Security report

OWASP top 10

|  | Likelihood | Impact | Risk | Possible actions | Planned |
| --- | --- | --- | --- | --- | --- |
| Broken Access Control | Low | High | Moderate | * Prevent unauthorised users from access   - Different levels of permission depending on the type of user | Yes |
| Cryptographic Failures | Moderate | High | Moderate | All passwords should be salted and hashed | Yes |
| Injection | Unlikely | High | Moderate | Using an ORM that prevents the possible injections or manually take care of it | Yes |
| Insecure design | Low | Moderate | Moderate | The application should follow the design patterns and conventions for writing good code. | Yes |
| Security misconfiguration | Moderate | Moderate | Moderate | The application is set with the required security values. | Yes |
| Vulnerable and Outdated Components | Very Low | Moderate | Moderate | Using the news versions of the software components and update the application if there are new ones | Yes |
| Identification and Authentication Failures | High | High | High | The application should not allow users to be able to create easy to crack passwords | No |
| Software and Data Integrity Failures | Low | Moderate | Low | The application uses trusted libraries and component providers. | Yes |
| Security Logging and Monitoring Failures | Moderate | Moderate | Moderate | Monitoring the application activities | No |
| Server Side Request Forgery (SSRF) | low | low | low | The application sends verified data only to trusted URLs | No |

1. **Broken access control**

Access control ensures that users are restricted to their authorised permissions, preventing them from accessing other functions that should be outside of their reach. In case of failure data might be leaked, modified or deleted.

1. **Cryptographic failure**

A cryptographic failure is a security failure that occurs when a third-party entity (apps, web pages, different websites) exposes sensitive data. To be exact, it’s when that entity does so without specific intent behind it. For this reason Sensitive information such as passwords, credit card numbers and more need to be stored securely. In general, the less sensitive information you store the better.

1. **Injection**

An injection might happen when the application does not verify the data received by the user. Dynamic queries or non-parameterized calls can be sent in order to expose or store data.

1. **Insecure design**

We have a reason for distinguishing between design flaws and implementation defects because they have different causes and solutions. A secure design can still have vulnerabilities if it is not implemented correctly. On the other hand, even a perfect implementation cannot fix an insecure design because it lacks the necessary security controls to defend against specific attacks. One of the reasons for insecure designs is the absence of business risk assessment during the development of the software or system, which results in a failure to determine the required level of security.

1. **Security misconfiguration**

The application might be vulnerable if it’s missing security hardening across any part of it. Other causes might be default accounts and their passwords that are still enabled or the security settings in the application servers are not set to secure values.

1. **Vulnerable and Outdated Components**

A vulnerable and outdated component is a software component that is no longer being supported by the developer, making it susceptible to security vulnerabilities. Many times, a component has known vulnerabilities that don’t

get fixed due to a lack of maintainers.

1. **Identification and Authentication Failures**

Confirmation of the user's identity, authentication, and session management is critical to protect against authentication-related attacks. There may be authentication weaknesses if the application automated attacks such as credential stuffing, brute force or default well known passwords.

1. **Software and Data integrity failures**

Software and data integrity failures relate to code and infrastructure that does not protect against integrity violations. An example of this is where an application relies upon plugins, libraries, or modules from untrusted sources, repositories, and content delivery networks (CDNs). An insecure CI/CD pipeline can introduce the potential for unauthorised access, malicious code, or system compromise.

1. **Security logging and monitoring failures**

Without logging and monitoring, breaches cannot be detected. Insufficient logging, detection, monitoring, and active response occurs any time auditable events, such as logins, failed logins, and high-value transactions, are not logged.

1. **Server side request forgery**

SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL).