Wolkgeluid

OWASP top 10 mitigation report

Based on Acunetix test runs from 07-01-2023

Luc Janssen

Acunetix

This document covers vulnerabilities discovered by an Acunetix test run on January 7th 2023, and recommendations on the mitigations of these issues.

Context

This report is a comprehensive analysis of security vulnerabilities that have been identified in the Wolkgeluid web application. The OWASP Top 10 is a widely accepted standard for identifying and mitigating the most critical web application security risks. The report covers the top 10 categories of vulnerabilities as identified by OWASP.

In this report, you will find a summary of each vulnerability and specific recommendations on how to mitigate and remediate each issue. It's important to remember that security is a process and not a product, therefore the recommendations are not one time fixes but should be integrated as a regular part of the development process. It's also important to keep in mind that the report is based on the results of a single Acunetix scan, and regular testing should be done as well as security testing should be done by using different tools and methods.

Contents

A02 Cryptographic Failures

The first thing is to determine the protection needs of data in transit and at rest. For example, passwords, credit card numbers, health records, personal information, and business secrets require extra protection, mainly if that data falls under privacy laws

Unencrypted connection

Issue

Unencrypted connection

Summary

Application was found to be connected over an unencrypted connection, which means that a potential attacker can intercept and modify data sent and received from this site. The potential impact of this vulnerability is possible information leaks.

Recommendations

1. Implement HTTPS on the application to ensure that all data sent and received is encrypted and secure. This will prevent any potential attackers from intercepting and modifying data in transit.
2. Regularly monitor the site for any suspicious activity or unauthorized access attempts.

A04 Insecure design

Insecure design is a broad category representing different weaknesses, expressed as "missing or ineffective control design." Insecure design is not the source for all other Top 10 risk categories. There is a difference between insecure design and insecure implementation. We differentiate between design flaws and implementation defects for a reason, they have different root causes and remediation. A secure design can still have implementation defects leading to vulnerabilities that may be exploited. An insecure design cannot be fixed by a perfect implementation as by definition, needed security controls were never created to defend against specific attacks. One of the factors that contribute to insecure design is the lack of business risk profiling inherent in the software or system being developed, and thus the failure to determine what level of security design is required.

Missing Headers

Issue

Missing Content Security Policy (CSP) and Permissions-Policy headers

Summary

It was detected that the application has an insecure design and is missing Content Security Policy (CSP) and Permissions-Policy headers. The lack of these headers can lead to vulnerabilities such as Cross Site Scripting and data injection attacks.

Recommendations

1. Implement Content Security Policy (CSP) in the web application by adding the Content-Security-Policy HTTP header and giving it values to control resources the user agent is allowed to load for that page.
2. Implement Permissions-Policy header to selectively enable and disable use of various browser features and APIs.
3. Regularly monitor the web application for any suspicious activity or unauthorized access attempts.

A05 Security Misconfiguration

Security misconfiguration is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched and upgraded in a timely fashion.

Missing Headers

Issue

Missing Content Security Policy (CSP) and Permissions-Policy header

For summary and mitigation refer to **A04 Insecure design**

Cookies without HttpOnly flag set

Issue

Cookies without HttpOnly flag set

Summary

It was detected that the web application has a security misconfiguration issue, specifically cookies without the HttpOnly flag set. This means that these cookies can be accessed by client-side scripts.

Recommendations

1. Set the HttpOnly flag for all cookies in the web application, which will prevent them from being accessed by client-side scripts.

A06 Vulnerable and Outdated Components

Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.

Missing Headers

Issue

Missing Content Security Policy (CSP) and Permissions-Policy header

For summary and mitigation refer to **A04 Insecure design**

Cookies without HttpOnly flag set

Issue

Cookies without HttpOnly flag set

For summary and mitigation refer to **A05 Security Misconfiguration**

A07 Identification and Authentication Failures

Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.

Unencrypted Connection

Issue

Unencrypted connection

For summary and mitigation refer to **A02 Cryptographic Failures**