**Rapport security code review:**

**Voorbereiding**:

Gebruik campagneLaravel.zip en campagne.sql.

Pak de zip uit en voeg de vendor-map toe met het commando “composer update”.

Maak database campagne aan en vul deze met het sql-bestand.

Start de applicatie.

**Opdracht:**

Controleer de code van Campagne op kwetsbaarheden op een aantal aspecten:

* Configuratie
* SQL injectie
* XSS
* Encryptie
* CSRF

Extra

Controleer de code van Campagne op kwetsbaarheden op een aantal aspecten:

* Autorisatie / access control

**Inleveren:**

Geef in de onderstaande tabellen aan

* of kwetsbaarheden aanwezig zijn op deze aspecten
* zo mogelijkheden in welke bestanden op welke regelnummers
* mogelijke beschermingsmaatregelen

Lever deze in op itslearning

|  |  |
| --- | --- |
| **Configuratie** (verkeerde instellingen, extensies, leesrechten, etc.?) | |
| Welke kwetsbaarheden zijn er? | 1. Debug staat op true. Daardoor kan de gebruiker sql error zien als hij onjusite id in de url typt en dat mag een gebruiker niet zien. 2. Inloggegevens zijn niet veilig. 3. Een gebruiker kan meerdere blogs verwijderen. |
| In welke bestand(en) en op welke regel(s) | 1. .env file regel 4. 2. .env file regel 15 en 16. 3. Nieuws.blade.php regel 14 t/m 20 |
| Welke tegenmaatregelen  kun je treffen? | 1. Debug mode moet op false staan. 2. Inloggegevens moeten velig zijn (dus geen root gebruiken). 3. Bepaalde recht geven aan elke gebruiker. |
|  | 1. Toepassen zodat de gebruiker maar bepaalde files kan uploaden zoals jpg, png of pdf bestanden. 2. Ervoor zorgen dat de gebruiker sommige tekens die kwetsbaar kunnen zijn niet mag gebruiken. |

|  |  |
| --- | --- |
| **SQL DB injectie** (kan de gebruiker met bepaalde invoerdata database-queries of database-acties veranderen?) | |
| Welke kwetsbaarheden zijn er? | Geen omdat er gebruik wordt gemaakt van ORM |
| In welke bestand(en) en op welke regel(s) | X |
| Welke tegenmaatregelen  kun je treffen? | X |

|  |  |
| --- | --- |
| **XSS** (kan de gebruiker JavaScript of i.d. te laten uitvoeren of laten opslaan?) | |
| Welke kwetsbaarheden zijn er? | 1. De gebruiker kan bestanden zoals javascript bestanden uitvoeren. 2. De gebruiker kan script code in de tekstvakken invoeren. |
| In welke bestand(en) en op welke regel(s) | 1. CampagneController regel 79 t/m 85 |
| Welke tegenmaatregelen  kun je treffen? | 1. Filteren |

|  |  |
| --- | --- |
| **Encryptie**  (is de versleuteling sterk of zwak, is de versleuteling te breken of te omzeilen?) | |
| Welke kwetsbaarheden zijn er? | 1. De hacker kan makkelijk de wachtwoorden raden. 2. De hacker kan makkelijk in de database komen |
| In welke bestand(en) en op welke regel(s) | 1. Login.blade.php regel 11 t/m 17 2. Env file regel 4 |
| Welke tegenmaatregelen  kun je treffen? | 1. Eisen instellen op het wachtwoord (Wachtwoord mag niet hetzelfde zijn of lijken op de gebruikersnaam) 2. Geen MD5 gebruiken is niet veilig 3. Database inloggegevens veranderen (Geen root) |

|  |  |
| --- | --- |
| **CSRF**  (wordt een http-request vanaf een ander domein geaccepteerd?) | |
| Welke kwetsbaarheden zijn er? | 1. De gebruiker of de hacker kan makkelijk zien wat voor data door de sever worden gestuurd. |
| In welke bestand(en) en op welke regel(s) | 1. Alle views pagina’s |
| Welke tegenmaatregelen  kun je treffen? | 1. Ervoor zorgen dat de website beveiligd wordt door @CSRF en Token te gebruiken. |

**EXTRA**

|  |  |
| --- | --- |
| **Autorisatie / access control**  (kan de gebruiker bepaalde acties uitvoeren zonder de juiste gebruikersrechten?) | |
| Welke kwetsbaarheden zijn er? |  |
| In welke bestand(en) en op welke regel(s) |  |
| Welke tegenmaatregelen  kun je treffen? |  |