5-1-2022

Doomen,Jens J.J.W.

Research Rapport

Inhoud

[Inleiding 2](#_Toc92791450)

[Hoe maak ik mijn applicatie beter beveiligd? 2](#_Toc92791451)

[Wat zijn de beste manieren om je applicatie veiliger te maken? 2](#_Toc92791452)

[Available product analysis 2](#_Toc92791453)

[Code review 2](#_Toc92791454)

[Wanneer wordt een applicatie als veilig beschouwd? 3](#_Toc92791455)

[Literature study 3](#_Toc92791456)

[Root cause analysis 3](#_Toc92791457)

[Wat zijn de risico’s als het gaat om beveiliging van de applicatie. 4](#_Toc92791458)

[Literature study 4](#_Toc92791459)

[Unit test 4](#_Toc92791460)

[Conclusie 5](#_Toc92791461)

[Bronnen 5](#_Toc92791462)

# Inleiding

Ik was door mijn applicatie aan het lopen en zag dat er niet heel veel voor security was gedaan. Het enige wat er een beetje veilig uitzag was het inloggen en registreren. Het is natuurlijk veel fijner om te weten dat de site waar je op veilig is en goed beveiligd.

# Hoe maak ik mijn applicatie beter beveiligd?

## Wat zijn de beste manieren om je applicatie veiliger te maken?

### Available product analysis

Er zijn heel veel verschillende manieren om je applicatie veilig te maken. Het belangrijkste en misschien wel een van de simpelere dingen is het schrijven van veilige code. Om code veilig te schrijven moet je kijken naar bugs, ook is het vermijden van dingen hardcode beter en wachtwoorden moeten worden geëncrypt. Als de code regelmatig wordt getest en zo word geschreven dat het snel en agile kan worden aangepast is het ook veel veiliger. Een ander punt om op te letten is het gebruiken van libraries. Er zitten er namelijk tussen die niet heel goed beveiligd zijn. Hetzelfde geld ook voor APIs, maak alleen gebruik van geautoriseerde APIs. Tot slot is het beter om gebruik te maken van authenticatie met een token.

(TRIPWIRE, 14-02-2018)

### Code review

Zoals net geconstateerd is het belangrijk om nette, schone en veilige code te schrijven. Om dit te kunnen controleren heb ik een code review uitgevoerd op zowel mijn front-end als back-end. Dit heb ik bereikt door via sonarcloud een scanner uit te laten voren. Deze scanner is wordt ook met elke nieuwe push naar git opnieuw uitgevoerd. Zo kan ik zien of mijn code ook echt verbeterd is. Op dit moment heeft mijn front-end 5 bugs en 53 code smells met 2 security hotspots. Dit is dus niet wat je wilt zien en nu weet ik waar ik nog kan werken aan mijn front-end. Mijn back-end ziet er wat beter uit, deze heeft namelijk op dit moment 0 bugs en 20 code smells met 4 security hotspots. Dit betekent dat ik ook hier op het gebied van security nog wel wat werk kan gaan doen.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## Wanneer wordt een applicatie als veilig beschouwd?

### Literature study

Het is erg lastig om aan te geven wanneer de applicatie veilig is. Dit komt omdat het nooit helemaal veilig kan zijn. Maar er zijn wel een paar dingen die je kunt zeggen over je app en hieruit kun je een conclusie trekken. Een applicatie is veilig als de gebruiker toegang krijgt als met de inlog gegevens die ook echt van de gebruiker zijn. Verder kan deze gebruiker dan natuurlijk alleen bij dingen die bij hem/haar horen. Denk hierbij aan het aanpassen van het account. Het is natuurlijk de bedoeling dat de gebruiker alleen zijn of haar eigen account kan aanpassen. Het is ook veilig als de ingelogde gebruiker die data achterlaat op de app alleen geëncrypte data achterlaat. Het is ook verstandig om activiteiten te loggen. Op deze manier kun je snel traceren wie er verantwoordelijk is bij een breuk op de security. Als er voldoende testen zijn geschreven wordt de applicatie ook als veilig beschouwd.

(VMware)

### Root cause analysis

Om te kunnen zien of mijn applicatie als veilig wordt gezien wil ik de 5 waarom vragen gaan stellen.

* Waarom is mijn applicatie nog niet veilig
* Omdat de data niet geëncrypt is. En er niet gelogd wordt wat er wordt gedaan.
* Waarom is de niet geëncrypte data niet veilig?
* Omdat nu de data van andere makkelijk te bereiken is.
* Waarom is het niet loggen van activiteiten niet veilig?
* Omdat nu iedereen kan doen wat hij wil en er niet achterhaald kan worden wie iets heeft gedaan wat niet toegestaan is.
* Waarom worden andere onderdelen wel als veilig gezien?
* Omdat er genoeg getest is en er een veilige authenticatie is.
* Waarom wordt met wel veilige dingen en onveilige componenten toch de hele app als onveilig gezien?
* Omdat over het algemeen de onderdelen die verbetert kunnen worden met security iets te groot zijn om goed te kunnen zeggen dat het veilig is.

## Wat zijn de risico’s als het gaat om beveiliging van de applicatie.

### Literature study

Er zijn ook lichte risico’s als het gaat om de beveiliging van een applicatie. Denk hierbij aan het gebruik van een firewall. Natuurlijk is het goed om deze te hebben en zijn ze zeker handig om te hebben. Toch hebben ze de neiging om valse informatie te geven. Het zou zomaar kunnen voorkomen dat er valse positieve of negatieve informatie wordt gegenereerd. Ook kost het veel om firewalls bij te kunnen houden. Ook kan er een risico liggen bij het reviewen van code. In mijn situatie waar ik alleen aan de code werk kan het voorkomen dat op een code review ik niet objectief ernaar kijk omdat het mijn eigen werk is. Om dit te voorkomen is het beter om een 3e ernaar te laten kijken of het automatisch te laten lopen. Een ander risico kan ook liggen in het feit dat er geen “fout” pad is geprogrammeerd. Als de gebruiker iets doet wat niet de bedoeling is kan dit heel de applicatie breken als er geen code is om dit gevaar op te vangen.

(sqreen, Setter, 01-04-2021)

### Unit test

Om deze “Foute route” te kunnen tegen houden en ervoor te zorgen dat de applicatie blijft werken. Wil ik Unit testen gaan schrijven. Maar hoe wil ik het precies gaan testen? Stel ik pak mijn authenticatie systeem. Hier kan de gebruiker toch een aantal dingen doen die natuurlijk niet zouden moeten mogen. Denk bijvoorbeeld aan het inloggen, de gebruiker zou kunnen proberen om in te loggen zonder alles in te vullen. Naast alle juiste wegen heb ik ook deze testen geschreven. Op deze manier kan ik er voor zorgen dat in de toekomst dit niet meer voorkomt. Na het schrijven zijn alle testen die gaan over mijn authenticatie systeem geslaagd.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## Conclusie

Als conclusie kan ik zeggen dat op sommige plekken mijn applicatie beveiligd is maar dat op andere plekken nog wel wat meer gedaan kan worden. De plekken waar nog werk gedaan moet worden is het encrypte van data en het loggen van de users. Het userloggen is iets wat ik misschien nog wel kan doen in de beperkte tijd die nog in het semester zit. Zo is toch de applicatie wat veiliger. Ik weet nog niet precies hoe ik dit wil gaan aanpakken maar hier kan ik nog wat onderzoek over doen.

## Bronnen

<https://www.tripwire.com/state-of-security/security-awareness/top-mobile-app-security-best-practices-developers/>

<https://www.vmware.com/topics/glossary/content/application-security.html>

<https://blog.sqreen.com/best-practices-build-secure-applications/>

<https://digital.ai/glossary/web-application-security>