**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.

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

**2. Methods of Intrusion Detection (12 Marks)**

**Intrusion Detection Systems (IDS) use different methods to detect malicious or unauthorized activity. The two primary methods are:**

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

**Working:**

**Signature-based detection identifies intrusions by comparing system activities, files, or network traffic to a database of known attack patterns, known as signatures. These signatures can represent known malware behavior, specific commands, or predefined traffic patterns associated with malicious actions.**

**When the IDS observes an activity that matches a stored signature, it generates an alert indicating a potential intrusion.**

**Example:**

**A worm that always sends a specific command to a particular network port can be easily detected by a signature-based IDS if that command is part of its signature database.**

**Advantages:**

* **Very accurate for known threats.**
* **Produces low false positives since it looks for exact matches.**

**Disadvantages:**

* **Unable to detect new, unknown (zero-day) attacks because no signature exists for them.**
* **Requires constant updates to the signature database to stay effective against newly discovered threats.**
* **Not effective against slightly modified versions of known attacks unless signatures are also updated for those variants.**

**Best Use Cases:**

* **Environments with well-understood threats.**
* **Systems requiring low false alarms.**
* **Detecting widespread malware and common exploits.**

**2. Anomaly-Based Detection**

**Working:**

**Anomaly-based detection focuses on identifying deviations from normal system or user behavior. It works by first creating a baseline of what is considered "normal" activity within a system or network. This baseline can include typical login times, average data usage, or usual patterns of application behavior.**

**Once this baseline is established, the IDS monitors real-time activities. Any deviation from the normal baseline is flagged as potentially malicious.**

**This method often uses statistical techniques, artificial intelligence, or machine learning algorithms to build models of normal behavior.**

**Example:**

**If a user who usually accesses files during business hours suddenly begins downloading large amounts of data late at night, the system may detect this as abnormal behavior and raise an alert.**

**Advantages:**

* **Capable of detecting new and unknown threats, including zero-day attacks.**
* **Effective against insider threats, data leaks, and unusual user activities.**
* **Does not depend on prior knowledge of specific attack signatures.**

**Disadvantages:**

* **Higher rate of false positives, especially during the initial learning phase or when normal usage patterns change.**
* **Requires time to learn what constitutes normal behavior.**
* **Needs regular updates to the behavior model to remain accurate over time.**

**Best Use Cases:**

* **Detecting sophisticated attacks not covered by known signatures.**
* **Identifying insider threats or compromised accounts.**
* **Monitoring critical systems for unusual behavior patterns.**

**Comparison Table**

| **Feature** | **Signature-Based Detection** | **Anomaly-Based Detection** |
| --- | --- | --- |
| **Detection Type** | **Known attack patterns** | **Deviations from normal behavior** |
| **Ability to Detect Zero-Day** | **No** | **Yes** |
| **False Positive Rate** | **Low** | **High** |
| **Maintenance Requirements** | **Frequent signature updates** | **Regular model retraining** |
| **Adaptability** | **Limited to known threats** | **Can adapt to new and unknown threats** |
| **Initial Setup Time** | **Fast** | **Longer (requires training)** |
| **Best Use Cases** | **Detecting common malware** | **Insider threats and unknown attacks** |

**🧩 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 two main types are:

**1. Network-based Intrusion Detection System (NIDS)**

**Definition:**

**A Network-based IDS (NIDS) is a security solution that monitors and analyzes traffic flowing across a network in real-time. It inspects data packets traveling through the network and attempts to detect suspicious patterns that may indicate an attack or intrusion.**

**How it Works:**

* **NIDS is usually placed at strategic points within the network, such as near routers, switches, or firewalls.**
* **It passively analyzes traffic by capturing packets directly from the network interface using tools like libpcap or WinPcap.**
* **It examines packet headers, payloads, and overall traffic patterns for known attack signatures or unusual behavior.**

**Detection Methods:**

* **Signature-based Detection: Compares network traffic to a database of known attack signatures (like virus definitions).**
* **Anomaly-based Detection: Learns normal network behavior and flags anything that deviates from this baseline.**
* **Protocol Analysis: Ensures protocols are used correctly; flags malformed or non-standard protocol usage.**

**Use Cases:**

* **Detecting Denial of Service (DoS) or Distributed Denial of Service (DDoS) attacks.**
* **Identifying port scans or unauthorized attempts to access network services.**
* **Monitoring internal traffic for lateral movement of malware.**

**Advantages:**

* **Covers the entire network segment; one NIDS can monitor many devices.**
* **No need to install software on individual hosts.**
* **Useful for detecting large-scale network attacks.**

**Disadvantages:**

* **Cannot see encrypted traffic (like HTTPS) unless placed behind a decryption device.**
* **May miss attacks that happen only on a single host (especially internal compromise).**
* **Can be overwhelmed by high traffic volumes, leading to missed alerts or false negatives.**

**Examples:**

* **Snort – An open-source NIDS that supports both signature and anomaly-based detection.**
* **Suricata – A high-performance NIDS with multi-threading support.**
* **Zeek (formerly Bro) – Focuses more on network behavior and traffic analysis.**

**2. Host-based Intrusion Detection System (HIDS)**

**Definition:**

**A Host-based IDS (HIDS) is installed on individual computers or servers. It monitors the internal operations of a system, such as system calls, application logs, file integrity, and user activity, to detect malicious activity.**

**How it Works:**

* **HIDS software runs on each host and analyzes system-level data like:**
  + **Operating system logs**
  + **Application logs**
  + **Configuration files**
  + **File system changes**
  + **Active processes**
* **It may use checksum or hash comparison to detect unauthorized changes to critical system files (e.g., /etc/passwd in Linux).**

**Detection Methods:**

* **Signature-based: Recognizes known malicious behavior patterns or file changes.**
* **Anomaly-based: Learns normal user and system behavior and flags deviations.**
* **File Integrity Monitoring (FIM): Detects unauthorized modifications to files.**

**Use Cases:**

* **Detecting privilege escalation attempts or unauthorized root/admin access.**
* **Monitoring sensitive files and configuration changes.**
* **Detecting malware or rootkits that do not generate network traffic.**

**Advantages:**

* **Can detect attacks that do not generate any network activity.**
* **Provides a detailed view of what is happening on a specific system.**
* **Useful for detecting insider threats or compromised accounts.**

**Disadvantages:**

* **Needs to be installed and managed on each individual host.**
* **Can be disabled by a sophisticated attacker if the system is compromised.**
* **Limited visibility into broader network context.**

**Examples:**

* **OSSEC – An open-source HIDS that supports log analysis, file integrity checking, and alerting.**
* **Tripwire – A tool for monitoring file integrity and alerting on changes.**
* **AIDE (Advanced Intrusion Detection Environment) – Another file integrity checker used in Unix-like systems.**

**Comparison Table:**

| **Feature** | **NIDS** | **HIDS** |
| --- | --- | --- |
| **Monitoring Scope** | **Network traffic** | **Specific host system** |
| **Deployment** | **On network perimeter or tap** | **On each machine (endpoint)** |
| **Can Detect** | **Network-based attacks, port scans** | **File tampering, privilege escalation** |
| **Resource Usage** | **Low on hosts; processing on network** | **High CPU/Memory usage on host system** |
| **Visibility** | **Broad network-level visibility** | **Deep host-level visibility** |
| **Encrypted Traffic** | **Cannot inspect (unless decrypted)** | **Can inspect if decrypted on host** |
| **Example Tools** | **Snort, Suricata, Zeek** | **OSSEC, Tripwire, AIDE** |

**🔐 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.