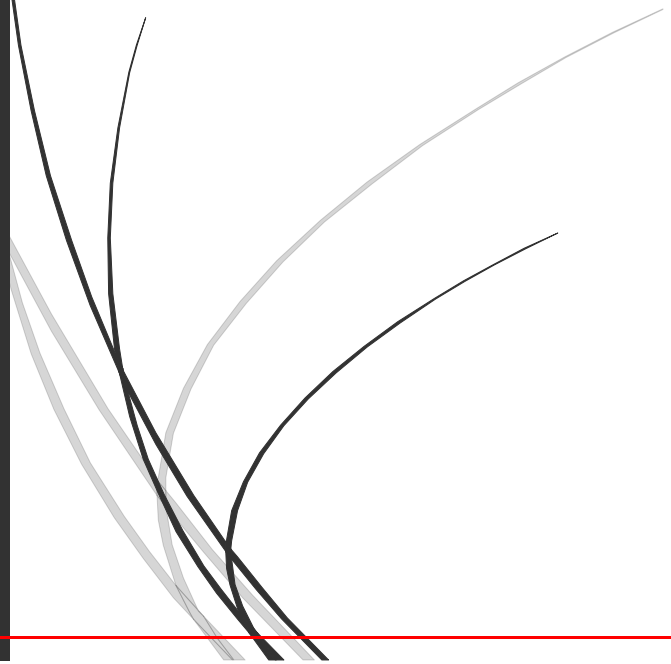
6/3/2022

Security Report

Individual Track



Ismail Chbiki

Table Of Content

Introduction

This report documents some security aspects of the software system of my Web App according to the OWASP Top 10 latest installment.

It describes some potential openings that can risk the application to be compromised and hacked. It also scales their severity, likelihood to happen and risk, and presents the possible planned actions.

Finally, this document adds a reasoning to indicate the meaning to the security risks and the impact of affecting the software system of my Web App.

Analysis Table

	Likelihood	Impact	Risk	Actions possible	Planned
A01 - Broken Access Control	Unlikely	Severe	High	N/A	N/A
A02 - Cryptographic Failures	Unlikely	Severe	High	Passwords are encrypted	Yes, Users' data need encryption as well
A03 - Injection	High	Very Severe	Very high	All input data need to be validated on the client and server side	Yes, More input validations will be added
A04 - Insecure Design	Unlikely	Low	Low	The objective is to write clean code with the recommended design principles	Yes, Code is intended to be well designed and structured

A05 - Security Misconfiguration	High	Severe	High	Recommended or solid custom configs need to be adapted	Yes,
A06 - Vulnerable and Outdated Components	Medium	Moderate	High	All the software components' versions are being occasionally updated/upgraded	Yes, Day-to-day versions updated and refactor
A07 - Identification and Authentication Failures	High	Severe	High	Password are encrypted but there is no automated mechanism to validate typed passwords strength	Yes, Some user input is validated but not throughout the whole app
A08 - Software and Data Integrity Failures	High	Severe	High	The software system might download and install some untrusted software or plugins that might compromise the software system	No measures are planned ahead yet to prevent this risk
A09 - Security Logging and Monitoring Failures	High	Severe	Very high	The app doesn't detect or monitor cyber-attacks.	No measures are planned ahead yet to prevent this risk
A10 - Server-Side Request Forgery	High	Severe	Very high	The server-side is secured and protected but more validation should be supplied to prevent forgery	Yes, Server-side and URL requests validation and control is planned to be added

Reasoning

A01 - Broken Access Control

This can risk users to access restricted paths and perform tasks that normally shouldn't be allowed to them.

A02 - Cryptographic Failures

The risk here lies in the possibility to read data of users (ex. Passwords, bank account numbers) if it is not encrypted and if it is written in plain text in case there is a compromise or a data leak.

A03 - Injection

This is one of the most severe attacks that threaten a lot of systems as attackers can inject commands to steal data or get access of some system via custom written queries that were not expected by the system.

A04 - Insecure Design

The risk here can be caused by a weak code architecture and design which causes the complexity of the system and results in unexpected defects.

A05 - Security Misconfiguration

The risk here can be produced in case of unproper configurations or/and enabling some unnecessary features while it could have been set to the minimum access and usage.

A06 - Vulnerable and Outdated Components

The vulnerability here is when components are left outdated which can result sometimes in some unsupported versions that can cause some harm to the software eventually

A07 - Identification and Authentication Failures

An extra measure to be taken to prevent this risk is to set two-factor authenticator so that in case for example some user's password is compromised then access can be allowed only after communicating some code to some third party where only the original user can approve and also access should be allowed only to registered and trusted devices

A08 - Software and Data Integrity Failures

The damage here can be caused in case of the software system is configured to download and update its component without some monitoring measures, then some malware can be installed in the system discretely.

A09 - Security Logging and Monitoring Failures

An extra measure to secure a software system is to configure some monitor-and-inform systems to prevent or report potential threats to the system.

A10 - Server-Side Request Forgery

The harm here is cause when the user-supplied URL is not being validated when fetching remote resources, then the URL can be falsified or/and forged and sent to some unexpected destination.

Conclusion

The software system of my web App is configured in a relatively securer way than before, by protecting some routes to allow communication between the client-side and server-side only when the user is authenticated and by checking the access level of the user.

The passwords are encrypted which means even if there is a data leak, the passwords cannot be easily read and used.

For certain user input, there is input validation in both client and server-side ends.

To conclude, I think my Web App is still insecure and very vulnerable as the security configurations were applied just recently straight after being learnt within the same week which means that they are not so solid because of the freshness of the learnt material and that there might be some unintended misconfigurations.

My future improvements to the Web App security are to keep updating the security system as my knowledge develops and grows bigger.