**Welke veiligheidsrisico’s bedreigen je applicatie en wat is de impact van tegenmaatregelen?**

**De risico’s**

Er brengen zich veel risico’s mee met het opbouwen van een applicatie. Denk hierbij aan data leaks, defacing, malware en ransomeware. De beveiliging van een applicatie is dan ook zeer belangrijk. Het is belangrijk om geen oude libraries en/of software te gebruiken. Veel updates zijn niet alleen om de software te optimaliseren of nieuwe functies uit te brengen, maar ook voor de security. In Apache Tomcat is het erg belangrijk om onjuiste configuratie te voorkomen. De standaard installatie kan altijd verbeterd worden door middel van aanpassingen of add-ons.

Bij het gebruik van postgres is er risico op SQL Injection. Wanneer er sprake is van input van de gebruiker (bijvoorbeeld formulieren) kan een aanvaller de gegevens manipuleren om zo code mee te sturen in het formulier. Deze code wordt dan uitgevoerd door de database database en kan negatieve gevolgen met zich meebrengen.

HTTP is ook een veel gebruikte methode voor aanvallen. Met HTTP verzoeken zijn er veel mogelijkheden om aanvallen uit te voeren op de applicatie. Denk hierbij bijvoorbeeld aan XSS(Cross-Site Scripting) en CSRF(Cross-Site Request Forgery).

**Voorkomen**

Wat het allerbelangrijkste is om mee te beginnen is kijken of alle software up to date is. Hieronder valt Java Runtime Environment(JRE), Apache Tomcat en third-party libraries.

Omdat er gewerkt wordt met een postgres JDBC , moet dit zo veel mogelijk afgesloten worden naar de open wereld. Hier kunnen restricties opgezet worden vanuit de database. Bijvoorbeeld dat je alleen kunt connecten met één IP-adres of een range van IP-adressen. Daarnaast kunnen er meerdere gebruikers aangemaakt worden voor andere taken, zodat er niet één master user is waar gebruikt van wordt gemaakt die overal bij kan. Er kan ook gebruik worden gemaakt van environment variables om hard-coded login gegevens te voorkomen.

Om te beschermen tegen SQL Injection kan er gebruik worden gemaakt van prepared statements. Zo worden de gegevens niet hardcoded meegegeven, maar geïnsert in het statement.

Voor HTTP is het belangrijk om gebruik te maken van een SSL(Secure Sockets Layer)/TLS(Transport Layer Security). Dit betekent dat er gebruikt wordt gemaakt van encryptie tussen client en server gegevens. In Tomcat kan er gebruik worden gemaakt van enforced HTTPS, zodat alle transacties in Tomcat via HTTPS verlopen in plaats van HTTP.