. **Signature-Based Detection:**
    - Detects known threats using a database of signatures (like antivirus).
2. **Anomaly-Based Detection:**
    - Detects unusual behavior that deviates from the normal pattern.

Components:

- **Sensors:** Collect data from systems or network traffic.
- **Analyzers:** Evaluate data to detect intrusions.
- **User Interface:** Allows administrators to view alerts and reports.

Benefits:

- Early detection of attacks.
- Helps identify both external and internal threats.
- Logs and reports aid in forensic analysis.

Limitations:

- May generate false positives.
- Signature-based IDS can't detect new threats (zero-day).
- Requires skilled staff for management and interpretation.

🛡️ 1. Need for Intrusion Detection Systems (IDS) – Detailed (12 Marks)

🔸 Introduction:

Modern networks are exposed to a wide range of cyber threats such as hacking, malware, insider attacks, and data breaches. Traditional security tools like firewalls cannot detect all types of threats—especially **internal attacks or sophisticated intrusion attempts**. That's where an **Intrusion Detection System (IDS)** is essential.

🔹 Key Reasons for Needing IDS:

1. **Detection of Known and Unknown Attacks**
    - IDS can identify both **signature-based threats** (known patterns) and **anomalous behaviors** (sudden deviations).
    - Example: A brute force attack on login credentials or unusual login times.
2. **Continuous Network and Host Monitoring**
    - Monitors **24/7 activity** in real-time to detect suspicious patterns.
    - Provides visibility into user behavior, system calls, and network traffic.
3. **Internal Threat Detection**
    - Firewalls mainly protect from external threats.
    - IDS helps in detecting **insider threats**, such as a disgruntled employee accessing sensitive data.
4. **Support for Incident Response**
    - Triggers alerts and logs details of suspicious activities.
    - Helps security teams investigate and respond **quickly to threats**.
5. **Forensic and Legal Evidence**
    - IDS logs can be used to **analyze breaches**, identify entry points, and support legal actions.
6. **Compliance and Regulatory Requirements**
    - Many standards like **ISO 27001, PCI-DSS, HIPAA, and GDPR** require intrusion detection as part of security monitoring.
7. **Protection Against Zero-Day Attacks**
    - Anomaly-based IDS can flag **unusual activities** even when there's no known signature.
8. **Integration with Security Ecosystem**
    - Can be integrated with **SIEM (Security Information and Event Management)** and **firewalls** for better coordination and automation.

🔸 **Conclusion:**

In a world where data breaches are increasingly common and damaging, an IDS is a **critical part of layered cybersecurity**. It doesn't replace firewalls or antivirus software but enhances them by **detecting threats they miss**.

🧠 **2. Methods of Intrusion Detection – Detailed (12 Marks)**

🔸 **1. Signature-Based Detection (Misuse Detection):**

➤ **Working:**

- Compares incoming traffic, files, or system activities with a **database of known attack patterns or "signatures."**

➤ **Example:**

- Detecting a worm that always sends a specific command to a port.

➤ **Advantages:**

- Very accurate for known threats.
- Low false positives.

➤ **Disadvantages:**

- Cannot detect new (zero-day) attacks.
- Requires constant updates to signature database.

🔸 **2. Anomaly-Based Detection:**

➤ **Working:**

- Learns what "normal" behavior looks like (using statistics, AI, or machine learning).
- Flags deviations from that norm as possible intrusions.

➤ **Example:**

- A user downloads 10GB of files at midnight, which is unusual behavior.

➤ **Advantages:**

- Can detect unknown or novel attacks.
- Good for spotting insider threats or data exfiltration.

➤ **Disadvantages:**

- Higher false positives (e.g., new software behavior may be flagged).

- Requires training time to understand normal patterns.

🔸 **3. Specification-Based Detection:**

➤ **Working:**

- Defines strict rules about how specific applications or protocols should behave.
- Any deviation from the rule is flagged as suspicious.

➤ **Example:**

- If a web server sends data before receiving a request, it may be flagged.

➤ **Advantages:**

- Low false positives.
- Does not need training or previous data.

➤ **Disadvantages:**

- Time-consuming to set up.
- Difficult to maintain for complex systems.

🔸 **4. Hybrid Detection:**

➤ **Working:**

- Combines **signature** and **anomaly** techniques to benefit from both.
- First checks if an event matches a known signature; if not, it checks for anomalies.

➤ **Advantages:**

- Improved detection capability.
- Balances detection and false positives.

➤ **Disadvantages:**

- More complex and resource-intensive.
- Requires more configuration and tuning.

🧩 Types of Intrusion Detection Systems (IDS) – 12 Marks

An **Intrusion Detection System (IDS)** can be categorized based on **where** it is deployed and **how** it monitors activities. The five main types are:

🔹 1. Network-Based IDS (NIDS)

📌 Definition:

NIDS monitors **network traffic** in real-time for suspicious activity. It is typically placed at strategic locations like **gateways, routers, or switches**.

📌 How it Works:

- Analyzes **packets flowing across the network**.
- Detects abnormal patterns, such as port scans, DDoS attacks, or worms.

📌 Pros:

- Can monitor large volumes of traffic.
- Detects attacks across the whole network.

📌 Cons:

- Encrypted traffic (HTTPS) is hard to inspect.
- Cannot detect internal host-level attacks.

📌 Example:

Snort (open-source NIDS).

🔹 2. Host-Based IDS (HIDS)

📌 Definition:

HIDS is installed on **individual systems (servers or endpoints)** and monitors **local system activity** like logs, file integrity, and system calls.

📌 How it Works:

- Monitors OS logs, system changes, and unauthorized file modifications.
- Can detect attacks specific to that system.

📌 Pros:

- Effective for detecting insider threats.
- Can monitor encrypted traffic as it's decrypted on the host.

📌 Cons:

- High overhead on the host system.
- Difficult to manage on large-scale systems.

📌 **Example:**

OSSEC (open-source HIDS), Tripwire.

🔹 **3. Protocol-Based IDS (PIDS)**

📌 **Definition:**

PIDS is designed to monitor and analyze **specific network protocols** like HTTP, FTP, or DNS.

📌 **How it Works:**

- Understands the **structure and behavior** of a specific protocol.
- Detects attacks that exploit protocol vulnerabilities.

📌 **Pros:**

- Precise detection of protocol-level attacks.
- Ideal for protecting web servers and application layers.

📌 **Cons:**

- Limited to specific protocols.
- Needs constant updates for new protocol changes.

📌 **Example:**

ModSecurity (used for HTTP/HTTPS traffic).

🔹 **4. Application Protocol-Based IDS (APIDS)**

📌 **Definition:**

APIDS operates at the **application layer**, analyzing traffic between the user and application.

📌 **How it Works:**

- Monitors API calls, database queries, or session behavior.
- Can detect **SQL injections, cross-site scripting (XSS), etc.**

📌 **Pros:**

- Deep insight into application-level behavior.
- Useful in detecting **business logic attacks**.

📌 **Cons:**

- Application-specific – not general-purpose.
- May require deep integration with applications.

📌 **Example:**

Custom IDS integrated into a banking app backend.

🔹 **5. Hybrid IDS**

📌 **Definition:**

Hybrid IDS combines multiple types (e.g., NIDS + HIDS) to leverage the advantages of each.

📌 **How it Works:**

- Uses NIDS to monitor network traffic.
- Uses HIDS to monitor host activity.
- Data is **correlated in a central management system**.

📌 **Pros:**

- Improved coverage and detection.
- Balances strengths of network and host detection.

📌 **Cons:**

- High complexity.
- Expensive and resource-heavy to deploy and maintain.

📌 **Example:**

Suricata (when used with host agents), IBM QRadar with endpoint agents.

🗒️ **Summary Table:**

Type	Monitors	Strength	Weakness
NIDS	Network traffic	Real-time, broad view	Can't inspect encrypted data
HIDS	Host logs/files	Detects insider/local attacks	Host resource overhead
PIDS	Specific protocols	Targeted protocol protection	Limited scope
APIDS	Application traffic	Detects app-layer (e.g. XSS, SQLi)	App-specific
Hybrid	Network + Hosts	High detection, broad + deep view	Expensive and complex

🔐 Password Management – 12 Marks

🔸 Introduction:

Passwords are the **first line of defense** for most systems and user accounts. Proper password management ensures that **authentication is secure**, reducing the risk of unauthorized access and data breaches.

🔹 1. Password Management Practices:

✅ a. Strong Password Policies

- Enforce minimum length (e.g., 8–12 characters).
- Require a mix of **uppercase, lowercase, numbers, and symbols**.
- Avoid using dictionary words or personal information.

✅ b. Password Expiry and Rotation

- Users are forced to change passwords **periodically** (e.g., every 90 days).
- Prevents continued use of compromised passwords.

✅ c. Account Lockouts and Login Attempts

- Lock accounts temporarily after a set number of failed attempts.
- Prevents **brute-force attacks**.

✅ d. Multi-Factor Authentication (MFA)

- Combines passwords with other authentication factors like **OTP, biometrics, or smart cards**.
- Adds an extra layer of security.

✅ e. Password Hashing and Storage

- Passwords should never be stored in plain text.
- Use strong **hashing algorithms** (e.g., bcrypt, SHA-256 with salt) to protect stored passwords.

✅ f. Password Managers

- Secure applications that store and autofill strong, unique passwords for each site.
- Examples: **Bitwarden, LastPass, KeePass**.

🔹 2. Password Management Tools:

- **Client-based tools:** KeePass, Bitwarden (local storage).
- **Cloud-based tools:** LastPass, Dashlane (access anywhere).
- **Enterprise-level tools:** Manage credentials across teams with audit logs and access control.

🔹 **3. Best Practices Summary:**

Good Practice	Why It's Important
Use strong, unique passwords	Prevents guessing and reuse attacks
Enable MFA	Adds a second line of defense
Avoid password reuse	Prevents one breach from affecting others
Don’t write down passwords	Prevents physical compromise
Use password managers	Stores and generates strong passwords

🔐 **Limitations and Challenges of IDS (5 Points × 3 Lines)**

1. False Positives

IDS may generate alerts for legitimate activity that resembles malicious behavior.
This leads to unnecessary alarms, wasting time and resources.
Frequent false positives can cause administrators to ignore real threats.

2. Cannot Prevent Attacks

An IDS only **detects** intrusions; it doesn’t **block** or stop them.
By the time an alert is raised, the system might already be compromised.
It must be paired with prevention systems (like firewalls or IPS).

3. Encrypted Traffic Limitations

IDS struggles to inspect encrypted data (e.g., HTTPS traffic).
Attackers can hide malicious payloads inside encrypted packets.
Decrypting traffic adds complexity and may violate privacy policies.

4. Skilled Attackers Can Evade IDS

Attackers may use stealthy techniques like **fragmented packets or obfuscation**.
Some IDS may not reconstruct such traffic correctly, missing the threat.
This makes it harder to detect advanced persistent threats.

5. Requires Constant Tuning and Expertise

IDS systems need **regular updates, rule tuning, and monitoring**.
Improperly configured IDS can become noisy or ineffective.
Security teams need expertise to analyze alerts and maintain the system.