OWASP TOP 10 Notify

Luc Janssen

# Wat is OWASP?

OWASP (Open Web Application Security Project) is een internationale non-profitorganisatie die openlijke beschikbare artikelen, methodologieën, documentatie, tools en technologieën produceert binnen de sector van web applicatie security.

# Wat is OWASP Top 10?

De OWASP Top Ten is een artikel wat bedoeld is om bewustzijn op te wekken over applicatie beveiliging door de 10 grootste security risico's in kaart te brengen. De aanbeveling van OWASP is dat bedrijven dit document zouden moeten adopteren om goede security te kunnen realiseren door security issues te mitigeren.

Volgens veel verschillende bronnen is het gebruik van de OWASP top 10 de meest effectieve stap om de development culture binnen een bedrijf te transformeren naar een die meer secure code produceert.

# OWASP TOP 10

[**A01:2021-Broken Access Control**](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)

Broken Access Control heeft als effect dat gebruikers kunnen handelen buiten de bedoelde machtigingen. Dit kan op zijn beurt leiden dat gebruikers acties kunnen uitvoeren of data kunnen bemachtigen waar hij of zij geen recht op hebben. Denk hierbij aan openbaarmakingen van data, wijzigen of vernietigingen van gegevens.

Broken access control komt voor bij 3.81% van de geteste applicaties.

De volgende stappen zijn ondernomen om deze problemen te mitigeren:

* Standaard uitstaan van directory listing
* API Endpoints beveiligd met “Abilities” in de accessTokens (Laravel Sanctum) waarbij iedere rol bepaalde abilities heeft. Zonder accessToken met de juiste ability is een endpoint niet bereikbaar
* API Rate limits (Request Throttling) zodat de api niet constant aangeroepen of aangevallen kan worden.
* Logging van failures

[**A02:2021-Cryptographic Failures**](https://owasp.org/Top10/A02_2021-Cryptographic_Failures/)

Cryptographic Failures is een ontzettend breed onderwerp en heeft veel verschillende oorzaken. Het effect is dat (sensitive) data niet goed versleuteld is waardoor het makkelijk ontcijferd kan worden.

Broken access control komt voor bij 3.81% van de geteste applicaties.

De volgende stappen zijn ondernomen om deze problemen te mitigeren:

* Rekening gehouden met GDPR
* Geen gebruik van onveilige encryptie of hashing algoritmes zoals SHA1 of MD5
* Geen gebruik van onveilige protocollen zoals SMTP of FTP
* Password en salt met laravel algoritme
* Peer reviewing van configuraties en settings

[**A03:2021-Injection**](https://owasp.org/Top10/A03_2021-Injection/)

Injectie aanvallen kunnen op veel verschillende manieren voorkomen. De kern van het probleem ligt rond het feit dat een aanvaller expres data aan het programma probeert te passeren als onderdeel van een query die dan anders uitgevoerd wordt en gevoelige data terug kan returnen.

De volgende stappen zijn ondernomen om deze problemen te mitigeren:

* Gebruik van LIMIT in queries
* Gebruik van Laravel Eloquent ORM. Escaped alle input gegeven aan de query builder.
* Laravel server side input validatie
* Gebruik van exec() vermeden

[**A04:2021-Insecure Design**](https://owasp.org/Top10/A04_2021-Insecure_Design/)

Insecure design is een brede categorie die ontzettend veel zwakke punten overkoepeld. Hoewel dit niet kan worden opgelost door een perfecte implementatie, is het wel mogelijk om in de designfase goed na te denken over de bescherming tegen specifieke aanvallen.

De volgende stappen zijn ondernomen om deze problemen te mitigeren:

* Integratie en Unit tests maken
* Limiteer/verspreid resource gebruik m.b.v queues.

[**A05:2021-Security Misconfiguration**](https://owasp.org/Top10/A05_2021-Security_Misconfiguration/)

Security misconfiguration gaat over de (opzet van) de configuratie van de beveiliging.

De volgende stappen zijn ondernomen om deze problemen te mitigeren:

* Geen gebruik van onnodige services
* Geen onnodige ports open zetten
* App debugging disabled in de API configuratie

[**A06:2021-Vulnerable and Outdated Components**](https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/)

Vulnerable and Outdated Components zijn services, libraries of onderdelen die gebruikt worden binnen de applicatie die out of date of kwetsbaar zijn. Dit heeft als effect dat een aanvaller op deze onderdelen kan checken en door een zwakke plek waar je geen controle over hebt.

De volgende stappen zijn ondernomen om deze problemen te mitigeren:

* Verwijderen van ongebruikte dependencies, onnodige functies, componenten en bestanden.
* Onderzoek doen naar te gebruiken libraries en deze monitoren
* Libraries en services uit officiële bronnen gebruiken

[**A07:2021-Identification and Authentication Failures**](https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/)

Het identificeren van een gebruiken en de authenticatie om dit te controleren is essentieel. Zonder dit onderdeel zou een aanvaller kunnen handelen uit naam van een andere gebruiker.

De volgende stappen zijn ondernomen om deze problemen te mitigeren:

* Brute forcing voorkomen met api request limiting
* Geen “out-of-the-box” credentials
* Minimum password lengte
* Server side generated AccessTokens met beperkte lifetime

[**A08:2021-Software and Data Integrity Failures**](https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/)

Software and Data integrity failures slaan op code en de infrastructuur die geen bescherming bieden tegen integriteitsschendingen. Een insecure CI/CD pipeline, modules uit onvertrouwde bronnen, etc. .

De volgende stappen zijn ondernomen om deze problemen te mitigeren:

* Libraries en dependencies komen uit vertrouwde repositories
* CI/CD workflows opgezet zonder hardcoded credentials (Github secrets)
* Gebruik van Sonarlint en Sonarcloud tijdens het development process om bekende vulnerbilities te kunnen ontdekken

[**A09:2021-Security Logging and Monitoring Failures**](https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/)

Security logging en monitoring gaat over het bewust zijn van breaches en het kunnen reageren hierop.

De volgende stappen zijn ondernomen om deze problemen te mitigeren:

* Alle requests worden gelogged
* Logging wordt na een bepaalde tijd verwijderd
* Log gegevens worden correct opgeslagen zodat derde partijen/aanvallers de log niet kunnen inzien

[**A10:2021-Server-Side Request Forgery**](https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_(SSRF)/)

Server-side Request Forgery komt voor wanneer een webtoepassing een externe bron ophaalt zonder dat de URL gevalideerd wordt. Hierdoor kan een aanvaller een verzoek doen naar een onverwachte bestemming zelfs als er een VPN op firewall aanwezig is ter bescherming.

Dit zijn de stappen die ik heb ondernomen om de risico’s te minimaliseren:

* Validatie van user input
* Redirects disabled
* Speciale response models returned naar de user