**A1:2017-Injection**

* Gebruik SQL-commando’s om SQL-injection te voorkomen
* Valideer binnenkomend data

**A2:2017-Broken Authentication**

* Geen zwakke wachtwoorden toestaan (nog implementeren in devolop)
* ~~Data encrypten voordat verstuurd wordt~~ (toegepast)
* Maximaal aantal login pogingen (nog implementeren in devolop)

**A3:2017-Sensitive Data Exposure**

* ~~Geen standard encryptie toepassen. Gebruikt up to date encryptie sleutels en algoritmes~~ (toegepast)

**A4:2017-XML External Entities (XXE)**

* Valideren van Xml-bestanden/ Weigeren van toestemming om Xml-bestanden te uploaden

(Wij gebruiken geen XML dus in hoeverre moet dit toegepast worden?)

**A5:2017-Broken Access Control [Merged]**

* Zorg da teen specifiek iemand gegevens kunnen aanpassen aan de database Zorg dat alle toegangscontrole gemonitord en gelogd worden
* Invalideer JWT-tokens wanneer gebruikers uitloggen (toegepast?)

**A6:2017-Security Misconfiguration**

* Gebruik geen HTTP
* Gebruik scanners om de zwakheden van je applicatie te vinden

**A7:2017-Cross-Site Scripting (XSS)**

* Een framework die beschermt tegen XSS
* Input valideren door middel van antiforgerytokens
* Niet vertrouwde http requests niet afhandelen

**A8:2017-Insecure Deserialization**

* Logging toevoegen zodat je kan traceren wat er is gebeurt

**A9:2017-Using Components with Known Vulnerabilities**

* ~~Gebruik alleen officiële componenten~~ (toegepast)
* ~~Geüpdatet versie van componenten~~ (toegepast)

**A10:2017-Insufficient Logging & Monitoring**

* Zorg dat logs gegenereerd worden zodat deze logs door een gecentraliseerd logbeheersysteem gebruikt kan worden.
* Het systeem zo monitoren dat er een alarm af gaat wanneer verdacht gedrag wordt gedecreteerd
* Zorg dat er een plan van herstel beschikbaar is indien iets fout is