CYBER SECURITY

SECURE MEDICAL SYSTEM

DUE TO 11/6

Alessandro Spinosi

Guillaume Kerckhofs

Samain Clément

Randi Dochot

Alix Declerck

Professor Romain Absil

Abstract. The goal of this project is to implement a secure client / server system handling patient's medical records, such as the ones available in hospitals.

We install un server using and existing tech combo (from Gerardo Fernandez) which is composed by docker to run the server, Nginx the server itself, Symfony 6.2 Boilerplate for the applicative framework and we change the database to mariadb

We create a client using npm run dev from the Node Package Manager

Contents

1. Project description

Installations procedures can be find here :
`https://github.com/Randi-Dcht/project_security_web`

2. Check-list

2.1. Confidentiality

- m.a.j todo

Do I properly ensure confidentiality?

- Are sensitive data transmitted and stored securely?
- Are sensitive requests sent to the server transmitted securely?
- Do I achieve end-to-end encryption (if relevant)?
- Does a system administrator have access to the sensible data of some arbitrary user?

2.2. Integrity

- m.a.j todo

Do I properly ensure integrity of stored data?

2.3. Non repudiation

- m.a.j todo
- Do I properly ensure non-repudiation?
- Do I use signature, certificates, a proper authority?

To use a certificat with a proper authority we need a domain name and we don't have that

(during the creation of a client we have to create a certificate request)

2.4. Strong authentification scheme

- m.a.j todo

Do I use a proper and strong authentication scheme?

- Do I follow OWASP guidelines?
- Is my authentication broken (cf. OWASP 10) [1]

2.5. Relying secrecy

- m.a.j todo

Do my security features rely on secrecy, beyond credentials?

2.6. Injection vulnerability

Am I vulnerable to injection?

- URL, SQL, Javascript and dedicated parser injections

2.7. Remanence attack vulnerability

- m.a.j todo

Am I vulnerable to data remanence attacks?

2.8. Replay attack vulnerability

- m.a.j todo

Am I vulnerable to replay attacks?

2.9. Fraudulent request forgery vulnerability

- m.a.j todo

Am I vulnerable to fraudulent request forgery?

2.10. Monitoring

- m.a.j todo

Am I monitoring enough user activity so that I can immediately detect maliciousor analyse an attack a posteriori?

Do I simply reject invalid entries, or do I analyse them?.

Can logs be falsified?.

2.11. Component security knowledge

- m.a.j todo

Am I using components with know vulnerabilities?

2.12. System

- m.a.j todo

Is my system updated?

2.13. Access control

- m.a.j todo

Is my access control broken (cf. OWASP 10) [1]? Do I use indirect references to resource or functions?

2.14. General security features

- m.a.j todo

Are my general security features misconfigured (cf. OWASP 10) [1]? Also, note that you will unlinkely graduate should you fail to

use a (at least self-signed) certificate for your server,.

use a framework (at least for the server's side),.

achieve end-to-end encryption (if relevant)..

The end to end encryption is handled

References

[1] "Owasp guidelines." [Online]. Available: `https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/`

UMONS, Belgium
Email address: `alessandro.spinosi@student.umons.ac.be`

UMONS, Belgium
Email address: `guillaume.kerckhofs@student.umons.ac.be`

UMONS, Belgium
Email address: `samain.clement@student.umons.ac.be`

UMONS, Belgium
Email address: `randi.dochot@student.umons.ac.be`
URL: `www.dochot.be`

UMONS, Belgium
Email address: `alix.Declerck@student.umons.ac.be`

UMONS, Belgium
Email address: `romain.absil@umons.ac.be`