10 Common Types of Vulnerabilities

1

Injection Vulnerabilities

- Occur when untrusted data is sent to an interpreter as part of a command or query. The most common form is SQL injection, but other types include command injection and LDAP injection.
- **Impact:** This can lead to data leakage, corruption, or unauthorized access to sensitive information. Attackers may execute arbitrary commands or queries, potentially controlling the system.
- **DevSecOps Role:** By integrating security checks early, DevSecOps can help prevent injection attacks by incorporating secure coding practices and regular code scanning for injection flaws.

2

Cross-Site Scripting (XSS)

- Occurs when an attacker injects malicious scripts into content that is then executed by another user's browser. It exploits vulnerabilities in web applications that don't properly validate user input.
- **Impact:** Attackers can steal session tokens, deface websites, or redirect users to malicious sites.
- **DevSecOps Role:** Automating input sanitization and output encoding during the development process helps minimize the risks of XSS vulnerabilities.

3

Broken Authentication and Session Management

- Arises when applications do not correctly handle authentication tokens, session identifiers, or passwords. Attackers exploit flaws in the authentication mechanisms to impersonate legitimate users.
- I**mpact:** Attackers can gain unauthorized access to sensitive data or functions by hijacking user sessions or bypassing authentication altogether.
- DevSecOps Role: Continuous monitoring and testing of authentication protocols through tools in the CI/CD pipeline ensure robust protection against such vulnerabilities.

4

Insecure Direct Object References (IDOR)

- Occurs when an application exposes internal implementation objects (e.g., files, directories, database records) without proper access control.
- **Impact:** Attackers can manipulate these references to access unauthorized data or resources.
- **DevSecOps Role:** Incorporating access control mechanisms and frequent security testing helps detect and prevent IDOR vulnerabilities during development.

10 Common Types of Vulnerabilities

5

Security Misconfiguration

- Happen when security settings in an application, server, or database are not implemented correctly or remain at default values.
- **Impact:** This can open doors for attackers to exploit weak passwords, outdated software, and unpatched systems.
- **DevSecOps Role:** Automating configuration management and continuous scanning for outdated components help mitigate misconfigurations.

6

Cross-Site Request Forgery (CSRF)

- Tricks users into submitting unintended requests, potentially performing unauthorized actions on their behalf.
- **Impact:** It can allow attackers to transfer funds, change account settings, or perform other malicious actions without the user's consent.
- **DevSecOps Role:** Implementing anti-CSRF tokens during development helps ensure that the application is resistant to such attacks.

7

Insecure Deserialization

- Occurs when untrusted or tampered serialized data is processed by an application. Attackers manipulate the deserialization process to execute arbitrary code.
- **Impact:** This can lead to remote code execution or denial of service (DoS) attacks.
- **DevSecOps Role:** By enforcing strict input validation and continuous security checks during the serialization process, such vulnerabilities can be mitigated early on.

8

Insufficient Logging and Monitoring

- Failure to log and monitor critical activities can prevent the detection of attacks. Without proper logging, security breaches might go unnoticed.
- **Impact:** Delayed detection increases the time attackers have to exploit vulnerabilities.
- **DevSecOps Role:** DevSecOps emphasizes continuous monitoring, ensuring that logs are maintained and anomalies are quickly detected to reduce response times.

10 Common Types of Vulnerabilities

9

Outdated Software Components

- Using outdated or unsupported components can introduce vulnerabilities because they are no longer maintained and patched by vendors.
- **Impact:** Attackers exploit known vulnerabilities in outdated components to compromise the application.
- **DevSecOps Role:** Automated dependency management tools integrated into the development workflow help ensure that only up-to-date, secure components are used.

10

Insufficient Data Protection

- This includes weak encryption or unencrypted sensitive data in transit or at rest. It makes it easier for attackers to intercept and exploit sensitive information.
- **Impact:** Data breaches and unauthorized access to personal or financial data.
- **DevSecOps Role:** Applying encryption best practices and automating encryption enforcement ensures that data remains secure throughout its lifecycle.

When discussing vulnerabilities in the context of software development and security, it's essential to recognize the various types that can impact an application's integrity. These vulnerabilities often arise from coding flaws, misconfigurations, or oversight during development. Here's a breakdown of common types of vulnerabilities:

By integrating security early in the development process (a core principle of DevSecOps), organizations can address these vulnerabilities proactively, reducing risk and enhancing application security throughout the software lifecycle.

WABBI