Security Analysis Document

Security analysis for U2.

Created by:

Petar Bakalov – 4634705

Contents

Versioning table 1

Introduction 1

Risks & Vulnerabilities 2

- Broken Access Control 2
- Monitoring and logging 2
- Overloading Traffic 3

Conclusion 3

Versioning table

Version	Date	Description
1.0	22/05/2024	Initialization of document

Introduction

In this document I will analyze some of the potential risks and vulnerabilities in U2, that could be exploited. Securing the application is crucial, as it uses personal

data like emails and passwords. Moreover, I will briefly discuss if I have solved the risks or how am I going to deal with them.

Risks & Vulnerabilities

Broken Access Control

This risk is about users having rights of different user group. In the case of "U2", this could be if a normal user has the rights of an admin. This can cause many issues, as the admin has the right to ban/delete user accounts as well as deleting videos.

This risk has been planned, and currently is being prevented, by handling user authorization using Auth0. Admin roles are assigned manually, while user roles are assigned on signup. The Auth0 database is secure and is handled by the company, therefore, breaking in the database and changing the user role has minimal possibility.

However, if the API endpoints are found, a normal user could call an admin action, without permission. This risk is prevented by securing the endpoints of the API. When a method is being called, the API requires access token for the user, and then extract the role and authorize the action.

In conclusion, broken access control is being handled by "U2" and the users are safe from that risk.

Monitoring and logging

Without proper monitoring, detecting and responding to security incidents in real-time is difficult. If they are noticed on time, proper actions could be taken to prevent them on time

This risk has been considered and will be dealt with when the application is deployed to Azure Cloud. Currently, there is a cluster, which is ready to be deployed, however, I am still investigating the pricing options of Azure. They provide many monitoring tools like overloading traffic for example.

Overloading Traffic

Overloading traffic attacks are performed, in order to crash the application, as it cannot handle the requests. It is a very common type of attack on many software solutions.

Microservice architecture solves the issue partially, as each service has a single responsibility, and many instances of them are created when the application is deployed. Currently, I have a gateway setup by Envoy Gateway, which works with Kubernetes clusters, and when deployed to Azure, a load balancer will be integrated.

Moreover, I plan to implement stress and load testing, for a predefined number of users that can use the application at a time. This will also ensure quality, after implementing new features, and make the application suitable for deployment.

Conclusion

In conclusion, risk mitigation is crucial when implementing enterprise software. Moreover, many of these risks should be considered in the beginning of development and even when creating the architecture.

Many of the risks regarding authentication, authorization and personal data protection in “U2” is being handled by Auth0, with their own database, which lowers the risk of data theft.

A big part of the risks will also be handled by Azure, using their tools, after deploying my cluster. I would also consider them secure and useful, when dealing with different types of miscellaneous attacks.