6/2/2023

OWASP report

Fontys University of Applied Sciences | ICT

Fabienne Leidekker

S3 SOFTWARE | VERSION 0.1

Version history

Version	Date	Author(s)	Changes
0.1	02-06-2023	Fabiënne Leidekker	Added OWASP report

Content

OWASP

Table

	Impact	Risk	Actions possible	Planned to be fixed
A1: Broken access control	Severe	High	The application checks if the object that is being changed is issued from the same user	Yes
A2: Cryptographic failures	Severe	Low	Only sensitive data id passwords and they are being encrypted	Yes
A3: Injection	Severe	High	Create parameterized queries	No, risk accepted
A4: Insecure design	Moderate	Moderate	Write unit and integration test to validate that all critical flows are resistant to the threat model	Yes
A5: Security misconfiguration	Minimal	Low	Logical error handling displayed to user	Yes
A6: Vulnerable and outdated components	Minimal	Low	Most up to date versions of the software is being used	Yes
A7: Identification and authentication failures	Moderate	Moderate	Password creation needs certain complexity, invalidate token when logout or a period of inactivity	Semi
A8: Software and data integrity failures	Minimal	Low	Libraries and dependencies used are from trusted repositories	Semi
A9: Security logging and monitoring failures	Minimal	Low	Ensure log data is encoded correctly	No, risk accepted
A10: Server-side request forgery	Severe	High	Disable HTTP, do not send raw responses to clients	No, risk accepted

Reasoning

There are a few risks that currently could harm my application. My app is not protected against SQL injections, so someone using the right input could cause harm to my system. I also have some identification and authentication failures. Password creation does not have any complexity, and I don’t have multi-factor authentication.

Conclusion

I think my application is moderately secured, but definitely in need if improvement. Some risks, like injection are quite important to take into consideration, because they can ruin up your whole

application. And with some other risks, I think I’m not fully protected against them. For example, I’m currently facing a bug where the access token doesn’t get destroyed when it’s expired.