# Inleiding

In dit onderzoek ga ik toelichten wat voor veiligheidsrisico’s er zijn binnen mijn applicatie en hoe ik dit het beste kan oplossen. Ik wil graag de CI/CD op mijn lokale netwerk beheren en ga dit dan ook onderzoeken hoe dat het beste kan.

# Onderzoeksvraag

Op welke manier kan ik CI/CD op een lokaal netwerk gebruiken en toch toegang hebben tot github zodat ik alles op een eigen server kan beheren?

## Hoe zit mijn lokaal netwerk in elkaar?

Er zijn verschillende apparaten verbonden in het netwerk. Deze apparaten kunnen allemaal met elkaar communiceren omdat deze op hetzelfde netwerk zitten. Zo ook een server wat in eerste instantie opgezet was als NAS. Een NAS is een soort server waar data op gedeeld kan worden met de rest van het netwerk. Deze NAS heb ik uiteindelijk verder ingericht zodat deze als meer dient dan alleen een gedeelde dataopslag. Er draait nu ook Jenkins en SonarQube op. Jenkins voor de CI/CD en SonarQube voor het analyseren van de code.

Verder zijn er geen poorten geopend die direct een toegang tot het netwerk toelaten.

## Hoe kan GitHub communiceren met een netwerk achter een firewall?

Omdat Jenkins draait achter een firewall, dat ervoor zorgt dat niemand het netwerk kan betreden dat daar niet toe bestemd is, kan GitHub niet een sein sturen naar Jenkins om aan te geven dat er een vernieuwde versie van een branch beschikbaar is. Het is mogelijk om Jenkins bereikbaar te maken voor GitHub door gebruik te maken van ngrok. Met ngrok maak je een eigen lokale server die verbinding maakt met ngrok.io. Het idee hier van is dat \*.ngrok.io de push event ontvangt en doorstuurt naar de ngrok server dat is gehost op het lokale netwerk.

GitHub heeft hier iets op verzonnen, namelijk smee.io. Dit is een dienst waarbij een unieke code gegenereerd kan worden via smee.io, deze code moet dan ingevuld worden in een client van smee.io. Ook moet er via GitHub een webhook worden aangemaakt met de link van smee.io waar alle push triggers naar toe gestuurd worden. De smee client zal alle push triggers doorsturen naar de /github-webhook/ endpoint van Jenkins. Hierdoor weet Jenkins dat er een push plaatsgevonden heeft op GitHub en zal Jenkins de code van GitHub halen om de build te starten.

Ditzelfde geldt voor webhookrelay.com, deze biedt ongeveer dezelfde oplossing. Er wordt aangemeld op een URL waar berichten over worden verstuurd, net als via smee gebeurt dit doormiddel van de GitHub Webhook.

(Creating webhooks, 2022)

(Neale, 2019)

(Production-ready Github and Jenkins setup behind a firewall, 2022)

## Hoe kan Jenkins communiceren met applicaties buiten het lokaal netwerk?

De firewall is zo geconfigureerd dat uitgaande verkeer is toegestaan, maar inkomend verkeer wordt tegengehouden. Hierdoor kan GitHub niet in het lokaal netwerk komen maar Jenkins wel naar GitHub om zo de repository aan te kunnen voor de pipeline.

## Hoe kan ik mijn lokale netwerk benaderen zonder security risico’s op te lopen?

Het is zo dat het lokale netwerk in principe niet te benaderen is vanuit buiten het netwerk. Hier heb ik een oplossing voor gevonden, namelijk een VPN. Om Jenkins en SonarQube te kunnen bereiken heb ik een VPN nodig, aangezien alle poorten dicht staan in het huidige netwerk. Aangezien WireGuard een plugin is dat bestaat in de plugin bibliotheek van de software van de NAS die ik gebruik, leek me dat een goede oplossing.

(chente, 2022)

# Smee.io

Uiteindelijk heb ik de keuze gemaakt om smee.io te gebruiken, echter is dit niet de veiligste optie omdat hier geen authenticatie voor nodig is. Op het moment dat iemand dezelfde ‘channel’ gebruikt zal hier data over gestuurd worden die ik gebruik om de GitHub Webhook te verbinden. Omdat dit wel een oplossing is die goed werkt en er geen data over gestuurd wordt dat informatie bevat wat niet openbaar mag worden maak ik er tot nu toe nog gebruik van. Op het moment dat ik het in gebruik zal nemen zal ik een andere oplossing moeten gebruiken. Webhookrelay maakt namelijk wel gebruik van authenticatie wat dus een goede vervanger zou zijn.

Zoals hieronder te zien is wordt er een verzoek gedaan naar de smee URL die ik in gebruik heb, deze wordt doorgestuurd naar mijn lokale netwerk waar de Jenkins omgeving op draait waarbij Jenkins vervolgens de repository ophaalt en gaat bouwen.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

(smee.io, 2022)

# Webhookrelay.com

Webhookrelay.com maakt gebruik van dezelfde principe als smee.io alleen werkt met een beveiligde verbinding. Om hier gebruik van te maken is het de bedoeling dat er tokens worden aangemaakt, dit moet gedaan worden met de CLI van webhookrelay. De token die gegenereerd wordt is de token waarmee de verbinding wordt gemaakt met webhookrelay.com door de client.

(Production-ready Github and Jenkins setup behind a firewall, 2022)

# Conclusie

In dit onderzoek ben ik tot de conclusie gekomen dat er veiligere oplossingen zijn voor het gebruiken van een server op een lokaalnetwerk omdat de oplossing die ik nu gebruik niet gebruik maakt van authenticatie waardoor iedereen die de URL weet de push events kan onderscheppen. Echter is dit geen probleem voor de veiligheid op dit moment omdat het nog in development fase is. Op het moment dat ik het verder uit wil werken naar productie zal ik de aanpak moeten aanpassen naar de bovengenoemde oplossingen.