Exam Cram Notes: Types of Vulnerabilities

1. What is a Vulnerability?

A **vulnerability** is a weakness in a system, network, or application that can be exploited by a threat actor to gain unauthorized access or perform malicious activities. These weaknesses can be found in software, hardware, processes, or even human behavior.

---

2. Common Types of Vulnerabilities

Understanding the various types of vulnerabilities helps in identifying and mitigating risks effectively.

1. **Software Vulnerabilities**
    - **Description**: Flaws or weaknesses in application code or software that can be exploited.
    - **Examples**:
        - **Buffer Overflow**: A condition where a program writes more data to a buffer than it can hold, causing memory corruption and potential code execution.
        - **SQL Injection**: Malicious SQL code inserted into an input field, allowing attackers to execute arbitrary database queries.
        - **Cross-Site Scripting (XSS)**: Injection of malicious scripts into web pages, which are executed by users' browsers.
        - **Remote Code Execution (RCE)**: Allows attackers to run arbitrary code on a victim machine.
    - **Mitigation**: Regular patching, secure coding practices, input validation, and using web application firewalls.
2. **Network Vulnerabilities**
    - **Description**: Weaknesses in network architecture, protocols, or devices that can be exploited by attackers.
    - **Examples**:
        - **Unencrypted Traffic**: Transmitting sensitive information (e.g., passwords) over unencrypted protocols like HTTP instead of HTTPS.
        - **Open Ports**: Exposed ports that provide an entry point for attackers (e.g., Telnet, SMB).
        - **Weak Encryption**: Use of weak algorithms like DES or outdated SSL/TLS versions.
        - **DNS Spoofing**: Attackers manipulate DNS records to redirect traffic to malicious sites.
    - **Mitigation**: Use of strong encryption (e.g., AES), closing unused ports, and using firewalls and intrusion detection systems.
3. **Authentication Vulnerabilities**
    - **Description**: Weaknesses in user authentication mechanisms that can lead to unauthorized access.
    - **Examples**:

        - **Weak Passwords**: Simple, easily guessable passwords or lack of password complexity.
        - **Brute Force Attacks**: Automated attacks that attempt to guess passwords.
        - **Insecure Session Management**: Failure to securely manage session IDs, leading to session hijacking.
        - **Lack of Multi-Factor Authentication (MFA)**: Relying solely on passwords for authentication.
    - **Mitigation**: Implementing MFA, enforcing strong password policies, using CAPTCHAs, and using secure session management practices.
4. **Configuration Vulnerabilities**
    - **Description**: Improper or insecure configurations in systems, software, or hardware that increase exposure to threats.
    - **Examples**:
        - **Default Credentials**: Using default usernames and passwords (e.g., admin/admin) for systems and devices.
        - **Misconfigured Firewalls**: Incorrect firewall rules that leave critical systems exposed to the internet.
        - **Overly Permissive Access Control**: Giving users excessive permissions or access to sensitive resources.
        - **Unpatched Software**: Running outdated or unpatched software that has known security vulnerabilities.
    - **Mitigation**: Harden configurations by disabling unnecessary services, changing default passwords, and regularly patching systems.
5. **Human Vulnerabilities (Social Engineering)**
    - **Description**: Exploiting human behavior or lack of awareness to bypass security controls.
    - **Examples**:
        - **Phishing**: Sending deceptive emails or messages to trick users into revealing sensitive information.
        - **Pretexting**: Manipulating individuals to divulge confidential information by pretending to be someone else.
        - **Tailgating**: Gaining physical access to secure areas by following an authorized person.
        - **Baiting**: Offering something enticing (e.g., free software or rewards) to trick users into compromising security.
    - **Mitigation**: User education and awareness, multi-factor authentication, physical security controls, and implementing security awareness training.
6. **Operating System Vulnerabilities**
    - **Description**: Weaknesses in operating systems (OS) that can be exploited by attackers to gain control over a machine or network.
    - **Examples**:
        - **Privilege Escalation**: Exploiting a flaw to gain higher-level privileges or root access.
        - **Unpatched OS**: Running operating systems without the latest security patches and updates.
        - **Vulnerable Services**: Unnecessary or insecure services (e.g., SMBv1) that can be exploited.

        - **Kernel Vulnerabilities**: Bugs or weaknesses in the OS kernel that allow malicious code to execute.
    - **Mitigation**: Regular OS updates and patches, disabling unnecessary services, and using anti-virus software.
7. **Physical Vulnerabilities**
    - **Description**: Weaknesses in physical security controls that allow attackers to gain unauthorized access to systems, networks, or sensitive data.
    - **Examples**:
        - **Uncontrolled Access**: Lack of secure entry points, allowing unauthorized individuals to enter restricted areas.
        - **Theft of Devices**: Laptops, hard drives, or other devices being physically stolen or lost, exposing sensitive data.
        - **Unprotected Data Storage**: Storing sensitive data without encryption or physical security controls.
    - **Mitigation**: Implementing physical security measures like access control, video surveillance, and device encryption.
8. **Cloud Vulnerabilities**
    - **Description**: Weaknesses within cloud environments or services that can lead to data breaches or unauthorized access.
    - **Examples**:
        - **Misconfigured Cloud Settings**: Cloud services that are incorrectly configured, leading to data exposure.
        - **Insecure APIs**: Vulnerabilities in cloud application programming interfaces (APIs) that can be exploited.
        - **Shared Responsibility Model Misunderstanding**: Not understanding what security is the provider's responsibility versus the customer's.
        - **Insufficient Authentication**: Weak or missing authentication methods for cloud access.
    - **Mitigation**: Secure cloud configurations, use of strong authentication, encryption, and regular audits of cloud services.
9. **Mobile Vulnerabilities**
    - **Description**: Vulnerabilities specific to mobile devices and apps.
    - **Examples**:
        - **Insecure Mobile Apps**: Apps that do not properly secure user data or communications.
        - **Mobile Device Theft**: Loss or theft of a mobile device that contains sensitive data.
        - **Malware**: Malicious apps or software designed to compromise mobile devices.
        - **Unpatched Mobile OS**: Running outdated mobile operating systems that have known vulnerabilities.
    - **Mitigation**: Mobile device management (MDM), app vetting, encryption, and regular OS updates.

---

3. Key Exam Tips

✅ **Know the common types of vulnerabilities** (software, network, human, etc.) and the types of attacks that can exploit them.
✅ Understand **vulnerability management** techniques, such as patching, securing configurations, and implementing strong access controls.
✅ Expect **scenario-based questions** on how to identify and mitigate vulnerabilities in different environments.
✅ Be able to differentiate between various types of vulnerabilities, such as **configuration flaws** versus **software flaws**.