THOMAS MORE

SOC MPI Oosterlo VZW

Realisatiedocument

Michiel Kuyken
Student Bachelor in de Elektronica-ICT – Cloud & Cyber Security

2024 - 2025

Inhoudsopgave

1. Inleiding

Voor het laatste onderdeel van de opleiding Elektronica-ICT met afstudeerrichting Cloud & Cybersecurity, moesten we stagelopen bij een bedrijf of organisatie. Ik heb de kans gekregen om aan de slag te gaan bij het MPI Oosterlo VZW. In deze instelling ondersteunen ze jongeren en volwassenen met mentale beperkingen. Ze werken hier dus met gevoelige informatie over cliënten en het is net deze informatie die nuttig kan zijn voor aanvallers. Om dit te beveiligen hebben ze aan mij gevraagd om een Security Operations Center (SOC) op te stellen.

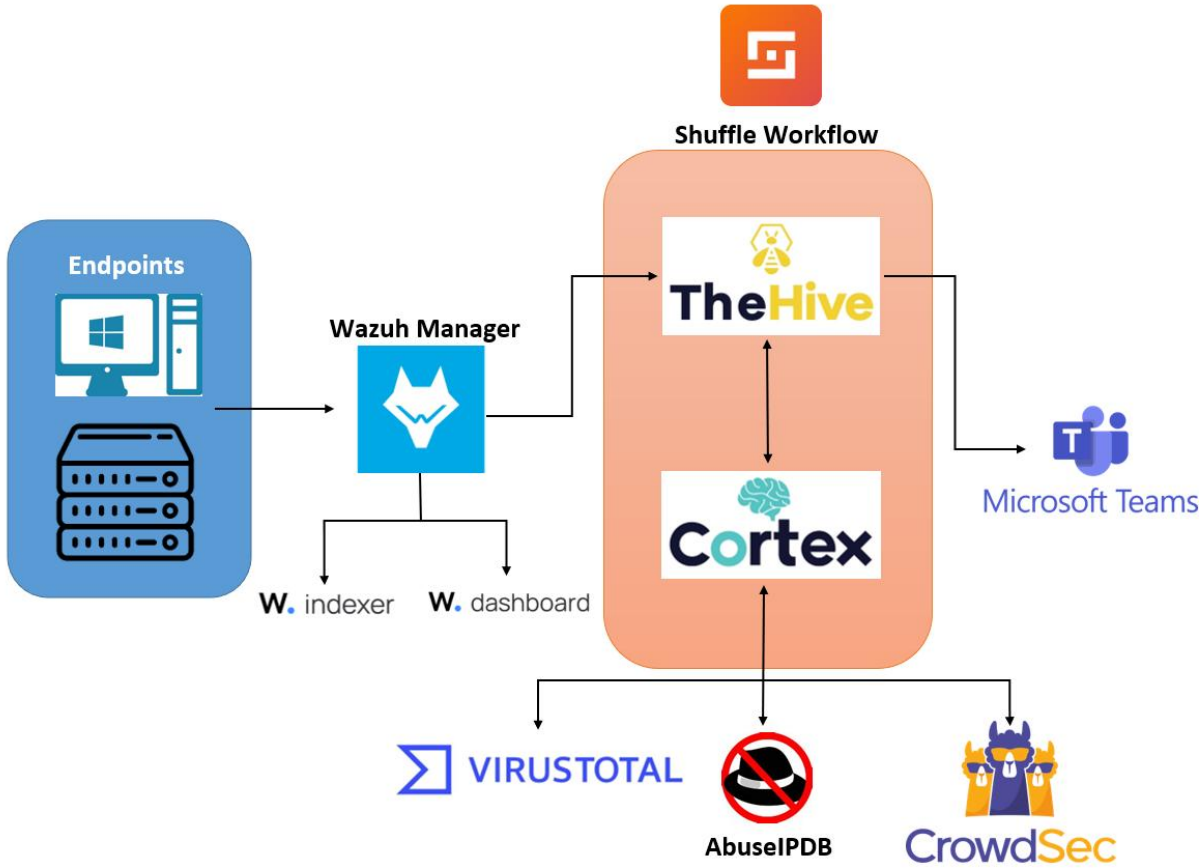
Binnen het MPI Oosterlo VZW was er tot nu toe nog geen centraal beveiligingssysteem dat op hun servers beveiligingsincidenten detecteert, analyseert en opvolgt. Hierdoor kunnen incidenten snel door de vingers glippen en achterdeuren openblijven voor aanvallers. Dit brengt natuurlijk risico's mee op het vlak van data- en netwerkbeveiliging. De hoofdopdracht lag dan ook bij het voorzien van een centraal dashboard waarop incidenten weergegeven worden. Op dit dashboard moeten ze CVE's, common vulnerabilities and exposures, kunnen bekijken en hiervan meldingen krijgen op Teams. Er werd ook nadrukkelijk gevraagd om de mogelijkheden met Wazuh te bekijken en een mogelijke integratie met Crowdsec te voorzien. Het belangrijkste was om de kosten zo laag mogelijk te houden. Dit betekende dus zoveel mogelijk open-source tools gebruiken.

In de drie maanden die ik aan mijn SOC heb kunnen werken, hebben het team en ik meerdere malen de scope van het project moeten aanpassen. Met elke stap die we namen zijn we verder gaan verfijnen wat nuttig was voor de organisatie en wat niet. Uiteindelijk bleven we over met onderstaande tools en workflow.

Een alert wordt gedetecteerd met Wazuh en deze wordt naar Shuffle gestuurd, waar er vervolgens een workflow start. Deze workflow maakt alerts en cases aan in TheHive en voegt hier observables aan toe. Deze worden gecontroleerd door analyzers die zich in Cortex bevinden. Ten slotte krijgt het IT-team een melding in een Teams-chat die vertelt wat er aan de hand is.

In dit document beschrijf ik voor u, de lezer, de volledige uitwerking van dit project. Ik begin met de analyse van de verschillende tools die ik gebruikt heb. In deze analyse komt u per tool te weten wat het doet, de mogelijkheden die het biedt voor het SOC en waarom de keuze voor deze tool gemaakt is.
Vervolgens bespreek ik de realisatie van het project, met alle overwinningen en obstakels die ik onderweg tegenkwam. Elke tool wordt hier tot in detail besproken. Het begint altijd met de installatie- en configuratieprocedure. Denk hierbij aan alle componenten die geïnstalleerd moeten worden en configuratiebestanden die aangepast moeten worden. Deze bestanden worden ook uitgebreid uitgelegd, wat in vele gevallen regel voor regel is, om een zo duidelijk mogelijk beeld te scheppen.
Hierna wordt er besproken hoe ik de tool heb gebruikt in het SOC en hoe deze geraadpleegd kan worden.
In het slot blik ik kort terug naar wat ik heb neergezet. Ik haal hier ook aan welke stappen MPI Oosterlo nog kan nemen in de toekomst om het project verder te optimaliseren.

Als rode draad doorheen dit document, heb ik onderstaand schema gemaakt. Hierop staat de weg afgebeeld die een alert aflegt van het moment dat het gedetecteerd wordt op een endpoint, tot de melding in Teams.



Figuur 1: SOC-schema

2. Analyse

Om een goed SOC op te kunnen zetten, is een goede analyse cruciaal. De tools die gebruikt worden moeten afgewogen worden om de beste keuzes te maken. In dit hoofdstuk wordt er een grondige analyse gedaan van de verschillende tools die in het SOC gebruikt zijn. Het doel hiervan is om een duidelijk beeld te geven van wat de tool is, de functionaliteiten die het biedt en waarom deze tool gebruikt wordt.

Om dit hoofdstuk zo overzichtelijk mogelijk te houden, wordt het SOC ingedeeld in zijn grootste componenten. Dit zijn het SIEM-gedeelte, het SOAR-gedeelte en Threat Intelligence. Wat deze delen exact inhouden, wordt per onderdeel duidelijk gemaakt. De gebruikte tools zijn op deze structuur ingedeeld.

2.1. SIEM

SIEM staat voor Security Information and Event Management en is een essentieel onderdeel van een SOC. Het stelt organisaties in staat om beveiligingsincidenten tijdig te identificeren, te beoordelen en erop te reageren. Een SIEM is een combinatie van Security Incident Management en Security Event management dat nu als één systeem samenwerkt.

SIM richt zich op het verzamelen en bewaren van logs en alerts over een langere periode. Deze alerts kunnen afkomstig zijn van verschillende bronnen zoals servers of firewalls. Het zorgt ervoor dat analisten de alerts achteraf kunnen onderzoeken en patronen kunnen herkennen.

SEM legt de nadruk op het monitoren van events om bedreigingen proactief op te sporen. Als er een verdacht patroon herkend wordt, genereert het automatisch een alert. Dit alert wordt dan op een dashboard getoond voor de gebruiker.

In dit onderdeel komt de tool Wazuh aan bod. Dit is gebruikt als SIEM-oplossing in het SOC en is één van de meest gebruikte open-source SIEM-platformen. Er wordt uitgelegd wat Wazuh doet, de functionaliteiten die het heeft en waarom het gebruikt wordt.

2.1.1. Wazuh

Wazuh is een open-source SIEM-platform dat bedrijven en organisaties helpt bij het monitoren, detecteren en reageren op beveiligingsincidenten. Het doet dit door het verzamelen van logs op de verschillende endpoints in de IT-infrastructuur. Wazuh analyseert deze logs op verdachte activiteiten, zoals brute-force attacks of bekende kwetsbaarheden, en creëert hieruit alerts. Deze alerts worden geclassificeerd op basis van ernst. Een alert met lage ernst, zoals bijvoorbeeld een succesvolle aanmelding, krijgt een laag meldingsniveau toegewezen, terwijl een alert met hoge ernst, zoals een brute-force aanval, een hoog meldingsniveau krijgt toegewezen. De meldingsniveaus gaan van 0 tot 15 en de gebruiker kan hier op filteren. Aan de hand van deze niveaus kan een organisatie zelf bepalen welke alerts ze belangrijk vinden om bij te houden en welke niet.

Wazuh werkt onderliggend met drie componenten: de Wazuh Indexer, het Wazuh Dashboard en de Wazuh Server. Deze componenten kunnen samen worden geïnstalleerd op één VM, of apart op verschillende VM's. Het voordeel van de componenten samen te houden is dat dit het installatieproces vergemakkelijkt. Wazuh voorziet een script in hun documentatie waarmee je snel een werkend Wazuh-platform kunt opzetten. Het nadeel hiervan is dat deze oplossing minder schaalbaar is. Voor een klein SOC is dit geen probleem, maar voor grote organisaties is het wel belangrijk om deze flexibiliteit te hebben. Daarom kunnen zij ervoor verkiezen om de componenten op aparte VM's te installeren. Op deze manier kunnen ze op lange termijn makkelijker rekening houden met een groeiend of krimpend aantal endpoints.

De Wazuh Indexer is verantwoordelijk voor het opslaan, doorzoekbaar maken en analyseren van alerts die gegenereerd zijn door de Wazuh Server. Het doet dit door de alerts te ontvangen van de Wazuh Server en deze te indexeren, zodat ze later makkelijk doorzoekbaar zijn. Zo kun je als analist snel alerts filteren om bijvoorbeeld alle alerts van een specifieke endpoint te krijgen van de afgelopen 24 uur. De alerts kunnen ook op het Dashboard zichtbaar gemaakt worden met verschillende soorten grafieken.
De Wazuh Indexer gebruikt achterliggend Opensearch, een open-source fork van het tegenwoordig betalende Elasticsearch. Elasticsearch is een zoek- en analysetool die in staat is om snel gestructureerde of ongestructureerde data te doorzoeken. Het staat Wazuh toe om hun alerts te filteren en doorzoeken op basis van parameters zoals meldingsniveau of endpoint. Opensearch wordt door Wazuh gekozen zodat Wazuh alles zelf in handen heeft. Ze kiezen zelf welke richting ze uit willen gaan en aangezien het open-source is, zijn er geen extra kosten. Wazuh kan nu ook hun Opensearch updates aanpassen aan die van de Wazuh Indexer om een soepele integratie tussen de twee te behouden.

De Wazuh Server is verantwoordelijk voor het beheren, configureren en updaten van de agents. Ze analyseren alle data die binnenkomt vanuit de agents. Aan de hand van decoders, regels en threat intel zoeken ze naar Indicators of Compromise (IoC's). Deze krijgen dan een label en worden opgeslagen in de Wazuh Indexer voor verdere analyse. De verschillende alerts worden zichtbaar gemaakt op het Dashboard. Via de Wazuh manager is het ook mogelijk om verschillende aanpassingen aan de configuratie te maken. Zo kun je instellen vanaf welke meldingsniveau je alerts wil beginnen genereren, op maat gemaakte filters maken voor de alerts, connecties maken met andere tools zoals Shuffle en TheHive, … Op deze manier voorziet Wazuh flexibiliteit voor zijn gebruikers en kan iedereen zijn Wazuh-platform instellen naar hun noden.

Het Wazuh Dashboard is de webinterface die de alerts visualiseert voor de gebruiker. Dit gebeurt voornamelijk aan de hand van grafieken. Het Dashboard is ook een plaats om aanpassingen te maken aan de configuratie. Dit kan door middel van verschillende tools op het Dashboard zoals DevTools. Hiermee kunnen er API-calls gedaan worden naar de Wazuh Indexer en kan data opgehaald, verwijderd of aangepast worden.

MOGELIJKHEDEN MET WAZUH

Wazuh biedt tal van extra mogelijkheden bovenop het maken en opslaan van alerts. Hieronder worden een aantal mogelijkheden gegeven, die ook mogelijk zijn in het kader van deze stageopdracht.

Security Configuration Assessement (SCA)

Wazuh monitort de systeem- en applicatieconfiguratie om te controleren of deze in orde zijn. De controle hiervan gebeurt met behulp van CIS-benchmarks of andere hardeninggidsen. Deze tools controleren een endpoint om te kijken hoe veilig deze is. Dit wordt weergegeven met een score. Des te hoger de score des te beter. Wazuh agents voeren periodieke scans uit waarin ze opzoek gaan naar gaten in de beveiliging of slechte configuraties waar misbruik van gemaakt kan worden en passen de score aan. De resultaten van de scans worden getoond op het Dashboard.

File Integrity Monitoring (FIM)

Wazuh monitort het filesysteem van de agent op alle veranderingen die gebeuren. Dit kan gaan van een aanpassingen in de inhoud van een bestand tot het veranderen van toegangsrechten op een map. Wazuh kan op deze manier schadelijke files detecteren en gecompromitteerde endpoints identificeren.

Threat hunting

Wazuh onderzoekt bepaald gedrag op endpoints dat kan duiden op een IoC. Het doet dit met behulp van het MITRE ATT&CK framework om veelvoorkomende Tactics, Techniques and Procedures (TTP's) op te sporen. TTP's beschijven hoe aanvallers hun aanvallen plannen en uitvoeren. Er wordt gezocht naar acties die overeenkomen met deze TTP's. Met behulp van threat hunting worden risico's opgespoord die voorbij de eerste securitylaag geraakt zijn.

Vulnerability detection

Wazuh controleert software op endpoints voor mogelijke CVE's die zich hierop bevinden. CVE's worden bijgehouden in verschillende databases. Op deze manier kunnen mensen, die dezelfde kwetsbaarheden op hun systemen hebben, deze databases raadplegen en een oplossing vinden. Wazuh deelt de CVE's in, op basis van de score die ze krijgen. Deze indeling is Low, Medium, High en Critical.

Incident response

Wazuh biedt de mogelijkheid om meteen tegen de alerts en threats in te gaan met hun Incident Response. De Incident Response schiet in actie als er aan bepaalde eisen voldaan worden. Een voorbeeld van een Response-techniek is om het endpoint volledig af te sluiten van het netwerk. Op deze manier is er geen dreiging voor andere systemen in hetzelfde netwerk en kan de threat ook niet meer verbinden met ons systeem.

Regulatory compliance

Tegenwoordig zijn er veel verplichtingen vanuit de wetgeving waar organisaties zich aan moeten houden. In de EU is er de GDPR-wetgeving (General Data Protection Regulation), een privacywetgeving die organisaties zegt hoe ze met gegevens moeten omgaan. Wazuh helpt bedrijven door te controleren of ze aan deze wetgeving voldoen. Wazuh biedt ook ondersteuning voor andere wetgevingen, die voor deze stageopdracht niet van belang zijn.

IT Hygiene

In IT is een geoptimaliseerd systeem nodig om alles soepel te laten werken en alles veilig te houden. Onnodige open poorten kunnen leiden tot achterdeuren voor hackers en dit moet vermeden worden. Wazuh biedt daarom de mogelijkheid om van alle endpoints de lopende applicaties, open poorten en informatie over het OS-systeem en hardware bij te houden. Zo kan er op een centrale plaats gekeken worden naar de aparte endpoints en kunnen er op een visuele manier verbeteringen gespot worden.

WAAROM WAZUH?

Voor mijn stageopdracht maakte ik gebruik van Wazuh als SIEM-platform. Vanuit mijn stageplaats kwam al de interesse om deze tool te gebruiken en dit stond ook in de stageopdracht vermeld. Ze hadden dit zelf al eens uitgetest en wilden graag de verdere mogelijkheden bekijken die Wazuh te bieden heeft.

De Dashboard functie kwam ook zeker van pas. In mijn stageopdracht stond vermeld dat ze graag een centraal dashboard wilden. Het Wazuh Dashboard is heel duidelijk en voldoet aan al hun eisen.

Een andere drijfveer is dat Wazuh een open-source platform is. Het MPI Oosterlo VZW is een vereniging zonder winstoogmerk, waardoor er gestreefd wordt om geen onnodige kosten te maken. Er werd ook duidelijk vermeld aan het begin van mijn stage dat de voorkeur altijd uitgaat naar open-source oplossingen die lokaal gehost kunnen worden. Wazuh ligt helemaal in lijn met wat ze op dit vlak in gedachten hebben en past daarom perfect in het SOC.

Toen ik de opdracht las om een SOC te maken voor het MPI Oosterlo, ging mijn voorkeur er al naar uit om Wazuh als SIEM te gaan gebruiken. Het is een tool die tijdens de lessen gezien is en die ik al eens eerder had gebruikt. Ik voelde me er dus best comfortabel bij om dit op te zetten in hun omgeving, zonder me zorgen te moeten maken dat er iets stuk kan gaan. Aangezien het SIEM het startpunt is van mijn SOC, leek het me best om met een vertrouwde tool te beginnen zodat ik eerst wat zelfvertrouwen kon opbouwen binnen de organisatie, alvorens nieuwe tools te gaan gebruiken.

Tenslotte ging mijn voorkeur ook uit naar Wazuh omdat het gewoon een heel goed platform is. De online documentatie op hun site is heel duidelijk en er is veel informatie te vinden als ik ergens vast zou lopen. Wazuh voorziet een ganse pagina voor enkel troubleshooting, die ik kon raadplegen zouden er ergens problemen zijn. Ook de online community is groot en op Reddit of Discord kun je altijd bij hen terecht voor vragen of staan er links naar YouTube video's waar het wordt uitgelegd.

WRM

In onderstaande WRM wordt Wazuh vergeleken met twee andere SIEM-oplossingen: Security Onion en Graylog. Een WRM is een waarde-risico matrix waarin verschillende opties vergeleken worden aan de hand van een puntensysteem. De optie met de hoogste score is meestal de beste keuze.

Criteria	Wazuh	Security Onion	Graylog
Kosten	5	5	5
Flexibiliteit	5	3	4
Integratiemogelijkheden	5	3	4
Alert detectie	5	4	3
Gebruiksvriendelijkheid	4	3	4
Schaalbaarheid	4	3	4
Eigen kennis	5	1	2
Keuze stageplaats	5	1	1
Totaal	38	23	27

Tabel 1: WRM voor Wazuh, Security Onion en Graylog

2.2. SOAR

SOAR staat voor Security Orchestration, Automation and Response en zorgt ervoor dat organisaties incidenten kunnen verzamelen en hierop automatisch kunnen reageren. Het speelt een belangrijke rol binnen het SOC om het incidentenbeheer vlot te doen verlopen en de efficiëntie te verhogen.

Een SOAR bestaat meestal uit verschillende platformen die samenwerken om een automatische werkstroom te creëren. Deze platformen komen samen in één omgeving waar de juiste configuraties gedaan worden om acties te coördineren, automatiseren en uit te voeren. Hierdoor wordt de efficiëntie verhoogd en handmatige taken verminderd. Het helpt organisaties om steeds complexere en snellere dreigingen tegen te gaan.

In dit deel van de analyse worden de verschillende SOAR-platformen besproken die gebruikt worden in het SOC. Van elk platform wordt er besproken wat het doet, de functionaliteiten die het biedt en waarom het gekozen is. Elk platform vervult ook zijn rol in de term SOAR. Shuffle staat in voor de orchestration en automation, Cortex zorgt voor automation en response en Teams staat ook in voor response. TheHive kan niet echt een categorie krijgen, maar vormt een essentieel onderdeel van de automation en als interface voor de gebruiker.

2.2.1. Shuffle

Shuffle is een open-source SOAR-platform dat ontworpen is om een groot probleem in de cybersecuritywereld aan te pakken: er moet te veel handmatig gebeuren. In de wereld van cybersecurity draait alles om snel en op een gestructureerde manier te reageren op aanvallen en bedreigingen. Sommige organisaties proberen dit handmatig te doen door eerst zelf op zoek te gaan naar de bedreiging, hierna de bedreiging zelf te onderzoeken en ten slotte zelf op zoek te gaan naar een oplossing. Deze tactiek zorgt voor veel saai en repetitief werk dat de aandacht van de belangrijke incidenten weghaalt. Het is voor grote organisaties ook niet haalbaar om deze werkwijze toe te passen. Daarom proberen zij één grote tool te maken die alles doet, van het opsporen van de bedreiging, tot het onderzoeken ervan, een case maken en de bedreiging op te lossen. Dit zijn zware platformen die veel tijd kosten om te maken en die bedrijven graag voor zichzelf houden. Shuffle probeert hiervan af te stappen door een platform aan te bieden waar dit allemaal gedaan kan worden, zonder zelf een zwaar platform te moeten maken. Hun missie is om met hun platform processen en workflows onderling te delen, een gestandaardiseerde aanpak van detecties en automatiseringen te creëren en samenwerkingen tussen organisaties te bevorderen. Zoals de eigenaar ook zegt: “*Cybersecurity is not a competition, and shouldn’t be treated as such.*”.

Shuffle integreert verschillende andere tools en platformen door gebruik te maken van hun API’s. De meeste platformen voorzien deze API’s zodat organisaties tijdrovende taken kunnen automatiseren. Shuffle zet nog een stap verder door een gebruiksvriendelijke web interface te maken, waaruit deze API-calls gemaakt kunnen worden naar de platformen. Hoewel de gebruiker de platformen zelf nog moet opzetten, bespaart dit veel tijd omdat hij niet zelf hoeft uit te zoeken hoe de API werkt of de calls moet opstellen en automatiseren. Shuffle voorziet een integratie met veel verschillende platformen en tools, waardoor het voor iedereen beschikbaar is.

Zoals net al vermeld werkt Shuffle met een web-interface, die het gebruik van Shuffle makkelijker maakt. Hierachter ligt een krachtige API die gebruikt wordt om de andere API’s aan te spreken. Voor gevorderde gebruikers is het ook mogelijk om rechtstreeks met deze onderliggende Shuffle API te werken, bijvoorbeeld om eigen integraties te bouwen of geautomatiseerde acties aan te zetten buiten de webinterface. Ongeacht welke optie gekozen wordt, staat hier security natuurlijk ook centraal. Shuffle maakt gebruik van Bearer Auth om de API-calls te beveiligen. Dit houdt in dat er bij elke request een API-key wordt meegestuurd die aangeeft dat jij de request naar de API maakt. Zo moet er niet constant een gebruikersnaam en wachtwoord meegestuurd worden. De token zorgt dus voor goede security van de API, maar behoudt toch zijn flexibiliteit om gebruikt te worden in scripts of andere automatiseringstools.

Om Shuffle te gebruiken zijn er enkele mogelijkheden beschikbaar. Je kunt alles zelf lokaal hosten, gebruik maken van de cloud versie of een mix van de twee en kiezen voor een hybride oplossing. Elk heeft zijn voordelen zodat er per organisatie gekeken kan worden wat het beste voor hun past. Zo heb je bij een lokale hosting meer controle over de infrastructuur, maar moet je dan ook alles zelf up-to-date houden. Dit is een goede oplossing voor kleinere organisaties of VZW's.
In de cloud geef je dit dan weer uit handen zodat je minder onderhoud hebt en makkelijker kunt schalen, maar hier kunnen wel kosten aan verbonden zijn. Er zijn limieten gezet op de gratis versie van de cloud, zoals maximaal 10 workflows in totaal of 2000 App-runs per maand. Je kunt deze limieten vergroten door een abonnement te nemen. Deze abonnementen kunnen vaste limieten hebben of op basis van je organisatie gemaakt worden.
Een hybride oplossing klinkt dan als het beste van twee werelden, maar de set-up is complexer en je hebt veel kennis nodig van zowel lokaal als cloud-hosting om dit goed te laten werken.
Kortom, er is voor iedere organisatie wel een oplossing en dit is een extra laag flexibiliteit die Shuffle aanbiedt.

MOGELIJKHEDEN MET SHUFFLE

Om Shuffle zijn automatisatie en flexibiliteit te geven, zijn er verschillende features toegevoegd om dit mogelijk te maken voor de eindgebruiker. Hieronder kunt u de meest gebruikte features vinden, die in het SOC gebruikt kunnen worden.

Workflows

Workflows zijn de kern van Shuffle en zorgen ervoor dat de dagelijkse taken geautomatiseerd worden. Ze bestaan uit apps, triggers, voorwaarden en variabelen om krachtige en gebruiksvriendelijke automatiseringen op te zetten.
Shuffle staat toe dat je meerdere workflows kunt maken die elk hun eigen doeleinde hebben. Zo kun je een workflow maken die zich puur richt op het maken van cases in TheHive vanuit Wazuh alerts en een andere workflow die zich focust op het analyseren van mails op phishing. Verschillende workflows kunnen op hetzelfde moment lopen, waardoor er meer taken gedaan kunnen worden op korte tijd. Ze kunnen ook automatisch gestart worden op basis van een webhook of een geplande taak.
Shuffle biedt verschillende templates aan die gemaakt zijn door andere gebruikers. Zo hoef je niet alle workflows zelf te maken en kun je inspiratie opdoen van wat anderen gemaakt hebben.

Apps (openAPI of zelfgemaakt in python)

Apps zijn de bouwstenen waaruit workflows opgebouwd worden. Het is een integratie met een externe tool, zoals Wazuh, TheHive, Outlook, VirusTotal … die gebruikt kunnen worden om verschillende acties uit te voeren op de tool. Deze acties kunnen bestaan uit het ophalen van alerts uit Wazuh, het aanmaken van een case in TheHive, een analyse starten met VirusTotal, … De acties worden vooraf gedefinieerd door de maker van de app en komen overeen met de functies die de externe tool aanbiedt.
Om ervoor te zorgen dat de actie succesvol uitgevoerd wordt, zijn er argumenten nodig. Deze argumenten kunnen vergeleken worden met variabelen, waarin je bepaalde gegevens meegeeft zoals een IP-adres dat onderzocht moet worden, of een alert waarvan we een case willen maken. Elke actie heeft enkele verplichte en optionele argumenten die bepaald worden door de maker van de app.
Apps worden dus vooral gebruikt om automatische acties uit te voeren op de externe tools. Standaard voorziet Shuffle meer dan 2500 apps, maar je kunt er ook zelf maken in Python.

Triggers

Automatisatie zou geen automatisatie zijn als er nog steeds zaken handmatig moeten gebeuren, zoals bijvoorbeeld het starten van een workflow. Daarom voorziet Shuffle verschillende triggers die een workflow kunnen starten. Er zijn in totaal vier soorten triggers die veel voorkomen: webhooks, schedules, subflows en user input.
Een webhook luistert naar inkomende data van andere tools, zoals een alert op Wazuh. Zodra die tool iets nieuws wil melden, schiet de rest van de workflow in gang om met de nieuwe data aan de slag te gaan. Zo kan er bijvoorbeeld een case gemaakt worden in TheHive van de Wazuh alert.
Een schedule is een taak die op een bepaald tijdstip de workflow aan zet. Dit kan gebruikt worden om elke dag nieuwe IOC's op te halen uit MISP en hier een lijst van op te sturen via mail.
Een subflow is een trigger van een workflow naar een andere workflow. Dit kan gebruikt worden bij het analyseren van een IP-adres. Je kunt telkens verschillende acties toevoegen die het IP-adres controleert, maar je kunt ook een workflow maken die deze acties ook doet en waarnaar verwezen wordt vanuit een andere workflow.
Ten slotte is er nog user input. Dit is wanneer een gebruiker een bepaalde handeling uitvoert, zoals het downloaden van een bestand, dat een workflow aanzet om te controleren of het bestand wel veilig is.

Marketplace

Om alle bovenstaande zaken makkelijk te kunnen vinden en niet zelf alles handmatig te moeten maken, voorziet Shuffle een marketplace. Op deze marketplace kun je van alles vinden zoals nieuwe apps en bestaande workflows. Shuffle vindt het belangrijk om dit soort informatie met elkaar te delen en daarom is een plaats zoals de marketplace net extra belangrijk om elkaar te helpen met het beveiligen van je organisatie.

WAAROM SHUFFLE?

Voor mijn stageopdracht maakte ik gebruik van Shuffle omwille van de automatisering die het biedt voor mijn SOC. Eén van de grote voordelen van een SOC is dat er veel zaken automatisch op een centrale plaats gebeuren. Dit bespaart veel tijd die anders verloren gaat aan repetitieve taken. Omdat het MPI Oosterlo VZW een kleine IT-dienst heeft, is deze automatisatie van cruciaal belang. Ze zijn namelijk niet enkel bezig met het onderzoeken van mogelijke bedreigingen, ze moeten ook het personeel ondersteunen als zij technische problemen ondervinden. Het is dus belangrijk dat als ze gebruik willen maken van mijn SOC, dat alles voor hen klaar staat om op mogelijke bedreigingen te reageren.

Shuffle is, net als Wazuh, een open-source tool die volledig lokaal gehost kan worden. De voorkeur ging al uit naar dit soort tools, zodat ze geen overbodige kosten maken en zelf veel controle over de tool hebben.

Om centraal beheer van mijn SOC te behouden, is Shuffle een centrale plaats om alle API-verbindingen te zien. Hoewel deze integraties ook in de andere tools zelf zit, zoals tussen Wazuh en TheHive, is het moeilijk om een overzicht hierover te houden. Met Shuffle zijn alle API-integraties en taken, die automatisch uitgevoerd worden, zichtbaar op één plaats. Dit zorgt voor een centrale plaats waar we makkelijk aan troubleshooting kunnen doen, mochten er fouten optreden.

Tenslotte is Shuffle een tool waarmee ik al eens gewerkt heb. Dit zorgt ervoor dat ik al een basis heb om mee van start te gaan.

WRM

Op onderstaande WRM wordt Shuffle vergeleken met twee andere SOAR-platformen: Crowdstrike Falcon en Splunk SOAR.

Criteria	Shuffle	Crowdstrike Falcon	Splunk SOAR
Kosten	5	2	1
Flexibiliteit	5	3	4
Integratiemogelijkheden	4	4	5
Automatiseringsmogelijkheden	5	4	5
Gebruiksvriendelijkheid	4	4	5
Schaalbaarheid	4	4	5
Eigen kennis	5	1	1
Totaal	32	22	31

Tabel 2: WRM voor Shuffle, Crowdstrike Falcon en Splunk SOAR

2.2.2. TheHive

TheHive is een open-sourceplatform voor security incident en response dat essentieel is voor beveiligingsteams zoals SOC-teams. Het stelt hen in staat om incidenten snel en doeltreffend af te handelen. Het biedt een platform aan waarop teams makkelijk kunnen samenwerken aan verschillende incidenten en hier analyse op kunnen uitvoeren. Door de vlotte integratie met andere beveiligingstools en de schaalbaarheid en flexibiliteit die het platform biedt, is het inzetbaar in zowel grote als kleine omgevingen. Dankzij de gebruiksvriendelijke interface worden de incidenten op een overzichtelijke manier bijgehouden en wordt analyses uitvoeren makkelijker.

TheHive werkt met organisaties om de incidenten naar op te sturen. Deze organisaties worden aangemaakt in het admin-portaal, waar het account beschikbaar voor wordt gemaakt als TheHive opgezet is. De organisaties kunnen voor verschillende doeleinden gebruikt worden. Bij kleinere omgevingen kan er slechts één organisatie opgezet worden waar alle incidenten bijgehouden worden. Bij grotere omgevingen is het voordeliger om verschillende organisaties op te zetten voor de verschillende analyseteams. Zo kunnen incidenten overzichtelijk blijven en moeten teams niet zoeken naar incidenten waar zij verantwoordelijk voor zijn.

Incidenten worden geplaatst in de verschillende organisaties. Dit kan handmatig gebeuren in TheHive zelf of automatisch door gebruik te maken van de API. De incidenten worden bewaard onder het tabblad “Alerts” waarin de uitleg over het incident vermeld staat.
In deze alerts kan belangrijke informatie, zoals IP-adressen, domeinnamen en bestandsnamen, opgeslagen worden in observables. Met deze observables kan er een case gemaakt worden waarin alle informatie gebundeld wordt. Door de integratie van TheHive met beveiligingstools zoals VirusTotal, kan er een automatische analyse gestart worden van het incident. Dit bespaart de gebruikers binnen de organisatie veel tijd om handmatig analyses uit te voeren en kunnen ze hun focus leggen op de response.

Om analyse uit te voeren in de organisaties, zijn er gebruikers nodig die toegang hebben tot de webinterface. Vanuit het admin-portaal is er de mogelijkheid om gebruikers toe te voegen aan een organisatie, zolang er hiervoor een e-mailadres wordt meegegeven. Op deze manier kan elke organisatie zijn eigen gebruikers beheren en kunnen zij enkel hun cases zien. In TheHive gebruiken ze Role-Based Access Control (RBAC) om de juiste rollen toe te kennen aan gebruikers. Deze rollen zijn:

- Admin:
    - Volledig administratieve rechten
    - Enkel op admin-portaal
    - Kan geen cases en alerts bekijken
- Org-admin:
    - Beheert gebruikers en configuraties binnen de organisatie
    - Heeft een API-key voor integratie met andere tools
    - Kan cases en alerts aanmaken en bekijken
    - Kan analyzers en responders starten
- Analist
    - Kan cases en alerts aanmaken en bekijken
    - Kan analyzers en responders starten
- Read-only
    - Kan cases en alerts bekijken

Organisaties kunnen hiermee bepalen hoe ze hun teams willen opstellen in TheHive en makkelijk de juiste rechten toekennen.

TheHive werkt met een licentiesysteem, ongeacht of je de community versie of een betalende versie gebruikt. Een organisatie moet zich registreren om in aanmerking te komen voor een licentie. Met deze licentie kun je één van de drie versies van TheHive aanvragen: community, gold of platinum. Elk hebben hun eigen voordelen, maar de betalende versies kunnen oplopen tot €25.000 voor on premise of zelfs €50.000 voor in de cloud.

MOGELIJKHEDEN MET THEHIVE

TheHive biedt tal van mogelijkheden om efficiënt te werken met de cases en alerts. Hieronder worden de belangrijkste, die in mijn SOC gebruikt kunnen worden, uitgelegd.

Samenwerken in real-time

Eén van de belangrijkste eigenschappen van TheHive, is de mogelijkheid om tegelijkertijd met meerdere gebruikers aan cases, taken en observables te kunnen werken. Dit biedt de organisatie een platform waar nauw opgevolgd kan worden wie wat aan het doen is en welke taken er nog moeten gebeuren. Dit bevordert de samenwerking tussen collega's.

Aanpasbare sjablonen

Om organisaties zoveel mogelijk vrijheid te geven, kunnen gebruikers hun eigen sjablonen aanmaken en bestaande sjablonen aanpassen. Deze sjablonen kunnen gebruikt worden voor het maken van cases of alerts, maar ook voor verschillende dashboards die beschikbaar zijn in TheHive. Hierdoor kan het platform afgestemd worden op de specifieke noden van de organisatie.

Informatiebeheer

Bij het onderzoeken van cases komt er veel informatie naar boven. Deze informatie kan nuttig zijn om andere cases op te kunnen lossen. Daarom voorziet TheHive een manier om informatie onderling met elkaar te delen. Dit kan door documenten aan te maken in TheHive met hierin de informatie over een IP-adres of domein. Andere analisten kunnen deze documenten dan gebruiken om hun cases op te lossen. Deze informatie kan ook toegevoegd worden aan de cases zelf. Zo kan er later opnieuw naar gekeken worden en is het duidelijk hoe de case opgelost is.

Beheer van observables

Observables zijn stukken informatie die toegevoegd worden aan een case. Ze geven meer informatie over de case en zijn een cruciaal onderdeel van de analyse. Het kan bijvoorbeeld gaan over het IP-adres waaruit een brute-force aanval is opgezet. Dit IP-adres wordt dan toegevoegd aan de case en kan verder onderzocht worden. Cases in TheHive zijn ook niet gelimiteerd aan slechts één observable, er kunnen er meerdere toegevoegd worden als deze volgens de analisten belangrijk zijn. Door de observables op een centraal punt te hebben, kunnen patronen makkelijk ontdekt worden en worden cruciale zaken minder snel over het hoofd gezien.

Threat intelligence integration

Om de observables te onderzoeken, wordt er gebruik gemaakt van threat intelligence. Dit zijn platformen waarop gebruikers een observable kunnen ingeven om er informatie over te krijgen. Zo kan er bepaald worden of een observable kwaadaardig is of niet.
Om de analyse hiervan vlotter te laten verlopen, heeft TheHive een integratie met verschillende threat intelligence platformen. Met MISP wordt er vanuit TheHive zelfs de mogelijkheid aangeboden om een connectie op te zetten met een API-sleutel. De overige platformen worden allemaal verbonden met Cortex, dat in het volgende deel aan bod komt. Door deze verbinding met Cortex zijn de threat intelligence-platformen zichtbaar en bruikbaar in TheHive. De rapporten die hierdoor gemaakt worden, komen ook bij in de case te staan. Het biedt een overzichtelijke manier om informatie van observables in cases te bekijken, zonder alle platformen één voor één af te moeten gaan.

WAAROM THEHIVE?

TheHive is één van de beste open-sourceplatformen dat gebruikt wordt voor security incident en response. Het zorgt voor een vlot alert- en casebeheer, doordat alle meldingen op één centrale plaats komen. Hier kunnen de incidenten gecategoriseerd en prioriteiten ingesteld worden. Zo is er een overzichtelijke beheerplek waarop de werknemers van het MPI Oosterlo VZW kunnen werken. De mogelijkheden die TheHive biedt om samenwerking tussen collega's te ondersteunen, maken het een geschikte keuze voor het SOC.

Een belangrijke drijfveer voor het gebruik van TheHive is de automatisering, die mogelijk is dankzij de integratie met verschillende tools. In een SOC wordt gewerkt met diverse beveiligingstools, maar het integreren ervan verloopt niet altijd even vlot. TheHive biedt standaardintegraties met onder andere Wazuh en Shuffle, twee tools die ook in mijn SOC gebruikt worden, wat het opzetten van workflows vereenvoudigt. Daarnaast maakt de integratie met Cortex, waar ik zo meteen dieper op inga, het uitvoeren van analyses een stuk efficiënter.

Ten slotte is TheHive een platform waar ik al eens eerder mee gewerkt heb. In het SOC dat ik voor school maakte, heb ik ook gebruik gemaakt van TheHive. Hierdoor heb ik al wat ervaring met het platform en gaat dit me helpen tijdens de realisatie.

WRM

Op onderstaande WRM wordt TheHive vergeleken met twee andere casebeheerplatformen: DFIR IRIS en FortiAnalyzer.

Criteria	TheHive	DFIR IRIS	FortiAnalyzer
Kosten	5	5	2
Flexibiliteit	4	4	3
Integratiemogelijkheden	4	3	4
Automatiseringsmogelijkheden	3	3	4
Gebruiksvriendelijkheid	4	4	4
Schaalbaarheid	3	3	4
Eigen kennis	5	1	1
Totaal	28	23	22

Tabel 3: WRM voor TheHive, DFIR IRIS en FortiAnalyzer

2.2.3. Cortex

Cortex is een open-source analyse- en responseplatform dat ontworpen is om beveiligingsteams te helpen met het analyseren van observables. Het zorgt ervoor dat de verschillende teams snel analyses kunnen uitvoeren op observables in TheHive. Cortex is, net als TheHive, een onderdeel van StrangeBee. Hierdoor zijn er veel mogelijkheden van TheHive ook beschikbaar in Cortex. Zo kunnen teams en collega's vlot samenwerken en analyses bekijken. Door een vlotte integratie met verschillende threat intelligence-tools, biedt het de gebruikers een centrale plaats om analyses uit te voeren. De schaalbaarheid en flexibiliteit die het platform bezit, maakt het in zowel grote als kleine omgevingen een nuttige toevoeging. Door de gebruiksvriendelijke interface, worden de analyses en responses op een overzichtelijke manier bijgehouden en getoond aan de gebruiker.

Net als TheHive werkt Cortex met organisaties, waarin de analyses en responses plaatsvinden. Deze organisaties worden opgezet vanuit het admin-portaal. Het account hiervoor wordt beschikbaar gemaakt van het moment dat Cortex opgezet is. Omdat dit in het deel over TheHive reeds is uitgelegd, wordt het hier niet herhaald. Een groot verschil met TheHive is dat er geen nood is aan een licentie voor Cortex. Het is dus vanaf de installatie volledig bruikbaar zonder limieten.

In Cortex wordt er niet met incidenten gewerkt, maar met analyzers en responders. Deze zijn zichtbaar op de webinterface en kunnen hier door organisaties beheert en geraadpleegd worden. De analyses kunnen handmatig uitgevoerd worden in Cortex door een observable op te geven en de analyzers aan te duiden die de analyse moeten uitvoeren. Hierna komt er een rapport vrij in het tabblad 'Jobs history', dat de gebruiker kan raadplegen voor de informatie over de observable. Hoe dit rapport eruitziet, is van analyzer tot analyzer verschillend.
Een analyse kan ook gestart worden via de API. Cortex voorziet een API-sleutel waarmee er een verbinding opgezet kan worden met andere tools, zoals TheHive of Shuffle. Bij het gebruik van de API komt de informatie niet altijd op de webinterface van Cortex. Bijvoorbeeld in TheHive zijn ze dan enkel zichtbaar in de case.

Een responder kan niet alleen via de API gestart worden, maar ook via de webinterface van TheHive, op voorwaarde dat alles juist is ingesteld. Wanneer een case, alert of observable geopend wordt in TheHive, is het mogelijk om daaruit een responder manueel te starten. Hiervoor moet de responder wel correct geconfigureerd zijn in Cortex én gedeeld worden met TheHive. In de praktijk worden responders echter vooral automatisch opgestart, bijvoorbeeld via een workflow in Shuffle, omdat ze bedoeld zijn om het incident response-proces te automatiseren en versnellen.

MOGELIJKHEDEN MET CORTEX

Cortex heeft twee belangrijke functionaliteiten: analyzers en responders. Beide zijn bedoeld om het SOC zoveel mogelijk te ondersteunen met automatisatie. Hoewel het gebruik van responders door de stageplaats als out-of-scope werd beschouwd, wordt hieronder toch kort toegelicht wat beide onderdelen doen, om zo een volledig beeld van Cortex te schetsen.

Analyzers
Analyzers worden voornamelijk gebruikt voor het analyseren van observables. Meestal gaat het hierbij om threat intelligence-platformen die informatie ophalen over bijvoorbeeld IP-adressen, hashes of domeinen. Denk aan platformen zoals VirusTotal of MISP. Cortex maakt het mogelijk om deze verschillende bronnen centraal aan te spreken, waardoor er veel minder tijd verloren gaat aan het apart opvragen van informatie bij elk platform. De analyzers maken op verschillende manieren verbinding met Cortex, maar de meest gebruikte methode blijft het gebruik van een API-sleutel.

Responders

Na het uitvoeren van een analyse kan er behoefte zijn aan een onmiddellijke reactie. Hiervoor biedt Cortex verschillende responders aan. Deze zijn gekoppeld aan tools die automatisch bepaalde acties kunnen uitvoeren, zoals het blokkeren van een IP-adres op de firewall of het in quarantaine plaatsen van een verdacht endpoint. Door dergelijke stappen snel te zetten, kunnen systemen tijdelijk beveiligd worden tot er een grondigere analyse kan plaatsvinden. Op basis daarvan kunnen vervolgens de juiste beslissingen genomen worden.

WAAROM CORTEX?

Ik heb Cortex in mijn SOC gebruikt omwille van de automatische analyses die ermee uitgevoerd kunnen worden. Dit zorgt ervoor dat een cruciaal deel van het SOC geautomatiseerd is. Het helpt de collega's van de helpdesk, zodat ze niet tussen andere problemen door constant analyses moeten uitvoeren. Ze kunnen nu kijken naar de analyses die Cortex heeft gedaan en gebaseerd op deze informatie de volgende stappen bespreken en uitvoeren.

De keuze van Cortex komt ook doordat het een onderdeel is van StrangeBee. Dit zorgt ervoor dat er goede compatibiliteit is tussen TheHive en Cortex. Ze zijn altijd op elkaar afgestemd en dit maakt de integratie van de tools ook eenvoudiger.

Cortex is een tool waar ik nog niet mee gewerkt heb, in tegenstelling tot de andere tools in het SOC. Het heeft een uitgebreide documentatie en online is er ook veel informatie over te vinden. De meeste tutorials die TheHive gebruiken, maken ook gebruik van Cortex omdat ze in hetzelfde ecosysteem zitten. Hierdoor ben ik zeker dat de integratie in mijn SOC goed zal verlopen.

Tenslotte is Cortex net als TheHive open-source. Dit houdt rekening met de kosten van het SOC zo laag mogelijk te houden. StrangeBee belooft ook dat het beide tools open-source zal houden, dus daardoor is het ook een goede oplossing voor op lange termijn.

WRM

Op onderstaande WRM wordt Cortex vergeleken met twee andere analyse- en responseplatformen: DFIR IRIS en Port.

Criteria	Cortex	DFIR IRIS	Port
Kosten	5	5	5
Flexibiliteit	4	4	3
Integratiemogelijkheden	5	4	4
Automatiseringsmogelijkheden	5	4	4
Gebruiksvriendelijkheid	4	4	3
Schaalbaarheid	4	3	4
Eigen kennis	5	1	1
Totaal	32	25	25

Tabel 4: WRM voor Cortex, DFIR IRIS en Port

2.2.4. Teams

Om de uiteindelijke melding te maken die de IT-dienst van het MPI Oosterlo op de hoogte moet stellen van een alert, wordt er gebruik gemaakt van Teams. Tegenwoordig is Teams in haast geen enkele organisatie meer weg te denken. Het is één van de grootste communicatieplatformen dat ontwikkeld is door Microsoft. Hierdoor zit het bij veel bedrijven bij in hun Microsoft 365-pakket.

Doordat het op zoveel verschillende plaatsen gebruikt wordt, waarbij vaak ook in gevoelige domeinen zoals bij overheidsinstanties, is er nood aan een sterke beveiliging. Teams is één van de best beveiligde communicatieplatformen, vooral in een Microsoft 365-ecosysteem. Microsoft beveiligt de onderlinge communicatie door de data te versleutelen. Het doet dit op twee verschillende momenten. Het eerste moment is wanneer data in rust is. Dit gaat over berichten die al verstuurd zijn en opgeslagen staan in de chatgeschiedenis. Deze worden versleuteld zodat ze voor niemand anders leesbaar zijn.
Het tweede moment is als data in beweging is. Dit gaat over berichten die verstuurd zijn en onderweg zijn naar hun bestemming. Teams zorgt ervoor dat deze berichten versleuteld zijn, zodat niemand zich tussen de verzender en ontvanger kan plaatsen om berichten te onderscheppen.

Om verder beveiliging te garanderen, werkt Microsoft met Multi-Factor Authenticatie (MFA). Dit houdt in dat een gebruiker na het aanmelden, nog een tweede vorm van authenticatie moet voltooien. Deze tweede authenticatie gebeurt via de Microsoft Authenticator app. Hierop moet de gebruiker het cijfer dat op het scherm staat, ingeven in de app en bevestigen met een biometrische verificatie (vingerafdruk of gezichtsscan).

Ten slotte voldoet Microsoft aan meerdere eisen van verschillende wetgevingen en industriestandaards. Het is in overeenstemming met de GDPR, de privacywetgeving van de EU, en de ISO27001, de internationale standaard voor informatiebeveiliging. Door hieraan te voldoen, garandeert Microsoft dat hun platform in orde is voor veilige communicatie.

MOGELIJKHEDEN MET TEAMS

In Teams is het makkelijk om een chat aan te maken en hierin informatie met elkaar te delen. Om automatische meldingen te maken, voorziet Teams werkstromen die toegevoegd kunnen worden aan de chats.

Werkstromen

Werkstromen bieden gebruikers de mogelijkheid om taken te automatiseren. Hiervoor wordt er een verbinding gemaakt met één of meerdere apps. Dit kan bijvoorbeeld gaan over de status van een pipeline in Github of in dit geval een melding vanuit een SOC. In totaal biedt Teams meer dan 100 verschillende werkstroomsjablonen aan.
In deze sjablonen zit er ook de mogelijkheid om een webhook op te zetten. Deze webhook dient als een aanspreekpunt voor externe applicaties om de Teams-chat aan te spreken. Door de juiste webhook te kiezen, kunnen er meldingen in de chat geplaatst worden. Het voorziet hiervoor een sjabloon dat de structuur van de melding bepaalt. Het enige waar de gebruiker zelf nog voor moet zorgen, is dat het de webhook op de juiste manier aanspreekt. Er wordt een bepaalde structuur verwacht, die nodig is om een melding te kunnen maken. Als deze niet gehanteerd wordt, kan Teams geen melding in de chat plaatsen. Het is dus van uiterst belang dat dit in orde is. Gelukkig kan de structuur, die de werkstroom verwacht, teruggevonden worden op het Power Automate-portaal. Dit is een Microsoft-portaal waarop alle informatie van werkstromen staat en dus ook de structuur die het verwacht.

WAAROM TEAMS?

Ik heb ervoor gekozen om Teams te gebruiken, omdat dit al aanwezig was in de organisatie. Het is het interne communicatieplatform voor het MPI Oosterlo VZW en dus de ideale oplossing. Ik heb dit wel vergeleken met andere oplossingen zoals Outlook en Discord. Het probleem met Outlook was dat er geen goede integratie was met Shuffle. Dit kon enkel als er gebruik gemaakt werd van de Azure AD, wat het MPI Oosterlo niet doet. Het gebruik van Outlook was dus al snel geen optie meer.

Discord heeft wel een goede integratie met Shuffle. Ik heb hier in mijn SOC voor school gebruik van gemaakt, maar vond het voor een stageopdracht te onprofessioneel. Er zouden zich op lange termijn ook te veel problemen voordoen wat dit geen goede oplossing maakt.

Aangezien Teams bij in het Microsoft 365-pakket zit, zorgt dit ook niet voor extra kosten. Zo blijft het totale kostenplaatje van het SOC zo laag mogelijk.

Tenslotte heb ik ook met Teams al ervaring gehad om een werkstroom op te zetten. Tijdens een project voor Cloud Engineering heb ik dit gebruikt om een statusmelding van een pipeline in een chat te zetten. Hierdoor weet ik al welke stappen ik moet nemen en hoe ik een werkstroom moet opzetten. Dit gaat opnieuw het proces bevorderen.

WRM

Op onderstaande WRM wordt Teams vergeleken met twee andere communicatieplatformen: Outlook en Discord.

Criteria	Teams	Outlook	Discord
Kosten	5	5	5
Professionaliteit	5	5	2
Ordelijkheid	5	3	4
Integraties	5	3	4
Gebruiksvriendelijkheid	4	3	4
Realtime communicatie	5	2	5
Eigen kennis	4	3	5
Totaal	33	24	29

Tabel 5: WRM voor Teams, Outlook en Discord

2.3. Threat Intelligence

Om de analyses uit te kunnen voeren op de alerts in TheHive, is er nood aan Threat Intelligence. Threat Intelligence zijn platformen waarop er informatie te vinden is over observables, zoals IP-adressen of domeinnamen. Deze informatie is meestal afkomstig van andere organisaties die reeds aangevallen zijn. Op verschillende Threat Intelligence-platformen wordt de informatie over de aanval opgeslagen. De platformen zijn ontwikkeld zodat iedereen toegang kan krijgen tot deze informatie.

Er zijn ook verschillende platformen ontwikkeld voor het gebruik in een SOC. Deze werken met een API waarnaar er verzoeken gestuurd kunnen worden. In dit deel worden enkele van deze platformen besproken die gebruikt worden in het SOC.

2.3.1. VirusTotal

VirusTotal is een gratis online platform dat observables analyseert door informatie op te halen van meer dan 90 verschillende beveiligingstools, zoals antivirussen en bloklijsten. De gebruiker krijgt een score te zien die vertelt hoeveel van de beveiligingstools de observable als schadelijk beschouwen.
Naast deze score wordt er extra informatie ter beschikking gesteld zoals:

- Meer informatie over de gebruikte beveiligingstools
- De categorisatie van de dreiging, zoals malware, phishing, botnet, …
- Meer informatie over deze categorieën en wat de risico's hiervan zijn

VirusTotal helpt gebruikers bij het opsporen van malware en valse positieven om bij te dragen aan de algemene veiligheid in het IT-landschap. Centraal staan de real-time updates die ervoor zorgen dat ze altijd mee zijn met de recentste veranderingen, zoals nieuwe schadelijke IP's of domeinen.

Er zijn verschillende manieren om een analyse te starten met VirusTotal. Eerst en vooral is er de website, de meest gebruikt oplossing. Op deze website is er een zoekbalk waarin de gebruiker de observables kan ingeven of uploaden zodat ze geanalyseerd worden.
Er is ook een browserextensie die webpagina's scant voor malware of virussen en bestanden die gedownload worden scant. Het geeft dan, net zoals bij de website, een score die aanduidt hoe veilig een pagina of bestand is.
Ten slotte is er nog een API. Deze is interessant voor automatische analyses uit te voeren. Door een verzoek naar deze API te sturen, wordt er opnieuw een score gegeven zoals bij de andere opties, maar nu in JSON-formaat.

Om valse positieven te voorkomen, heeft VirusTotal de VirusTotal-community opgericht. In deze community wordt er gekeken naar de resultaten die gegeven worden voor de observables. Als er fouten worden opgespoord of nieuwe dreigingen worden ontdekt, kan de community helpen met deze aan te geven.

MOGELIJKHEDEN MET VIRUSTOTAL

Virustotal onderzoekt verschillende soorten observables. Hieronder wordt er besproken welke observables dit zijn en wat de gebruiker van informatie krijgt.

IP-adressen

Een belangrijke observable om te onderzoeken zijn IP-adressen. Online zijn er veel lijsten met bekende schadelijke IP-adressen en IP's die gebruikt worden in botnets. VirusTotal heeft een integratie met deze lijsten zodat er automatisch getoond wordt of het IP in één van deze lijsten te vinden is. Natuurlijk is dit niet het enige wat een IP gevaarlijk kan maken voor een organisatie. Daarom laat VirusTotal historische informatie zien over activiteiten gekoppeld aan het IP-adres. Deze activiteiten bevatten:

- Bestanden die gehost zijn op het IP-adres
- Domeinen die gekoppeld zijn aan het IP-adres
- Gedetecteerde kwaadaardige activiteiten, zoals Command & Control servers

Sommige tools geven een reputatiescore zoals malicious, suspicious of clean. Dit kan helpen om mensen aan te sporen verder onderzoek te doen naar het IP-adres.
Virustotal geeft ook enkele belangrijke metadata mee. Dit kan het ASN bevatten, het indentificatienummer van het netwerk, waarin het IP zich bevindt. Soms wordt ook de locatie van het IP bekend gemaakt. Dit is handig voor organisaties die willen achterhalen uit welke landen de meeste aanvallen komen, zodat ze deze landen eventueel kunnen blokkeren.

URL's

Eén van de grootste beveiligingsrisico's tegenwoordig is phishing. Vaak bevatten mails een link naar een schadelijke website. VirusTotal controleert URL's om te kijken of deze veilig zijn. Het scant de URL met tientallen website-scanners en bloklijstdiensten zoals Google Safe Browsing, Fortinet, Sophos, …
De scanresultaten bevatten informatie zoals:

- Is de site phishing-gerelateerd?
- Malware die gehost wordt op de site
- Classificatie van de site (verdacht, spam)

Het toont ook andere informatie zoals doorverwijzingen die de URL uitvoert. Hierdoor kan de URL als veilig beschouwd worden, maar gebeurt er een doorverwijzing naar een URL die schadelijk is. Door beide te controleren kan bepaald worden of de URL helemaal veilig is. De relatie die de URL heeft met andere domeinen, bestanden en IP-adressen wordt ook gecontroleerd en getoond aan de gebruiker.

File hashes

Een file hash is de unieke identificatie van een bestand. Het is een onomkeerbaar identificatienummer waardoor bestanden online herkend kunnen worden. VirusTotal controleert deze hash om te kijken of het bestand ergens gebruikt wordt voor slechte doeleinden en of er malware aan gelinkt is. Bij het opzoeken van een hash op VirusTotal krijg je informatie over het bestand zoals:

- Eerdere uploads of scans van het bestand op VirusTotal
- Een volledig scanrapport van de antivirusprogramma's en welke het bestand als schadelijk detecteren
- Extra details zoals het bestandstype, de grootte, metadata die bij in het bestand zit, …

VirusTotal linkt het bestand ook aan URL's, IP-adressen, downloadlocaties en gerelateerde samples om verdere detectie makkelijker te maken.

Domeinen

Domeinen worden gebruikt om te verwijzen naar een specifiek adres op het internet, zoals microsoft.com. Deze adressen kunnen gebruikt worden om aanvallen te starten of malware te verspreiden. Dit gebeurt vaak door een domein na te maken, zoals rnicrosoft.com, waar mensen niet nauwkeurig genoeg naar kijken. Ze proberen de website of mail zo realistisch mogelijk te maken, om gebruikers malware te laten downloaden of gevoelige data te delen. Domeinen worden ook vaak gebruikt in phishingmails waarin ze een email sturen vanuit hun nepdomein dat lijkt op een officiële mail van Microsoft.
VirusTotal verzamelt de volgende gegevens en informatie van domeinen, om ze aan hun gebruikers te laten zien:

- IP-adressen gelinkt met het domein
- Geregistreerde subdomeinen
- Malware- of phishing-hosting
- Reputatiescores

Voor verdere analyse is het ook mogelijk om de WHOIS-gegevens te krijgen van het domein. Dit is informatie zoals wie het domein heeft geregistreerd, wanneer het geregistreerd is, wanneer het verloopt, waar het geregistreerd is, … Aan de hand van deze informatie kan er dieper worden gecontroleerd of een domein onveilig is voor een organisatie of niet.
Het is ook mogelijk om de SSL-certificaten van het domein te bekijken. Dat een site HTTPS gebruikt, betekent niet altijd dat deze veilig is. Een certificaat voor HTTPS kan komen van een onofficiële site zoals Let’s Encrypt om mensen een vals gevoel van veiligheid te geven. Daarom is het belangrijk om te achterhalen waar het certificaat vandaan komt, om te bepalen of het domein veilig is.
Ten slotte is het ook mogelijk om verbonden bestanden en URL’s van het domein te bekijken en deze te onderzoeken.

WAAROM VIRUSTOTAL?

Ik heb ervoor gekozen om VirusTotal in mijn SOC te gebruiken omwille van de diverse analyses die het kan uitvoeren. Aan de hand van deze analyses kunnen verschillende observables onderzocht worden. Zo kan ik zoveel mogelijk veiligheid bieden aan het MPI Oosterlo in verschillende scenario's. De meeste observables worden automatisch geanalyseerd met de Shuffle workflow, maar ze kunnen ook handmatige controles doen. Dit kan handig zijn bij phishing-mails die van een verdacht domein komen of een verdachte bijlage bevatten. Ook de links die in de mails staan kunnen makkelijk onderzocht worden.

Virustotal is een gratis platform wat opnieuw de kosten laag houdt. Ik voldoe hiermee aan de eisen van een zo laag mogelijk kostenplaatje te hebben en zoveel mogelijk open-source tools te gebruiken.

Voor VirusTotal en alle andere Threat Intelligenceplatformen die ik hierna nog bespreek, maak ik geen WRM. Er is geen keuze gemaakt voor VirusTotal omdat het beter is dan andere platformen. Het is een platform waar ik al bekend mee ben en het daarom geïntegreerd heb in mijn SOC. Ook is het de bedoeling om op lange termijn zoveel mogelijk Threat Intelligenceplatformen toe te voegen. Hiermee kunnen valse positieven vermeden worden en zoveel mogelijk informatie verkregen worden over de observables.

2.3.2. Crowdsec

Crowdsec is een online platform en service dat gespecialiseerd is in het herkennen van kwaadaardige IP-adressen. Het doet dit door informatie op te halen uit zijn eigen gemeenschap. Crowdsec biedt namelijk een intrusion detection & prevention system (IDS/IPS) aan die gebruikers op hun servers kunnen installeren. Als er op deze servers dan een kwaadaardig IP-adres gedetecteerd wordt, slaagt Crowdsec dit IP-adres en de informatie hierover op. Crowdsec kan deze informatie dan tonen aan andere gebruikers.

Er kan op verschillende manieren gebruik gemaakt worden van Crowdsec. De eerste manier is de IDS/IPS die zojuist besproken is. Door het installeren van de Crowdsec-agent, kan verdachte activiteit gedetecteerd worden. Uit deze activiteit wordt dan de nuttige informatie gehaald, in het geval van Crowdsec gaat dit vooral over het IP-adres, en deze informatie kan later opnieuw gebruikt worden om bedreigingen te voorkomen.

De tweede manier zijn de bloklijsten die Crowdsec aanbiedt. Per agent kunnen er via de webinterface van Crowdsec bloklijsten toegevoegd worden. Deze blocklijsten zijn opgesteld vanuit de informatie die het uit de IDS/IPS haalt, maar het maakt ook gebruik van informatie uit andere Threat Intelligenceplatformen. Met behulp van deze informatie kunnen verbindingen geblokkeerd en bedreigingen tegengegaan worden.

De laatste manier is hoe Crowdsec in het SOC gebruikt gaat worden. Crowdsec biedt zijn eigen CTI-platform (cyber threat intelligence) aan. Dit hanteert hetzelfde principe als VirusTotal, maar dan met enkel IP-adressen. Het heeft ook een API die gebruikt kan worden binnen het SOC om automatische analyses uit te voeren.

MOGELIJKHEDEN MET CROWDSEC

Crowdsec is, zoals eerder al vermeld, vooral gefocust rond het analyseren van IP-adressen. Hieronder wordt er besproken welke informatie dit biedt voor de gebruiker.

IP-adressen

IP-adressen zijn één van de belangrijkste observables om te onderzoeken. Dit kan in het SOC aan de hand van de API. Hiermee kan een IP-adres onderzocht worden en wordt er getoond of het een kwaadaardig IP-adres is of niet. De kern van Crowdsec ligt dus heel kort bij die van VirusTotal, maar dan enkel met IP-adressen.
In Crowdsec wordt ongeveer dezelfde informatie getoond als in VirusTotal. Hoewel het niet laat zien hoeveel antivirussen het IP herkennen, toont het hoe groot het risico is dat verbonden is aan het IP-adres. Het toont het land van herkomst en hoe vaak het recent voorbij gekomen is in aanvallen. De aanvallen die vanuit het IP opgezet zijn, worden ook getoond op de site zodat gebruikers een beter beeld kunnen krijgen.

WAAROM CROWDSEC?

Ik heb Crowdsec gebruikt in mijn SOC voor dezelfde reden als VirusTotal. Het geeft duidelijke informatie over observables en helpt bij het uitsluiten van valse positieven. Des te meer informatie erover een observable beschikbaar is, des te beter dat er op gereageerd kan worden. Er wordt daarom bij dit deel opnieuw geen WRM gemaakt, omdat meer informatie altijd beter is.

Het gebruik van Crowdsec is volledig gratis. Iedereen kan een account aanmaken en gebruik maken van het CTI-platform en API. Dit is opnieuw in lijn met de vereisten om een zo laag mogelijk kostenplaatje te hebben en zoveel mogelijk open-source tools te gebruiken.

Tenslotte werd er een mogelijke integratie met Crowdsec gevraagd vanuit het MPI Oosterlo. Hoewel de originele gedachte hiervan de IDS/IPS was, heb ik met hen afgestemd dat dit gebeurt door Wazuh. In plaats daarvan heb ik, in samenspraak met mijn stagementor, ervoor gekozen om het te gebruiken als CTI.

3. SOC-realisatie

Nu alle tools geanalyseerd zijn, kan de realisatiefase beginnen. Om deze realisatiefase zo duidelijk mogelijk te beschrijven, wordt er per tool eerst de installatie en configuratie beschreven en daarna hoe de tool gebruikt werd. Er is voor deze structuur gekozen, om zoveel mogelijk transparantie te bieden aan het MPI Oosterlo VZW en aan u, de lezer.

Zonder te weten hoe alles geïnstalleerd en geconfigureerd is, wordt troubleshooting bij eventuele problemen moeilijk. Bij het verder uitbreiden van het SOC, zoals extra Wazuh-instanties toevoegen, is het ook belangrijk dat ze compatibel zijn met de instanties die al geconfigureerd zijn. Daarom wordt er per tool in detail bekeken hoe het is opgezet. Hierbij komen ook de configuratiebestanden in detail aan bod, zodat er zeker geen informatie overgeslagen wordt.

Na de installatieprocedure volgt de uitleg over hoe de tools gebruikt zijn. Hierin is informatie te vinden over de webinterface en aanpassingen die hier gemaakt zijn, zoals de workflow in Shuffle. Er wordt in detail uitgelegd hoe alle aanpassingen gemaakt zijn en wat hun doel is.

3.1. SIEM

Het opbouwen van een SOC begint bij het opzetten van een goed SIEM. Zonder een werkend SIEM kan er geen alert gemaakt worden en kan er dus ook geen response opgezet worden. Daarom begint de realisatie met de uitleg over Wazuh.

In Wazuh krijgen de alerts niet allemaal hetzelfde formaat. Daarom is er ook op zoek gegaan naar een oplossing hiervoor. Er is gebruik gemaakt van Fluentd om alerts naar Graylog te sturen. In Graylog zouden de alerts omgevormd kunnen worden om een uniform formaat te maken. Hoewel dit uiteindelijk niet werkte, kunt u de uitleg hierover in dit document terugvinden.

3.1.1. Wazuh

Voor het SIEM wordt er gemaakt van Wazuh. Zoals al eerder vermeld werd, is Wazuh een open-source SIEM-oplossing die helpt bij het monitoren van endpoints en detecteren van alerts. Dit vormt het hart van ons SOC en is het platform dat als eerste opgezet wordt.

SETUP

Wazuh bestaat uit drie componenten: de Wazuh Indexer, de Wazuh Server en het Wazuh Dashboard. Deze componenten zijn essentieel om Wazuh goed te laten werken. Om de componenten samen te installeren, zijn er twee mogelijkheden die Wazuh ons aanbiedt. Enerzijds kunnen we alles op één virtuele machine (VM) plaatsen om het installatieproces makkelijker te maken. Dit gaat wel ten koste van toekomstige schaalbaarheid. Anderzijds kunnen de componenten op drie verschillende VM's geïnstalleerd worden. Dit kost meer resources en kan lastiger zijn, maar biedt de mogelijkheid om verder te schalen op lange termijn.

Ik heb beide opties geprobeerd, maar omdat mijn SOC zich in een dynamische omgeving bevindt waar regelmatig veranderingen plaatsvinden, leek het mij het beste om de laatste optie te kiezen. Hierdoor bied ik niet alleen een oplossing die ze de komende tijd kunnen gebruiken, maar ook één die aangepast kan worden aan toekomstige noden. Momenteel is alles met single-node geïnstalleerd, wat betekent dat er slecht één van elke component is. Er was nog geen behoefte om uit te breiden naar een multi-node cluster, omdat dit meer resources in beslag zou nemen.

Wazuh Indexer

Vooraleer de alerts gedetecteerd kunnen worden, moet er eerst een plaats zijn om ze op te slaan. Daarom begint de installatieprocedure van Wazuh met het installeren van de Wazuh Indexer. Er bevinden zich hier ook enkele belangrijke configuratiebestanden die nodig zijn om de Wazuh-omgeving op te zetten. Zonder de Wazuh Indexer is het dus niet mogelijk om de componenten met elkaar te laten werken.

a) Installatie

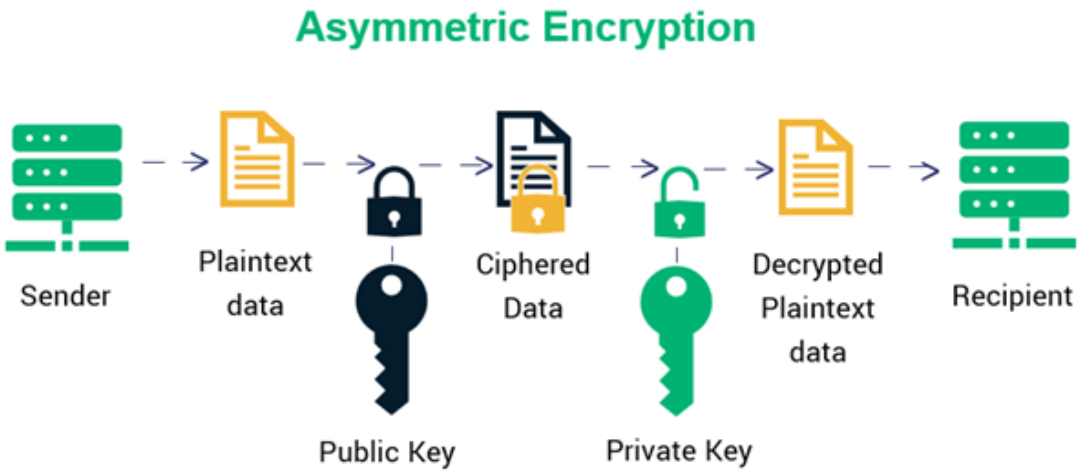
De installatiegids van Wazuh begint met installeren van de Wazuh Indexer. Dit is het centrale component waarop de alerts worden opgeslagen. Er wordt met de Indexer begonnen omdat hier het 'config.yml' bestand wordt opgeslagen, dat u hieronder kunt zien. Dit bestand bevat alle IP's van de verschillende componenten. Zo weet Wazuh welke verbindingen er opgezet worden. Ook worden hier de namen van de nodes bewaard, die we later gaan nodig hebben bij de configuratie van de componenten.

```
nodes:
  # Wazuh indexer nodes
  indexer:
    - name: node-1
      ip: "172.17.0.232"
    #- name: node-2
    #  ip: "<indexer-node-ip>"
    #- name: node-3
    #  ip: "<indexer-node-ip>"

  # Wazuh server nodes
  # If there is more than one Wazuh server
  # node, each one must have a node_type
  server:
    - name: wazuh-1
      ip: "172.17.0.225"
    #  node_type: master
    #- name: wazuh-2
    #  ip: "<wazuh-manager-ip>"
    #  node_type: worker
    #- name: wazuh-3
    #  ip: "<wazuh-manager-ip>"
    #  node_type: worker

  # Wazuh dashboard nodes
  dashboard:
    - name: dashboard
      ip: "172.17.0.233"
```

Figuur 2: Config.yml bestand uit mijn SOC

Tussen deze componenten wordt veel gevoelige informatie gedeeld. Wazuh voorziet daarom een script waarmee certificaten gegenereerd worden om de communicatie te encrypteren. De certificaten moeten hierna nog gecomprimeerd worden, zodat ze gedeeld kunnen worden tussen de verschillende VM's.

Voordat de Wazuh Indexer geïnstalleerd kan worden, moet er eerst een GPG-sleutel voorzien worden. Dit is een sleutelpaar dat bestaat uit een private en publieke sleutel. Dit sleutelpaar controleert softwarepakketten om na te kijken of ze afkomstig zijn van de maker en niet aangepast zijn door een derde partij. Het garandeert dat de bestanden niet corrupt of gemanipuleerd zijn tijdens de overdracht.
Wazuh maakt gebruik van asymmetrische encryptie. Hierbij wordt data versleuteld met een publieke sleutel en kan deze alleen worden ontgrendeld met de bijbehorende private sleutel, die Wazuh in bezit heeft. Op de figuur hieronder kunt u zien hoe dit in zijn werk gaat. Omdat de data enkel ontcijferd kan worden als er niks aangepast is, weten we dat de integriteit niet geschonden is en dat het bestand veilig is. Als er bij het decrypteren fouten opduiken, wordt er een waarschuwing uitgestuurd dat het bestand waarschijnlijk niet meer veilig is.



Figuur 3: Proces asymmetrische encryptie

Nu dat de integriteit gecontroleerd kan worden, kan de Wazuh Indexer geïnstalleerd worden op de VM. Wazuh voorziet hiervoor een package die eenvoudig te installeren is. In deze package zit alles wat nodig is om te kunnen beginnen aan de configuratie van de Wazuh Indexer, zoals Opensearch en de configuratiebestanden.

b) Configuratie

Nu alle nodige bestanden en software geïnstalleerd zijn, kan de Indexer geconfigureerd worden. De eerste aanpassingen gebeuren in het configuratiebestand van Opensearch, 'opensearch.yml', dat hieronder weergeven wordt. Het eerste wat aangepast wordt, is de 'network.host'. Dit bevat het IP-adres of de hostnaam waarnaar er geluisterd moet worden voor HTTP-verkeer alsook intern transportverkeer. Dit adres of hostnaam wordt gebruikt om verbindingen te accepteren.
Het volgende dat aangepast wordt is de 'node.name'. Dit bevat de naam van de node, zoals het ook aangegeven staat in de config.yml. Als deze naam hiervan afwijkt, zal Wazuh de node niet herkennen en kan er geen verbinding opgezet worden.
Het laatste dat nog aangepast moet worden is de 'cluster.initial_master_nodes'. Hieronder staan de namen van alle nodes, die zich in een cluster bevinden. In ons geval gaat dit slechts één node zijn, omdat we werken met een single-node installatie. Als er gebruik gemaakt wordt van een cluster, worden hier de namen van de andere nodes ingegeven, zoals ze ook opgegeven staan in de config.yml.

```
network.host: "172.17.0.232"
node.name: "node-1"
cluster.initial_master_nodes:
- "node-1"
#- "node-2"
#- "node-3"
cluster.name: "wazuh-cluster"
#discovery.seed_hosts:
#  - "node-1-ip"
#  - "node-2-ip"
#  - "node-3-ip"
node.max_local_storage_nodes: "3"
path.data: /var/lib/wazuh-indexer
path.logs: /var/log/wazuh-indexer

plugins.security.ssl.http.pemcert_filepath: /etc/wazuh-indexer/certs/indexer.pem
plugins.security.ssl.http.pemkey_filepath: /etc/wazuh-indexer/certs/indexer-key.pem
plugins.security.ssl.http.pemtrustedcas_filepath: /etc/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.transport.pemcert_filepath: /etc/wazuh-indexer/certs/indexer.pem
plugins.security.ssl.transport.pemkey_filepath: /etc/wazuh-indexer/certs/indexer-key.pem
plugins.security.ssl.transport.pemtrustedcas_filepath: /etc/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.http.enabled: true
plugins.security.ssl.transport.enforce_hostname_verification: false
plugins.security.ssl.transport.resolve_hostname: false

plugins.security.authcz.admin_dn:
- "CN=admin,OU=Wazuh,O=Wazuh,L=California,C=US"
plugins.security.check_snapshot_restore_write_privileges: true
plugins.security.enable_snapshot_restore_privilege: true
plugins.security.nodes_dn:
- "CN=node-1,OU=Wazuh,O=Wazuh,L=California,C=US"
#- "CN=node-2,OU=Wazuh,O=Wazuh,L=California,C=US"
#- "CN=node-3,OU=Wazuh,O=Wazuh,L=California,C=US"
plugins.security.restapi.roles_enabled:
- "all_access"
- "security_rest_api_access"

plugins.security.system_indices.enabled: true
plugins.security.system_indices.indices: [".plugins-ml-model", ".plugins-ml-task", ".opendistro-alerting-config", ".opendistro-alerting-alert*", ".opendist
```

Figuur 4: Screenshot opensearch.yml

De overige opties kunnen op hun standaardinstellingen blijven staan, aangezien ze voor deze stageopdracht niet relevant zijn om aan te passen. Hieronder geef ik een korte toelichting waarvoor de verschillende opties dienen:

- 'cluster.name': De naam van de cluster waarin deze node actief is.
- 'node.max_local_storage_nodes': Het maximaal aantal nodes op één fysieke host.
- 'path.data': De locatie waar Opensearch indexdata opslaat.
- 'path.logs': De locatie waar logbestanden opgeslagen worden.
- 'plugins.security.ssl.http.*': De locatie van de certificaten voor extern HTTPS-verkeer.
- 'plugins.security.ssl.transport.*': De locatie van de certificaten voor intern HTTPS-verkeer via API.
- 'plugins.security.ssl.http.enabled': Activeert HTTPS-verkeer op de API-interface.
- 'plugins.security.ssl.transport.enforce_hostname_verification': Bepaalt of de node verbindingen toelaat, ongeacht of de hostnaam overeen komt met de naam in het certificaat.
- 'plugins.security.ssl.transport.resolve_hostname': Bepaalt of Opensearch het IP-adres probeert te vertalen naar een hostnaam.
- 'plugins.security.authcz.admin_dn': De Distinguished Name van de admin-gebruiker.
- 'plugins.security.check_snapshot_restore_write_privileges': Controleert of een gebruiker schrijfrechten heeft voor snapshots en herstellingen.
- 'plugins.security.enable_snapshot_restore_privilege': Activeert snapshot- en herstelrechten voor gebruikers.
- 'plugins.security.nodes_dn': De Distinguished Names van de nodes in een cluster.

- ‘plugins.security.restapi.roles_enabled’: De rollen die toegang hebben tot de REST API
- ‘plugins.security.system_indices.enabled’: Het systeem herkent en beschermt interne indexen.
- ‘plugins.security.system_indices.indices’: De indexen die het systeem herkent en beschermt

Nu de Wazuh Indexer geïnstalleerd is en de Opensearch-database correct geconfigureerd werd, kunnen de certificaten uitgerold worden. Hiervoor maken we een nieuwe map aan, in dit geval ‘/etc/wazuh-indexer/certs’. Deze locatie moet overeenkomen met het pad dat in ‘opensearch.yml’ opgegeven werd. De certificaten worden vervolgens uitgepakt uit het eerder aangemaakte ZIP-bestand en in deze map geplaatst. Hierna kan de service voor de Wazuh Indexer aangezet en gestart worden.

Ten slotte rest enkel nog het initialiseren en testen van de cluster. Wazuh voorziet hiervoor een script dat zich bevindt op ‘/usr/share/wazuh-indexer/bin/indexer-security-init.sh’ en dat gebruikt wordt om de cluster op te starten.
Dit script is voornamelijk bedoeld om de beveiligingsinstellingen van de Wazuh Indexer te initialiseren of bij te werken. Het focust daarbij op TLS-certificaten en gebruikersrollen, en controleert of de configuratie in ‘opensearch.yml’ correct is ingesteld.
Zo worden de certificaatgegevens ingeladen, die in het configuratiebestand zijn opgegeven. Zoals eerder vermeld bevat ‘opensearch.yml’ paden naar de TLS-certificaten voor zowel HTTP- als transportbeveiliging. Daarnaast maakt het script gebruik van de rollen die eveneens in ‘opensearch.yml’ zijn gespecificeerd. Na controle initialiseert of herconfigureert het de gebruikers en rollen die nodig zijn voor een veilige communicatie tussen nodes of tussen Wazuh en de Indexer.
Tot slot zorgt het script ervoor dat een single-node of multi-node cluster op een veilige manier kan worden opgestart, met correcte authenticatie en versleuteling.
Hierna kunnen we de cluster installatie testen met het volgende commando:
curl -k -u admin:admin https://<WAZUH_INDEXER_IP_ADDRESS>:9200

Dit geeft de volgende output in de CLI:


```
{
  "name" : "node-1",
  "cluster_name" : "wazuh-cluster",
  "cluster_uuid" : "C9Pqes31QpiTe989N9MHjg",
  "version" : {
    "number" : "7.10.2",
    "build_type" : "deb",
    "build_hash" : "dae2bfc93896178873b43cdf4781f183c72b238f",
    "build_date" : "2025-04-30T10:51:28.815931460Z",
    "build_snapshot" : false,
    "lucene_version" : "9.12.1",
    "minimum_wire_compatibility_version" : "7.10.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "The OpenSearch Project: https://opensearch.org/"
}
```

Figuur 5: Output cluster installatie

Wazuh Server

Nu er een plaats is om alerts op te slaan, moeten er natuurlijk ook alerts binnenkomen. Hiervoor wordt de Wazuh Server opgezet. Deze communiceert met de endpoints om alerts op te halen en tijdelijk te bewaren, vooraleer ze naar de Wazuh Indexer gaan.

a) Installatie

De Wazuh Server bestaat uit twee grote componenten: de Wazuh Manager en Filebeat.
De Wazuh Manager is het hart van de Wazuh-opstelling. Het is verantwoordelijk voor het ontvangen, analyseren en indelen van data die binnenkomt via de Wazuh Agents. De manager voert loganalyse uit, detecteert bedreigingen, controleert de integriteit van bestanden, voert proactieve dreigingsdetectie uit en genereert alerts. Verder beheert het ook de configuratie van Agents en de regels die aangeven of iets verdacht is.
Filebeat is een logverzamelaar die ontworpen is om logbestanden te verzenden. In Wazuh wordt Filebeat gebruikt om deze logbestanden van de Wazuh Manager naar de Wazuh Indexer door te sturen. Dit maakt het mogelijk om de gegevens te indexeren en visueel te analyseren op het Wazuh Dashboard.

Vooraleer deze componenten geïnstalleerd kunnen worden, moet er opnieuw een GPG-sleutel geïnstalleerd worden, om te controleren of de softwarepakketten van Wazuh komen. Hierna kunnen de Wazuh Manager en Filebeat geïnstalleerd worden. Hiermee is de installatie van de Wazuh Server voltooid en kan er aan de configuratie begonnen worden.

b) Configuratie

De installatiegids van Wazuh begint met het configureren van Filebeat. Hiervoor moet eerst het configuratiebestand ‘filebeat.yml’ geïnstalleerd worden. Wazuh voorziet hier een download voor, die met curl geïnstalleerd kan worden. Het configuratiebestand, zoals het in mijn SOC geconfigureerd is, kunt u op de schermafbeelding op de volgende pagina vinden.

Nu kan de configuratie van Filebeat beginnen. Voor mijn SOC moet enkel ‘hosts’ aangepast worden. Dit is een lijst met alle IP-adressen van de Indexers. Standaard staat dit op ‘localhost’, maar omdat onze Indexer zich op een andere VM bevindt, wordt het IP-adres hiervan ingegeven. In een multi-node cluster moeten alle IP-adressen van de Indexers opgegeven worden.

De overige opties kunnen op hun standaardinstellingen blijven staan, aangezien ze voor de stageopdracht niet relevant zijn om aan te passen. Hieronder geef ik een korte toelichting waarvoor de verschillende opties dienen:

- ‘protocol’: Het protocol dat gebruikt wordt om onderlinge verbindingen op te zetten.
- ‘username’: De gebruikersnaam van Opensearch, wordt als variabele in de keystore meegegeven.
- ‘password’: Het wachtwoord van Opensearch, wordt als variabele in de keystore meegegeven.
- ‘ssl.certificate_authorities’: De locatie van het certificaat dat gebruikt wordt om de Opensearch certificaten te valideren.
- ‘ssl.certficate’: Cliëntcertificaat dat Filebeat gebruikt om zich als vertrouwde cliënt te identificeren.
- ‘ssl.key’: Cliëntsleutel dat Filebeat gebruikt om te bewijzen dat het de eigenaar is van het certificaat.
- ‘setup.template.json.enabled’: Schakelt een aangepast JSON-template in voor indexen.
- ‘setup.template.json.path’: Locatie waar de aangepaste JSON-template zich bevindt.
- ‘setup.template.json.name’: Naam van de aangepaste JSON-template in Opensearch.
- ‘setup.ilm.enabled’: Schakelt Index Lifecycle Management in, wat het automatisch beheer van Indexen- en logretentie is.
- ‘filebeat.modules’: Activeert een Filebeat-module.
- ‘module’: Naam van de Filebeat-module.
- ‘alerts’: Schakelt het verzamelen van Wazuh-alerts in.
- ‘archives’: Schakelt het verzamelen van gearchiveerde Wazuh-logs in.
- ‘logging.level’: Stelt het logniveau van logs in.
- ‘logging.to_files’: Schakelt in dat logoutput naar bestanden wordt geschreven

- ‘logging.files’: Logs worden opgeslagen in ‘/var/log/filebeat’ met bestandsnaam ‘filebeat’. Het bewaart tot max 7 logbestanden, die elk leesrechten hebben voor iedereen en schrijfrechten voor de eigenaar.
- ‘logging.metrics.enabled’: Schakelt het loggen van prestatiestatestieken in.
- ‘seccomp’: Staat alle systeemaanroepen toe of blokkeert ze.
- ‘Syscalls’: Staat per naam systeemaanroepen toe of blokkeert ze.

```
# Wazuh - Filebeat configuration file
output.elasticsearch:
  hosts: ["172.17.0.232:9200"]
  protocol: https
  username: ${username}
  password: ${password}
  ssl.certificate_authorities:
    - /etc/filebeat/certs/root-ca.pem
  ssl.certificate: "/etc/filebeat/certs/filebeat.pem"
  ssl.key: "/etc/filebeat/certs/filebeat-key.pem"
setup.template.json.enabled: true
setup.template.json.path: '/etc/filebeat/wazuh-template.json'
setup.template.json.name: 'wazuh'
setup.ilm.overwrite: true
setup.ilm.enabled: false

filebeat.modules:
  - module: wazuh
    alerts:
      enabled: true
    archives:
      enabled: false

logging.level: info
logging.to_files: true
logging.files:
  path: /var/log/filebeat
  name: filebeat
  keepfiles: 7
  permissions: 0644

logging.metrics.enabled: false

seccomp:
  default_action: allow
  syscalls:
  - action: allow
    names:
    - rseq
```

Figuur 6: Schermafbeelding filebeat.yml uit mijn SOC

Nu het configuratiebestand van Filebeat in orde is, moet er een keystore aangemaakt worden. Dit is een veilige opslagplaats voor gevoelige gegevens zoals wachtwoorden, API-keys en certificaten. Filebeat gaat dit gebruiken om de gebruikersnaam en het wachtwoord van Opensearch op te slaan, zodat dit niet in het configuratiebestand staat. De Filebeat keystore is onderdeel van Elastic Beats en is bedoeld voor gebruik met Elasticsearch. Omdat Opensearch een fork is van Elasticsearch, is dit grotendeels compatibel met Elastic Beats. Daardoor kan hier toch gebruik van gemaakt worden in het SOC.
Om de gegevens op te slaan, wordt er gebruik gemaakt van de keystore-tool. Deze tool dient om op een gebruiksvriendelijke manier aanpassingen in de keystore te maken. Als de gegevens hierin zijn opgeslagen, zijn ze bereikbaar voor alle configuratiebestanden op de Wazuh Server. Zou het wachtwoord van OpenSearch ooit aangepast worden, moet het enkel in de keystore veranderd worden en niet in alle aparte configuratiebestanden. Dit zorgt voor extra flexibiliteit.

Om de configuratie van Filebeat af te ronden, moeten er nog enkele belangrijke bestanden gedownload worden van Wazuh. Dit zijn de Wazuh alerts template en de Wazuh module, die daarnet besproken zijn in het configuratiebestand van Filebeat.
De Wazuh alerts template is een bestand dat de algemene structuur bepaalt van alerts. Het bevat informatie over hoe logdata en alerts correct gestructureerd, verwerkt en geïndexeerd moeten worden. Zo kunnen deze gegevens vlot doorzocht en gevisualiseerd worden. Het bestand krijgt ook leesrechten voor de groep die het bestand bezit en alle andere gebruikers.
De Wazuh module is een kant-en-klare configuratie waarmee Filebeat automatisch Wazuh logs en alerts kan inlezen, verwerken en doorsturen naar Opensearch.

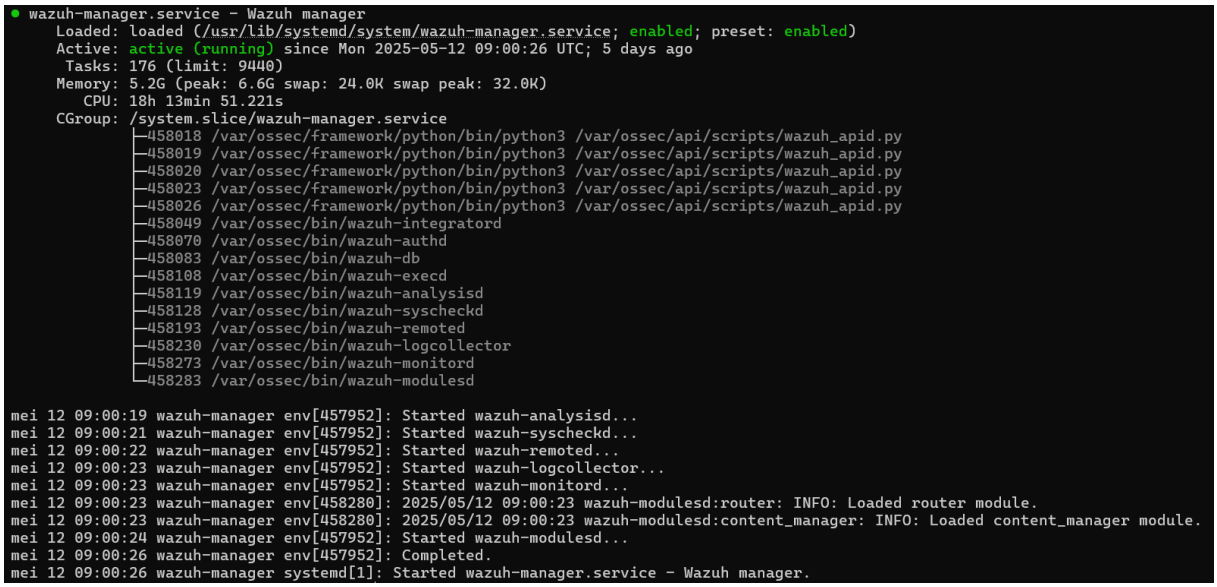
Nu Filebeat volledig geconfigureerd is, kunnen de certificaten uitgerold worden. Dit zijn dezelfde die bij de Wazuh Indexer gebruikt zijn en waarvan het TAR-bestand gekopieerd is naar deze node. De certificaten dienen in de Wazuh Server voor de verbinding tussen Filebeat en Opensearch, dat zojuist besproken is, en de verbinding tussen de Wazuh Manager en Wazuh Indexer te beveiligen, wat zo dadelijk besproken wordt. Het proces van het uitrollen van certificaten verloopt overal hetzelfde. Het begint met het opzetten van een nieuwe map, waar de certificaten in komen. In de Wazuh Server wordt dit bijgehouden in '/etc/filebeat/certs' dat lees- en uitvoerrechten geeft aan de eigenaar van de map. Het TAR-bestand wordt uitgepakt op deze locatie en de eigenaar van de certificaten krijgt hier leesrechten op. De eigenaar van zowel de map en de bestanden hierin, wordt aangepast naar het rootaccount.

De laatste grote configuratiestap die nog resteert, is het configureren van de Wazuh Manager. Dit is nodig om een connectie op te zetten tussen de Wazuh Server en Wazuh Indexer. De Wazuh Server bezit namelijk het centrale configuratiebestand 'ossec.conf' wat bepaalt hoe de Wazuh Server en Agent zich gedragen en hoe deze met andere componenten communiceren.
Vooraleer we hier aanpassingen aan gaan maken, moeten eerst gegevens worden toegevoegd aan de Wazuh keystore. Deze keystore hanteert hetzelfde principe als de Filebeat keystore, maar dan voor de Wazuh Manager. De inloggegevens van de Wazuh Indexer worden toegevoegd aan deze keystore met de wazuh-keystore tool. Deze inloggegevens gaan later gebruikt worden in configuratiebestanden om een connectie te maken met de Wazuh Indexer.
Hierna kan de configuratie in 'ossec.conf' aangepast worden. Hierin wordt het blok 'Indexer' aangepast, dat u hieronder kunt zien, met de nieuwe informatie. In het blok 'hosts' wordt het IP-adres van de host aangepast naar het IP-adres van de Wazuh Indexer. Bij een multi-node cluster kunnen hier nog meerdere hosts en hun IP-adres aan toegevoegd worden.
Het blok 'ssl' bevat de paden naar de locaties van de certificaten om de communicatie tussen de Wazuh Manager en Indexer veilig te maken. Dit is voor het SOC niet aangepast en blijft op de standaardinstellingen staan.
Aangezien het volledige 'ossec.conf'-bestand meerdere pagina's in beslag neemt, wordt hier niet alles van toegelicht in dit document. U kunt het volledige bestand vinden in Bijlage 1.

```
<indexer>
  <enabled>yes</enabled>
  <hosts>
    <host>https://172.17.0.232:9200</host>
  </hosts>
  <ssl>
    <certificate_authorities>
      <ca>/etc/filebeat/certs/root-ca.pem</ca>
    </certificate_authorities>
    <certificate>/etc/filebeat/certs/filebeat.pem</certificate>
    <key>/etc/filebeat/certs/filebeat-key.pem</key>
  </ssl>
</indexer>
```

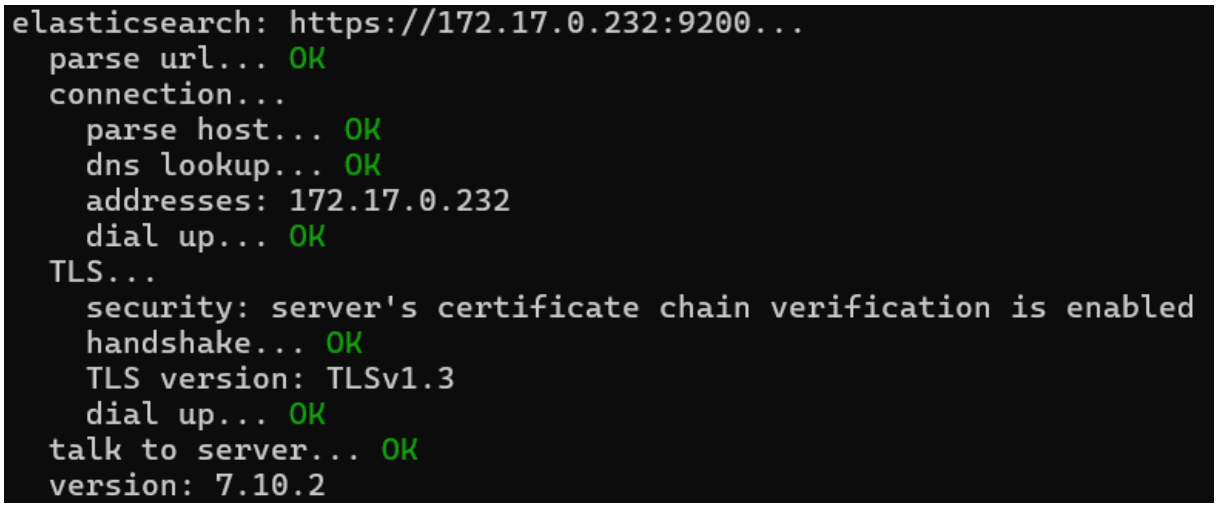
Figuur 7: Schermafbeelding uit ossec.conf: Indexer

Alle configuraties zijn nu gebeurd en er rest enkel nog om alle services aan te zetten en te testen of het werkt. Eerst wordt de wazuh-managerservice aangezet. Deze controleert of de configuratie goed ingesteld is en voert dit uit. Bij aanpassingen in de toekomst aan de configuratie, zal deze service ook telkens opnieuw opgestart moeten worden. Het wordt aangeraden om te testen of de service goed is opgestart, om verdere problemen te voorkomen. Als er toch fouten zijn, worden deze ook weergeven in de logs van de wazuh-managerservice. Hieronder vindt u een schermafbeelding van de wazuh-managerservice als hij succesvol aanstaat.



Figuur 8: Schermafbeelding status wazuh-managerservice

Als de wazuh-manager goed is opgestart, kan de service van filebeat aangezet worden. Net als bij de wazuh-managerservice controleert de filebeatservice of de configuratie goed ingesteld is en voert dit uit. Om te controleren of de service goed is opgestart, heeft Filebeat zijn eigen test. Dit is ‘filebeat test output’, dat de verbinding met Opensearch en de TLS-certificaten gaat testen.
Hieronder kunt u in de schermafbeelding de output van dit commando zien als alles goed werkt.



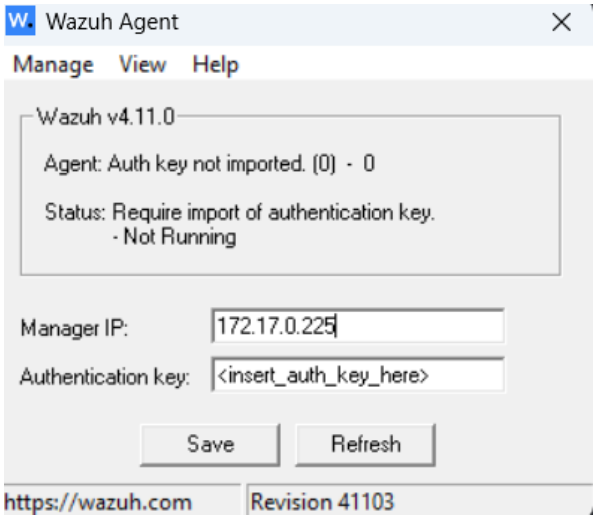
Figuur 9: Schermafbeelding testoutput Filebeat

Wazuh Agent

De Wazuh Manager is natuurlijk niks zonder een Wazuh Agent waar de logs en alerts vandaan komen. Omdat de Wazuh Agent op verschillende besturingssystemen geïnstalleerd kan worden, voorziet Wazuh hiervoor verschillende installatiegidsen. Op mijn stageplaats wordt vooral gebruik gemaakt van Windows- en een paar Linux-servers. Daarom worden enkel de installaties hiervan besproken in dit realisatiedocument.

a) Windows

De installatie op Windows gebeurt met een EXE-bestand, dat geïnstalleerd kan worden vanuit de website van Wazuh. Dit bestand installeert de Wazuh Agent op het endpoint. Dit geeft een configuratiescherm, dat u in de afbeelding hieronder ziet, waarin het IP-adres van de Wazuh Server ingevuld moet worden.



Figuur 10: Schermafbeelding configuratiescherm Wazuh Agent

Als de gebruiker dit scherm niet opent, moet het handmatig toegevoegd worden aan het 'ossec.conf' bestand. Dit bestand bevindt zich in 'C:\Program Files (x86)\ossec-agent'. Hier kunnen echter wel enkele problemen opduiken met toegangsrechten tot dit bestand voor gebruikers. Daarom heb ik een uitgebreide handleiding geschreven voor het MPI Oosterlo. Deze kunt u vinden in Bijlage 2.

Ten slotte moet de wazuh-agentservice opnieuw opgestart worden om de installatie te voltooien. Na enkele minuten is het nieuwe endpoint toegevoegd.

b) Linux

De installatie op Linux is korter met minder stappen dan de installatie op Windows. Het begint met het installeren van de GPG-sleutel. Deze wordt gebruikt om de installatie van de Wazuh agent te controleren. Hierna kan de Wazuh agent geïnstalleerd worden.

Als de Wazuh Agent geïnstalleerd is, wordt er geen configuratiescherm getoond. Het IP-adres van de server moet handmatig in het bestand 'ossec.conf', dat zich bevindt in '/var/ossec/etc', toegevoegd worden. Op onderstaande schermafbeelding kunt u de aanpassing in 'ossec.conf' zien.

```
<client>
  <server>
    <address>172.17.0.225</address>
    <port>1514</port>
    <protocol>tcp</protocol>
  </server>
  <config-profile>ubuntu, ubuntu24, ubuntu24.04</config-profile>
  <notify_time>10</notify_time>
  <time-reconnect>60</time-reconnect>
  <auto_restart>yes</auto_restart>
  <crypto_method>aes</crypto_method>
</client>
```

Figuur 11: Schermafbeelding 'ossec.conf' op Linux-endpoint

Wazuh Dashboard

Ten slotte moet enkel nog het Dashboard opgezet worden. Hierop kunnen gebruikers de informatie en alerts zien die Wazuh verzameld heeft.

a) Installatie

Het laatste component dat nog geïnstalleerd moet worden, is het Wazuh Dashboard. Dit is het component dat centraal alles toont dat op de andere componenten gedaan wordt. De logs en alerts die door de Wazuh Agent naar de Wazuh Server gestuurd worden, kunnen hier bekeken en gefilterd worden op de indexen bepaalt door de Wazuh Indexer.

De installatie van het Wazuh Dashboard begint, net als bij alle andere installaties, met het installeren van de GPG-sleutel. Nu kan de installatie gevalideerd worden en het Wazuh Dashboard geïnstalleerd worden. Wazuh voorziet hiervoor een pakket waarin alle bestanden zitten, zodat er direct na de installatie aan de configuratie begonnen kan worden.

b) Configuratie

Om het Wazuh Dashboard te configureren, voorziet Wazuh het configuratiebestand ‘opensearch_dashboards.yml’, dat zich in ‘/etc/wazuh-dashboard’ bevindt.

In het configuratiebestand wordt als eerste ‘server.host’ aangepast. Hier komt het IP-adres van het Wazuh Dashboard te staan, waarmee de andere hosts en gebruikers kunnen verbinden.
Het tweede dat wordt aangepast is ‘opensearch.hosts’. Dit zijn de hosts waar een Wazuh Indexer op draait. Het Wazuh Dashboard moet hiermee kunnen verbinden om data en de indexering hiervan op te halen.

De overige opties kunnen op hun standaardinstellingen blijven staan, aangezien ze voor de stageopdracht niet relevant zijn om aan te passen. Hieronder geef ik een korte toelichting waarvoor de verschillende opties dienen:

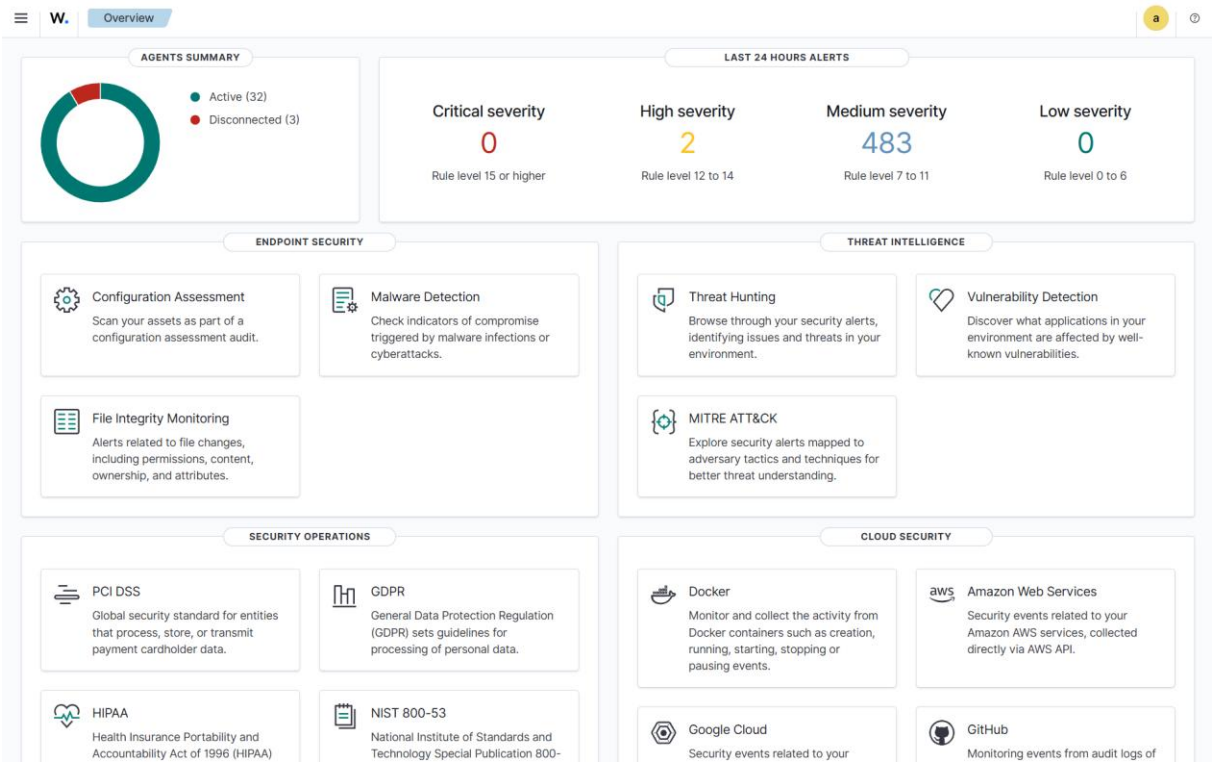
- ‘server.port’: De poort waarop het Dashboard draait.
- ‘opensearch.ssl.verificationMode’: De manier waarop er gecontroleerd wordt op SSL (TLS)
- ‘opensearch.requestHeadersAllowlist’: De lijst HTTP-headers die toegestaan zijn om van het Dashboard naar Opensearch door te sturen.
- ‘opensearch_security.multitenancy.enabled’: Bepaalt of er meerdere data-omgevingen voor meerdere gebruikers mag zijn.
- ‘opensearch_security.readonly_mode.roles’: Rollen voor gebruikers die enkel het Dashboard mogen lezen.
- ‘server.ssl.enabled’: Schakelt SSL in voor HTTPS-verbindingen.
- ‘server.ssl.key’: De locatie van de private sleutel.
- ‘server.ssl.certificate’: De locatie van het certificaat.
- ‘opensearch.ssl.certificateAuthorities’: De locatie van het CA-certificaat dat het Opensearch SSL-certificaat controleert.
- ‘uiSettings.overrides.defaultRoute’: De standaardpagina die geladen wordt bij het openen van het Dashboard

```
server.host: 172.17.0.233
server.port: 443
opensearch.hosts: https://172.17.0.232:9200
opensearch.ssl.verificationMode: certificate
#opensearch.username:
#opensearch.password:
opensearch.requestHeadersAllowlist: ["securitytenant","Authorization"]
opensearch_security.multitenancy.enabled: false
opensearch_security.readonly_mode.roles: ["kibana_read_only"]
server.ssl.enabled: true
server.ssl.key: "/etc/wazuh-dashboard/certs/dashboard-key.pem"
server.ssl.certificate: "/etc/wazuh-dashboard/certs/dashboard.pem"
opensearch.ssl.certificateAuthorities: ["/etc/wazuh-dashboard/certs/root-ca.pem"]
uiSettings.overrides.defaultRoute: /app/wz-home
```

Figuur 12: Schermafbeelding opensearch_dashboards.yml uit mijn SOC

DASHBOARD

Op onderstaande afbeelding kunt u het dashboard zien na de installatie van alle componenten. Het bevat veel informatie dat in aparte vakken wordt geplaatst om een duidelijk overzicht te bieden aan de gebruiker. In dit deel van het document, worden de belangrijkste vakken overlopen en beschreven wat deze doen.



Figuur 13: Wazuh Dashboard van MPI Oosterlo

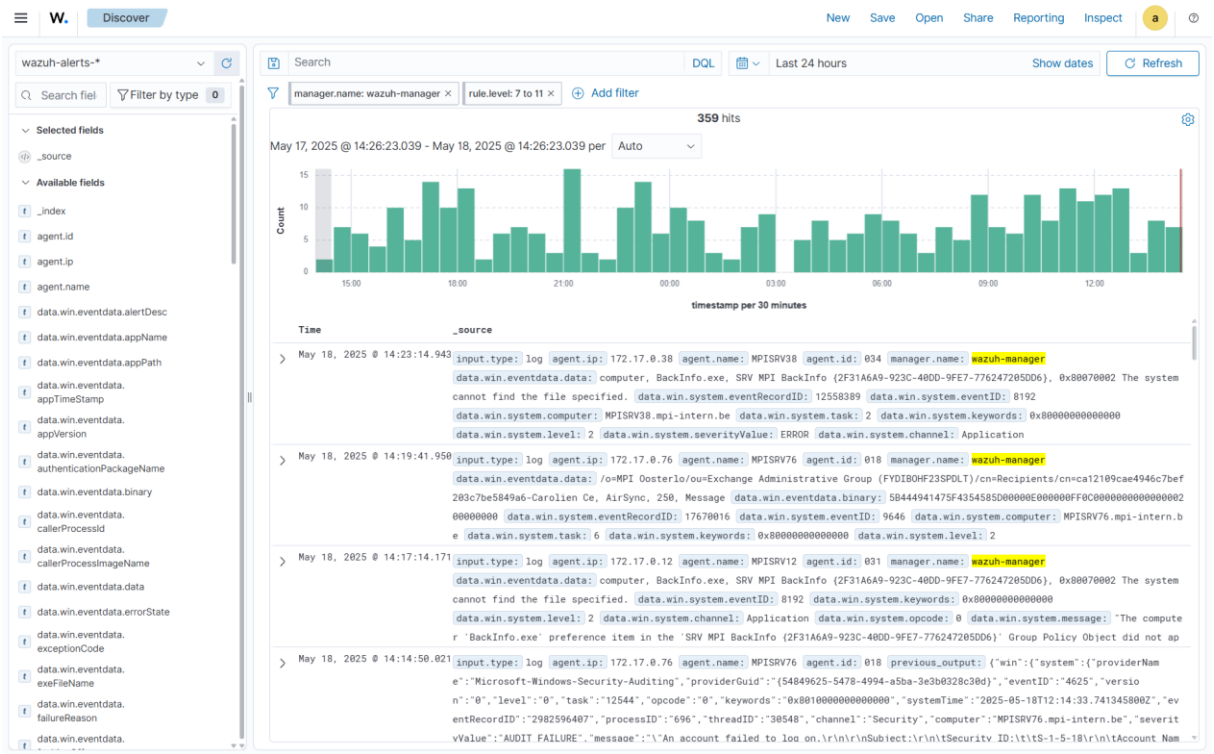
Alerts

Het belangrijkste tabblad, dat rechtsboven zichtbaar is op het Dashboard, zijn de Alerts. Hier staan alle alerts en logs die Wazuh heeft binnengekregen binnen een bepaalde tijdsduur (standaard 24 uur). Op de schermafbeelding kunt u alle alerts zien die in de afgelopen 24 uur op het SOC gegenereerd zijn.
Wazuh heeft vier categorieën op basis van meldingsniveau:

- Low severity: niveau van 0-6
- Medium severity: niveau van 7-11
- High severity: niveau van 12-14
- Critical severity: niveau van 15

Het hoofdonderdeel van dit tabblad is natuurlijk om alle alerts te laten zien. Het doet dit door ze onder elkaar te plaatsen en een kleine uitleg toe te voegen waarover de alert gaat. Deze alerts kunnen opengeklapt worden om meer informatie te tonen. Deze informatie wordt getoond in de vorm van een tabel of in JSON-formaat. Aan de hand hiervan kan er bepaald worden hoe de alerts opgelost kunnen worden. Als er dan een oplossing gevonden is, is het makkelijk om dit probleem elders ook op te lossen. Doordat de Wazuh Indexer alle data al geïndexeerd heeft, kan de gebruiker in het Dashboard eenvoudig filteren op deze indexen. Dit kan de naam van de agent zijn of het type alert dat gegenereerd is.

Een andere manier waarop de alerts getoond worden, is in de grafiek die boven de alerts staat. Deze geeft de alerts weer die per half uur gegenereerd zijn. Als er dan momenten zijn met veel alerts of iemand wil kijken of de alert zich vaker voordoet in korte tijdsspanne, kan er gefilterd worden door op de balk van de grafiek te klikken. Wazuh toont dan enkel de alerts die zich in die periode hebben voorgedaan.
Wazuh kan ook meer dan enkel de afgelopen 24 uur laten zien. Er kan gefilterd worden tot meerdere dagen en ook kleinere intervallen zijn mogelijk. Dit voorziet flexibiliteit voor gebruikers die willen kijken wat er tijdens het weekend gebeurd is.



Figuur 14: Schermafbeelding tabblad Alerts

Agent Summary

Een van de belangrijkste onderdelen van Wazuh is de ‘Agent Summary’. Hierin krijgt de gebruiker een overzicht te zien van alle endpoints die verbonden zijn met de Wazuh Server. Van deze endpoints krijgt Wazuh alerts en logs doorgestuurd voor analyse. Op de schermafbeelding hieronder kunt u zien hoe dit onderdeel eruitziet.

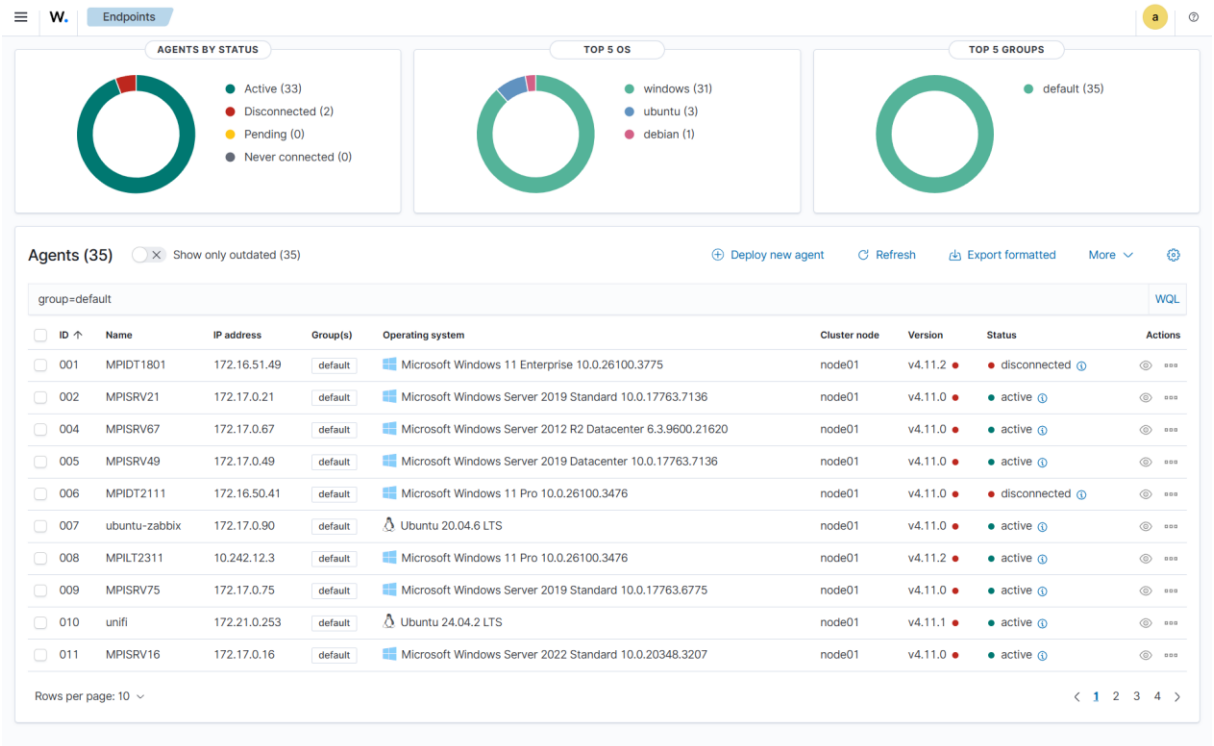
Om te beginnen zien we linksboven de status van alle endpoints. Meestal is dit Active of Disconnected, maar bij het toevoegen van een nieuwe endpoint komt deze eerst in Never connected en daarna in Pending. Een endpoint die status Disconnected heeft, betekent meestal dat deze uitstaat. Er kan dus hier gecontroleerd worden of een server uitgevallen is.

Naast de status van de endpoints staan de top vijf meest gebruikte besturingssystemen. Dit is handig om te weten welke besturingssystemen het vaakst voorkomen. Als er dan een nieuwe CVE ontdekt wordt voor een besturingssysteem, kan er hier snel gecontroleerd worden welke servers bijgewerkt moeten worden.

Agents kunnen ingedeeld worden in verschillende groepen. Als een omgeving veel servers heeft met dezelfde functie, kunnen die in een groep gezet worden. Zo kan er bijvoorbeeld makkelijk naar alle fileservers gezocht worden. Aangezien het MPI Oosterlo geen grote omgeving heeft, was de onderverdeling in groepen niet van toepassing.

Het laatste tabblad toont alle Agents. Hier kan per Agent informatie gevonden worden over bijvoorbeeld het besturingssysteem, de naam, het IP-adres en de status. Het biedt een makkelijk overzicht over je omgeving en ze kunnen per 10, 25, 50 of 100 Agents getoond worden. Via hier kan er ook een Agent geopend worden. Dit toont dan de resultaten van de andere blokken op het Dashboard, maar dan voor de specifieke Agent. Ook kan hier de Inventory Data gevonden worden die bijvoorbeeld de open poorten weergeeft.

Ten slotte is het ook mogelijk om via hier een nieuwe Agent toe te voegen. Er moeten dan enkele gegevens ingevuld worden, zoals het besturingssysteem en het IP-adres. Aan de hand van deze informatie genereert Wazuh een commando dat, na het uitvoeren ervan op de endpoint, de Wazuh Agent toevoegt.



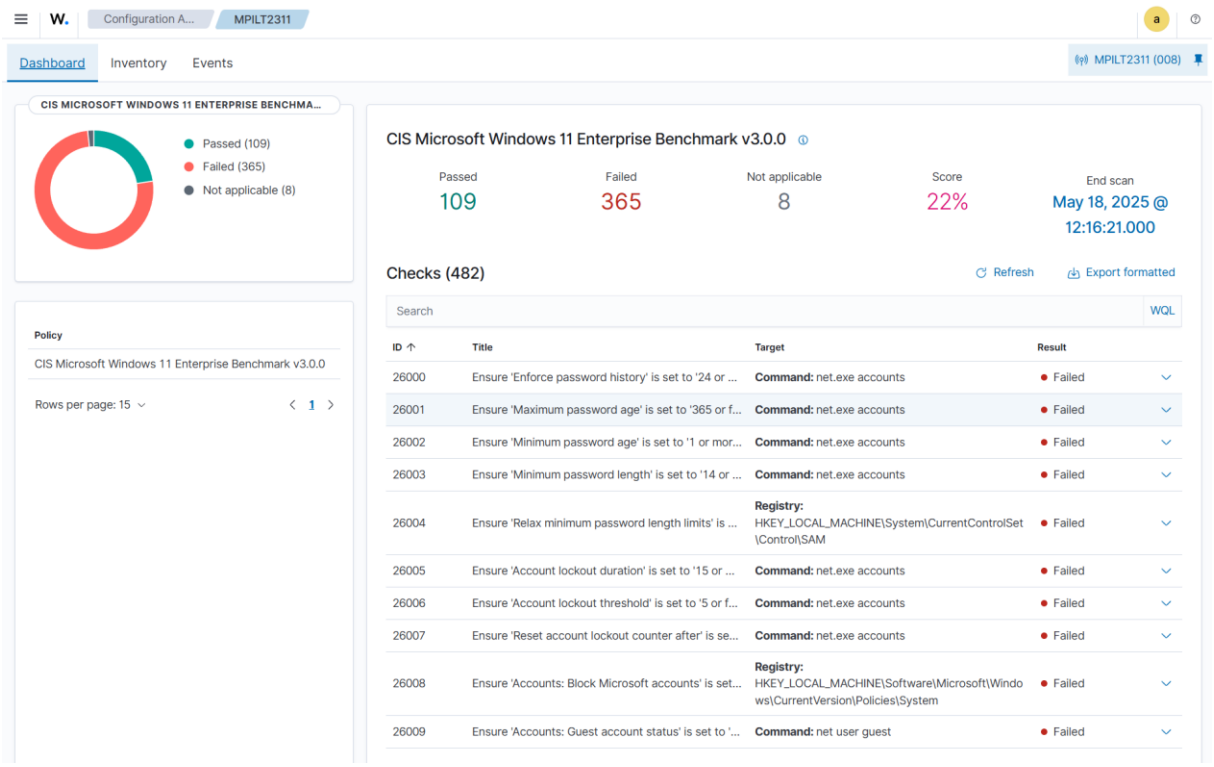
Figuur 15: Schermafbeelding tabblad ‘Agent Summary’

Configuration Assessement

Om te testen hoe veilig hoe een Agent is, doet Wazuh testen om een veiligheidsscore te bepalen. Deze veiligheidsscores zijn gebaseerd op CIS Benchmarks, die met behulp van een script test of een endpoint aan de vereisten voldoen. Zoals u op de schermafbeelding hieronder kunt zien, liggen deze scores vaak niet hoog. Dit komt omdat op persoonlijke toestellen, zoals de laptop die u hieronder kunt zien, er genoeg vrijheid moet zijn voor de gebruiker om niet te veel gehinderd te worden. Als de gebruiker bij elke actie zijn wachtwoord moet ingeven, dan is het systeem wel veilig maar niet bruikbaar.

Toch is het handig om deze benchmark in de gate te houden en er bepaalde zaken van te implementeren. Op kritieke servers waar weinig veranderingen op gebeuren, is het beter om een hogere score te hebben. De vereiste om gebruiksvriendelijk te zijn ligt hier lager dan de vereiste om veilig te zijn.

Wazuh laat zien welke testen het heeft uitgevoerd en welke hiervan geslaagd zijn of gefaald hebben. Een gebruiker kan makkelijk kijken welke maatregelen het kan implementeren, om zijn systeem beter te beveiligen.



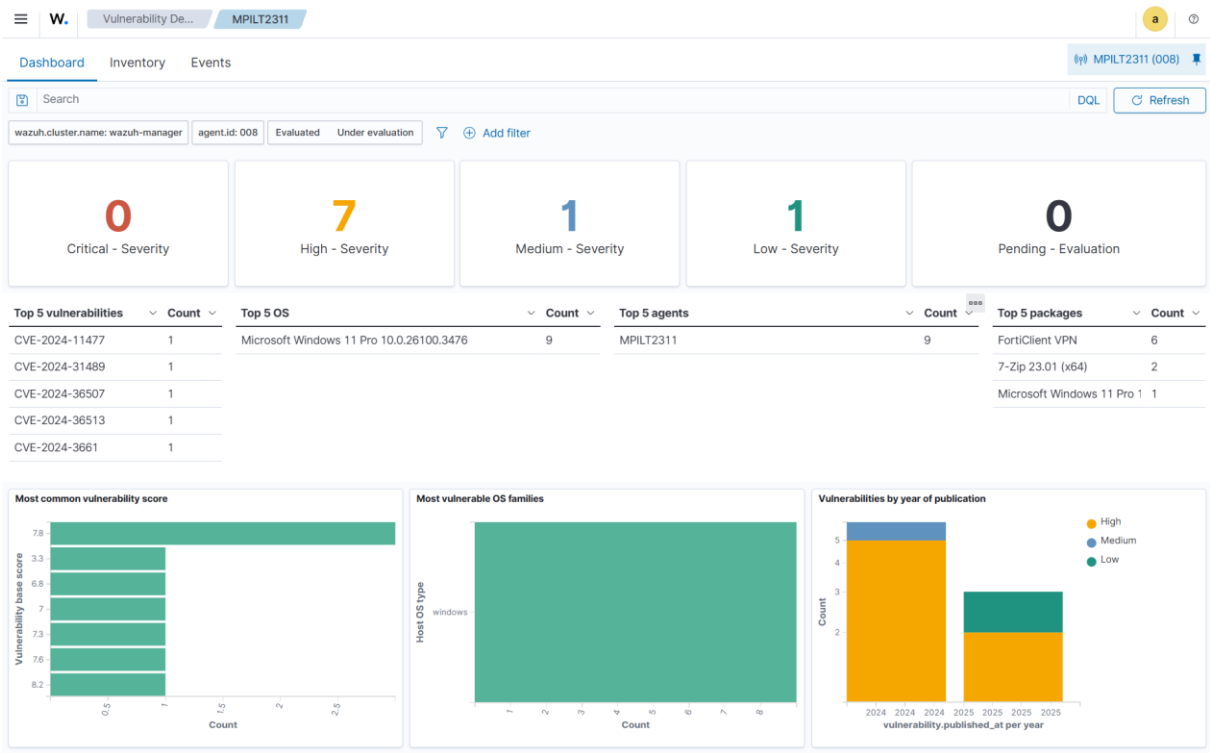
Figuur 16: Schermafbeelding tabblad 'Configuration Assessement'

Vulnerability Detection

CVE's zijn kwetsbaarheden die gevonden zijn in besturingssystemen en software. Ze worden geclassificeerd door de MITTRE ATT&CK framework zodat alle kwetsbaarheden dezelfde classificering hanteren. Deze CVE's zijn belangrijk om aanvallen tegen te gaan en systemen veilig te houden. Daarom biedt Wazuh het tabblad 'Vulnerability Detection' aan waar de CVE's per Agent zichtbaar zijn.
Wazuh verdeelt de CVE's in vijf categorieën op basis van hun CVSS-score:

- Critical: Score van 9.0-10.0
- High: Score van 7.0-8.9
- Medium: Score van 4.0-6.9
- Low: Score van 0.1-3.9
- Pending evaluation: Hebben nog geen score ontvangen

Onder deze categorieën laat Wazuh zien welke CVE's het meeste voorkomen en bij welk besturingssysteem. Het laat ook de package zien die de kwetsbaarheid bezit.
Hoewel Wazuh geen directe oplossing biedt, kan er gezocht worden naar de CVE om een oplossing te vinden.

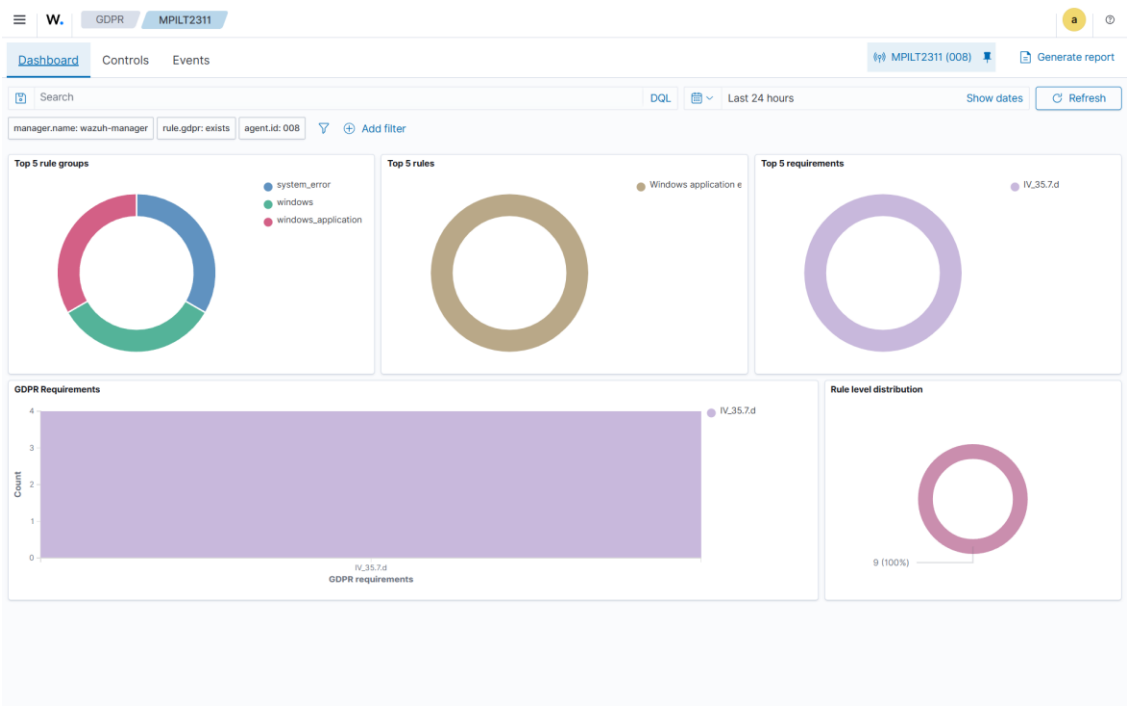


Figuur 17: Schermafbeelding tabblad 'Vulnerability Detection'

GDPR

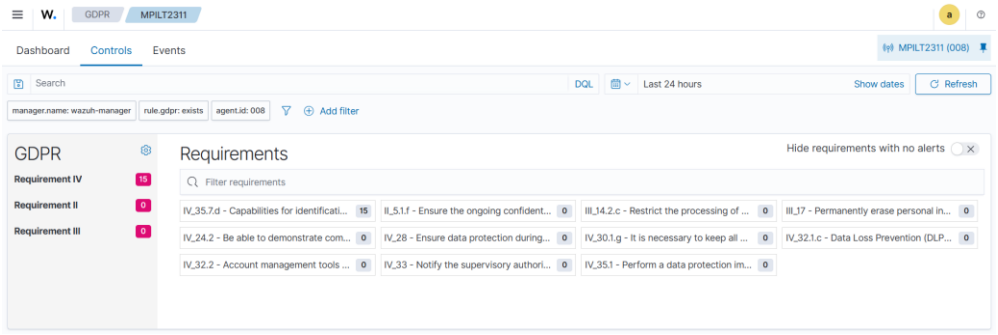
Het laatste tabblad dat nog besproken wordt, is het tabblad ‘GDPR’. Dit is de General Data Protection Regulation die is opgesteld door de Europese Unie voor het beschermen van data van de Europese burgers. Het MPI Oosterlo VZW verwerkt veel cliëntengegevens, waardoor data beveiliging van cruciaal belang is. Wazuh gebruikt de GDPR om te controleren hoe veilig Agents zijn en hoe goed ze voldoen aan de wetgeving.
Om dit te controleren gebruikt Wazuh de alerts en logs die het binnenkrijgt en analyseert het of het voldoet aan de wetgevingen van de GDPR. Als er ergens een fout is, laat Wazuh zien welke regel het niet aan voldoet.

Op het onderstaande Dashboard laat Wazuh zien aan welke regelgevingen er het vaakst niet wordt voldaan worden en welke alerts eraan gelinkt zijn. Dit biedt een duidelijk overzicht voor de gebruikers en zo kunnen ze ook snel zien aan welke regelgevingen ze het minste aan voldoen. Het Dashboard kan zowel individuele Agents laten zien, zoals op de schermafbeelding hieronder, of alle Agents samen.



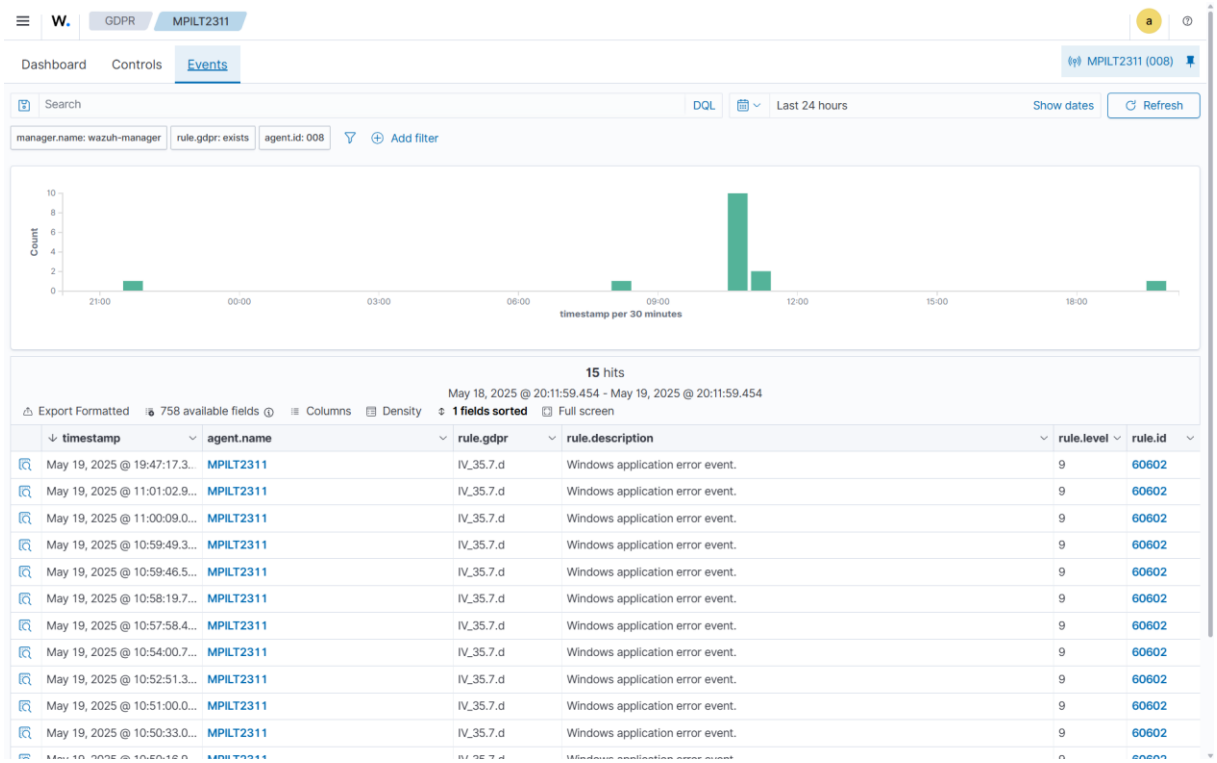
Figuur 18: Schermafbeelding tabblad ‘GDPR > Dashboard’

Als een gebruiker meer informatie wil over de verschillende regelgevingen, kan dit bij ‘Controls’ gevonden worden. Hier staan alle vereisten waar Wazuh op controleert en hoeveel alerts hieraan gelinkt zijn. Bij het openklikken van een vereiste, staat er een beschrijving en welke alerts eraan gelinkt zijn. Met deze informatie kan de gebruiker op zoek gaan naar een manier om zowel de alert op te lossen, alsook te voldoen aan de GPDR.



Figuur 19: Schermafbeelding tabblad ‘GPDR > Controls’

Ten slotte is er nog het tabblad Events. Hier krijgt de gebruiker een volledig overzicht over alle alerts en de vereisten van de GDPR die daaraan gelinkt zijn. Het toont dit op dezelfde manier als het alerts toont in het tabblad ‘Alerts’. Er kan hier ook makkelijk gefilterd worden op verschillende indexen zodat er een beter beeld gevormd kan worden van hoe vaak er niet aan een vereiste voldaan wordt en of dit terugkerende problemen zijn.



Figuur 20: Schermafbeelding tabblad ‘GDPR > Events’

Natuurlijk zijn dit niet de enige functies die Wazuh te bieden heeft. Het heeft tal van functies die nuttig zijn voor verschillende organisaties en gebruikers. Er is gekozen om deze functies te laten zien, omdat deze de belangrijkste zijn voor het gebruik van het SOC en waar het MPI Oosterlo het meeste mee in aanraking gaat komen.

CUSTOM RULES

Na het opzetten van Wazuh, inclusief de agents en het analyseren van alerts via het dashboard, kan het nodig zijn om bepaalde aanpassingen te doen. Zo kunnen sommige alerts een te laag meldingsniveau krijgen en dus verhoogd moeten worden. Andere alerts kunnen dan weer te vaak voorkomen in korte tijd, wat kan leiden tot "alert flooding". Om dit aan te pakken, biedt Wazuh de mogelijkheid om custom rules te maken. Deze bepalen hoe alerts gelogd worden of kunnen zelfs nieuwe alerts genereren op basis van meerdere voorgaande meldingen.

Een voorbeeld uit het SOC is het verhogen van het meldingsniveau bij meerdere mislukte SSH-inlogpogingen. Standaard krijgt zo'n incident meldingsniveau 10, maar omdat dit niet voldoende opvalt in de verdere afhandeling binnen het SOC, wordt het meldingsniveau verhoogd naar 13 wanneer meerdere meldingen kort na elkaar gedetecteerd worden. Deze custom rules worden beheerd in het bestand: '/var/ossec/etc/rules/local_rules.xml'.

Hieronder is een schermafbeelding te zien van dit 'local_rules.xml'-bestand. Alle aangepaste regels bevinden zich in de groep 'custom'. Binnen het SOC zijn er twee types custom rules opgesteld:

- Ignore rule: Deze houdt bij hoe vaak een alert met een bepaald ID voorkomt. Wanneer dit te vaak gebeurt binnen een bepaalde tijdspanne, wordt de alert tijdelijk genegeerd om overbelasting te voorkomen.
- Elevation rule: Deze verhoogt het meldingsniveau van alerts die standaard te laag geclassificeerd zijn, zodat ze beter opvallen.

Een voorbeeld van een 'ignore rule' is toegepast op alerts die gegenereerd worden door een ongeldig RADIUS-clientverzoek. Deze meldingen kwamen zo vaak voor, dat ze het zicht op andere alerts verstoorden. De ingestelde regel bepaalt dat wanneer er twee meldingen binnen 60 seconden binnenkomen, diezelfde alert vervolgens 900 seconden lang genegeerd wordt. Op deze manier blijft belangrijke informatie behouden, zonder dat één veelvoorkomende alert alles overstemt.

Voor de detectie van SSH brute-force aanvallen werd een 'elevation rule' gebruikt. Zo'n aanval probeert via veelgebruikte gebruikersnamen en wachtwoorden toegang te krijgen tot het systeem. Standaard werden deze meldingen met meldingsniveau 10 geregistreerd, wat als 'medium' wordt beschouwd. Dit was onvoldoende opvallend. Daarom werd een regel opgesteld die een nieuwe alert genereert met niveau 13 wanneer er vier alerts met niveau 10 binnen 60 seconden worden gedetecteerd. Zo is de bedreiging beter zichtbaar en kan er sneller worden ingegrepen. Omdat Wazuh meerdere alert-ID's gebruikt voor dit soort incidenten, zijn voor de meest voorkomende ID's afzonderlijke rules toegevoegd om niets over het hoofd te zien.

```
<group name="custom, ignore">
  <rule id="100002" level="10" frequency="2" timeframe="60" ignore="900">
    <if_matched_sid>61110</if_matched_sid>
    <description>The Trend Micro Unauthorized Change Prevention Service service depends on the tmactmon service which failed to start because of the follow
The dependency service or group failed to start.</description>
  </rule>

  <rule id="100003" level="13" frequency="4" timeframe="60">
    <if_matched_sid>5551</if_matched_sid>
    <description>PAM: Multiple failed logins in a small period of time. Possible brute force attack.</description>
  </rule>

  <rule id="100004" level="13" frequency="4" timeframe="60">
    <if_matched_sid>2502</if_matched_sid>
    <description>PAM: Multiple failed logins in a small period of time. Possible brute force attack.</description>
  </rule>
  <rule id="100005" level="13" frequency="4" timeframe="60">
    <if_matched_sid>5758</if_matched_sid>
    <description>PAM: Multiple failed logins in a small period of time. Possible brute force attack.</description>
  </rule>
</group>
```

Figuur 21: Schermafbeelding van rules uit 'local_rules.xml'

3.1.2. Graylog en Fluentd

Een nadeel van Wazuh is dat logs geen uniform formaat hanteren. Dit betekent dat de indexen, die gecreëerd worden door de Wazuh Indexer, niet altijd dezelfde namen hebben voor hetzelfde stuk data. In het SOC is er bijvoorbeeld het IP-adres waar een aanval vandaan komt. Wazuh geeft bij Windows de index ‘data.win.eventdata.clientIPAddress’ en bij Linux ‘data.srcip’. Zo wordt het moeilijk om hier automatisch op te filteren en mee te werken. Daarom is er op zoek gegaan naar een manier om de indexen aan te passen en één universele index te hebben, zoals bijvoorbeeld ‘srcip’.

Wazuh zelf biedt hier geen oplossing voor, dus is de zoektocht naar een oplossing hiervoor begonnen. Online wordt veel gesproken over de mogelijke integratie van Wazuh met Graylog om uniforme logs te krijgen. Graylog is een SIEM-oplossing, net als Wazuh, maar biedt ook de mogelijkheid om logs te transformeren. Dit houdt in dat met Graylog een uniform logformaat gemaakt kan worden. Om deze logs in Graylog te krijgen, wordt er gebruik gemaakt van Fluentd. Dit is een open source data verzamelaar die van verschillende bronnen gegevens kan verzamelen en doorsturen. In de integratie met Graylog zorgt het ervoor dat Wazuh logs in het juiste formaat staan, zodat Graylog ze kan accepteren.

Helaas heeft deze opstelling nooit naar behoren gewerkt. De logs kwamen niet aan in Fluentd of Graylog. Er is uitgebreid onderzoek gedaan naar de mogelijke oorzaken, maar er is geen definitieve oplossing gevonden. Een tijdelijke workaround is terug te vinden in het gedeelte over Shuffle. In dit hoofdstuk wordt de poging tot integratie met Wazuh toegelicht. De installaties van Graylog en Fluentd worden hierbij buiten beschouwing gelaten, aangezien deze geen directe invloed hadden op de werking van het SOC. De focus ligt dan ook uitsluitend op de geprobeerde integratie met Wazuh.

AANPASSINGEN OP VM VAN FLUENTD

Om de integratie met Wazuh op te zetten, is er gebruik gemaakt van Hadoop. Dit is een open source software die gebruikt wordt voor gegevensbewerking. Hadoop wordt vooral gebruikt om logs op te slaan, zodat Graylog niet overbelast wordt. Fluentd houdt geen rekening met de limieten die Wazuh instelt voor het doorsturen van logs naar Shuffle. In plaats daarvan stuurt Fluentd simpelweg alle logs die Wazuh ontvangt, ongefilterd door. Het gebruikt hiervoor Hadoop Distributed File System (HDFS), een bestandssysteem gemaakt voor Hadoop.

Voordat Hadoop geïnstalleerd kan worden, moeten er enkel aanpassingen gemaakt worden in het configuratiebestand van Fluentd, dat zich bevindt in '/etc/fluent/fluentd.conf'. Hierin wordt een nieuwe blok toegevoegd, dat alle logs met tag 'wazuh' naar Hadoop en de standaarduitvoer kopieert. Op de schermafbeelding hieronder kunt u configuratie in 'fluentd.conf' zien. Hieronder wordt kort toegelicht wat de verschillende opties doen:

- '<match wazuh>': Geeft aan welke tag een log moet bezitten om aan de vereisten te voldoen.
- '@type copy': Bepaalt dat de logs gekopieerd worden naar de opgegeven opslagplaatsen.
- '<store>': Het blok waarin een opslagplaats gedefinieerd wordt.
- '@type webhdfs': Geeft het type opslag weer, in dit geval HDFS.
- 'host': Het IP-adres voor de verbinding met de Hadoop-server op te zetten.
- 'port': De poort voor de verbinding met de Hadoop-server op te zetten.
- 'append': Bepaalt hoe nieuwe logs worden opgeslagen in HDFS.
- 'path': Het pad naar het bestand waar nieuwe logs naar worden opgeslagen.
- '<buffer>':
- 'flush_mode': Bepaalt hoe nieuwe logs weggeschreven worden naar HDFS, in dit geval direct in plaats van in batches.
- '<format>': Het blok waarin het logformaat bepaald wordt.
- 'type': Het logformaat waarin de logs bijgehouden worden.

```
<match wazuh>
  @type copy
  <store>
    @type webhdfs
    host localhost
    port 9870
    append yes
    path "/Wazuh/%Y%m%d/alerts.json"
    <buffer>
      flush_mode immediate
    </buffer>
    <format>
      @type json
    </format>
  </store>

  <store>
    @type stdout
  </store>
</match>
```

Figuur 22: Schermafbeelding uit 'fluentd.conf' van blok om logs naar HDFS te sturen

Er kan nu een nieuwe gebruiker aangemaakt worden voor Hadoop. Deze gebruiker zal gebruikt worden om een SSH-verbinding op te zetten met de Hadoop-server, zonder telkens een wachtwoord te moeten invoeren. Hiervoor moeten eerst SSH-sleutels gegenereerd worden voor de nieuwe gebruiker en toegevoegd worden aan de lijst met 'authorized keys'.

Vervolgens kan Hadoop geïnstalleerd worden. De aangemaakte gebruiker wordt daarbij de eigenaar van de map hadoop, die zich bevindt in '/usr/local'. In deze map staan alle bestanden en services die nodig zijn om Hadoop correct te laten functioneren.

Vooraleer de services aangezet kunnen worden, moet er eerst nog een kleine aanpassing gemaakt worden in het configuratiebestand 'core-site.xml' dat zich bevindt in '/usr/local/hadoop/etc/hadoop'. Deze aanpassing specifieert hoe HDFS bereikt wordt vanuit Hadoop. Op de schermafbeelding hieronder ziet u de configuratie in 'core-site.xml'. Hieronder wordt kort toegelicht wat de verschillende opties doen:

- '<property>': Elke instelling in de configuratiebestanden van Hadoop, worden in een 'property'-blok geplaatst.
- 'name': De naam van de instelling die veranderd wordt, in dit geval het standaard bestandssysteem.
- 'value': De waarde die toegevoegd wordt aan de instelling, in dit geval het nieuwe bestandssysteem en waar het zich bevindt.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

Figuur 23: Schermafbeelding van 'core-site.xml'

Nu kan het Hadoop-bestandssysteem klaargemaakt worden voor gebruik en de nodige services gestart worden. Hadoop is nu bereikbaar via een webinterface op het IP-adres van de VM en de poort 9870. Op HDFS moet er nog een folder aangemaakt worden waar logs opgeslagen worden. Deze moet overeenkomen met wat er in 'fluentd.conf' gespecifieerd is, in dit geval '/Wazuh'.

De laatste aanpassing die moet gebeuren, is in het configuratiebestand ‘hdfs-site.xml’, dat zich bevindt in ‘/usr/local/hadoop/etc/hadoop’. Hier moeten toevoegbewerkingen ingeschakeld worden in HDFS. Zonder dit toe te voegen aan de configuratie, kunnen er geen nieuwe logs toegevoegd worden aan de HDFS.
De eerste functie die wordt ingeschakeld, is de ‘dfs.webhdfs.enabled’-functie. Deze zorgt ervoor dat er via de webinterface aanpassingen gemaakt kunnen worden in de HDFS.
De tweede functie is de ‘dfs.support.append’-functie. Deze zorgt ervoor dat bestanden toegevoegd mogen worden aan de HDFS.
De laatste functie is de ‘dfs.suport.broken.append’-functie. Dit is de oude variant van de ‘dfs.support.append’-functie en wordt eerder toegevoegd als een best-practice.
Hierna wordt Hadoop herstart en is de configuratie ervan klaar.
Op onderstaande schermafbeelding kunt u deze configuratie zien. De configuratie hanteert hetzelfde formaat als in ‘core-site.xml’.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>dfs.webhdfs.enabled</name>
    <value>true</value>
  </property>
  <property>
    <name>dfs.support.append</name>
    <value>true</value>
  </property>
  <property>
    <name>dfs.support.broken.append</name>
    <value>true</value>
  </property>
</configuration>
```

Figuur 24: Schermafbeelding van ‘hdfs-site.xml’

INTEGRATIE MET WAZUH

Nu kan de integratie opgezet worden in Wazuh. Hiervoor worden er aanpassingen gemaakt in ‘ossec.conf’. Deze aanpassingen worden in drie blokken geplaatst: ‘socket’-blok, ‘localfile’-blok en ‘fluent-forward’-blok. In het socket-blok wordt een socket gedefinieerd waarop de Fluentd-forwarder luistert. Deze socket dient als communicatiekanaal tussen de verschillende processen die nodig zijn voor de integratie. Hieronder wordt kort toegelicht wat de verschillende opties doen:

- ‘name’: De naam die aan de socket gegeven wordt.
- ‘location’: Het pad naar waar de socket opgezet wordt.
- ‘mode’: Het protocol dat gebruikt wordt om gegevens te versturen.

In het localfile-blok wordt opgegeven welk bestand naar de Fluentd-forwarder gestuurd moet worden. In dit geval is het het ‘alerts.json’-bestand dat meegegeven wordt, omdat hier alle logs in opgeslagen worden. Hieronder wordt kort toegelicht wat de verschillende opties doen:

- ‘log_format’: Het formaat van de logs.
- ‘location’: De locatie waar het logbestand zich bevindt.
- ‘target’: De socket waar de logs naartoe gestuurd worden.

Het laatste blok is het ‘fluent-forward’-blok. Dit blok dient om de connectie op te zetten met de server waar Fluentd op draait. Hieronder wordt kort toegelicht wat de verschillende opties doen:

- ‘enabled’: Schakelt de functie in.
- ‘tag’: De tag waarmee de gegevens verstuurd worden.
- ‘socket_path’: De locatie waar de socket zich bevindt.
- ‘address’: Het IP-adres van de server waarop Fluentd draait.
- ‘port’: De poort die gebruikt wordt om de verbinding op te zetten.

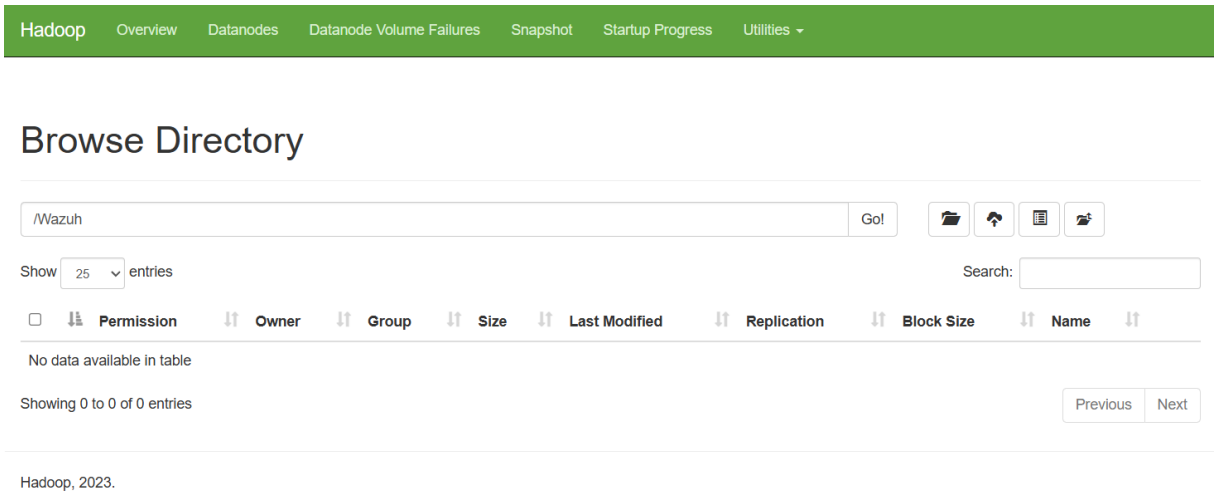
Hierna moet de Wazuh Manager opnieuw opgestart worden en zou de integratie moeten werken.
Op de schermafbeelding hieronder kunt u de integratie zien in het bestand ‘ossec.conf’.

```
<ossec_config>
  <socket>
    <name>fluent_socket</name>
    <location>/var/run/fluent.sock</location>
    <mode>udp</mode>
  </socket>
  <localfile>
    <log_format>json</log_format>
    <location>/var/ossec/logs/alerts/alerts.json</location>
    <target>fluent_socket</target>
  </localfile>
  <fluent-forward>
    <enabled>yes</enabled>
    <tag>wazuh</tag>
    <socket_path>/var/run/fluent.sock</socket_path>
    <address>172.17.0.230</address>
    <port>24224</port>
  </fluent-forward>
</ossec_config>
```

Figuur 25: Schermafbeelding uit ‘ossec.conf’ van integratie

WEBINTERFACE

Via de webinterface kunnen de alerts nu bekeken worden. Deze zouden onder ‘Utilities > Browse the filesystem’ in de map Wazuh zichtbaar moeten zijn. Zoals u op onderstaande schermafbeelding kunt zien, is dit niet het geval. De reden waarom is momenteel nog onbekend.



Figuur 26: Schermafbeelding van webinterface Hadoop.

3.2. SOAR

Nu Wazuh alerts begint te genereren, kan hierop een geautomatiseerde response worden voorzien. Hiervoor wordt gebruikgemaakt van verschillende tools die ervoor zorgen dat alerts automatisch geanalyseerd worden en dat de helpdesk van MPI Oosterlo VZW op de hoogte wordt gebracht.

In dit onderdeel wordt de SOAR-omgeving opgezet. Deze bestaat uit de eerder besproken tools zoals Shuffle, TheHive, Cortex en Microsoft Teams. Er wordt toegelicht hoe deze tools met elkaar samenwerken en hoe ze geïntegreerd zijn met het SIEM-systeem. Ook de implementatie van threat intelligence wordt in dit hoofdstuk behandeld, aangezien dit een centrale rol speelt binnen de werking van Cortex.

3.2.1. Shuffle

De realisatie begint met Shuffle. Dit platform vormt het hart van de SOAR-omgeving en coördineert de volledige automatiseringsworkflow. Shuffle zorgt ervoor dat de helpdesk van MPI Oosterlo zo weinig mogelijk tijd hoeft te besteden aan repetitieve handelingen, waardoor er meer ruimte is voor het effectief oplossen van alerts. In dit onderdeel wordt de installatie van Shuffle besproken, samen met een gedetailleerde uitleg van de workflow die hiervoor werd opgezet.

SETUP

De installatie van Shuffle gebeurt typisch via Docker met Docker Compose, maar is ook mogelijk met Kubernetes. Beide opties zorgen voor een efficiënte installatie en het eenvoudig beheren van de verschillende onderdelen van het platform. Voor het SOC is de keuze gemaakt om te werken met Docker, omdat dit de meest gebruikte manier is.

Shuffle bestaat uit verschillende componenten, zoals de frontend, backend en database, die elk draaien in een aparte Docker-container. Deze containers worden beheerd via Docker Compose, een tool die alle configuratie-informatie bundelt over de verschillende services en hun onderlinge afhankelijkheden. Dankzij Docker Compose kunnen de componenten afzonderlijk worden opgestart en beheerd, wat het onderhoud en toekomstige aanpassingen vereenvoudigt.

Docker Compose zorgt er ook voor dat de omgeving op een gestructureerde manier wordt opgezet. Het zorgt ervoor dat alle componenten met elkaar verbonden zijn en correct geconfigureerd zijn. Hierdoor kan Shuffle snel opgezet worden en kunnen gebruikers er direct mee aan de slag gaan. Dit betekent echter niet dat er zich geen problemen kunnen voordoen.

a) Installatie

Vooraleer er aan de installatie begonnen kan worden, moeten eerst de nodige tools geïnstalleerd zijn. De installatie van Shuffle gebeurt via Docker met Docker Compose, wat betekent dat dit geïnstalleerd moet worden. Dit kan gebeuren met behulp van de documentatie van Docker.

Docker heeft enkele tools nodig vooraleer het geïnstalleerd kan worden. Net als bij Wazuh heeft het ook een GPG-sleutel om de installatie te controleren. Deze wordt in de map '/etc/apt/keyrings' geplaatst en de map krijgt leesrechten voor alle gebruikers. Hierna kan het Docker repository toegevoegd worden aan de Apt-bronnen van Ubuntu. Hierdoor kan Docker geïnstalleerd en bijgewerkt worden met het apt-commando.

Er kan nu verder worden gegaan met het installeren van Docker. Dit bevat verschillende Docker-componenten, zoals Docker Compose en de Docker CLI. Deze componenten zorgen ervoor dat alle functionaliteiten van Docker mogelijk zijn. Dit kan getest worden door de standaardcontainer van Docker te runnen genaamd 'hello-world'. Hieronder kunt u zien wat dit weergeeft.

```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:dd01f97f252193ae3210da231b1dca0cffab4aadb3566692d6730bf93f123a48
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Figuur 27: Schermafbeelding Docker container 'hello-world'

Hierna kan de installatie van Shuffle zelf beginnen. Shuffle heeft een Github repository waar alle nodige bestanden zich in bevinden. Deze repository wordt gekopieerd naar de VM waarop Shuffle geïnstalleerd wordt. In de repository staat een docker-compose.yml-bestand. In dit bestand staan alle configuraties van de verschillende componenten uitgeschreven. Door dit bestand uit te voeren met Docker Compose, gaan alle componenten correct opgezet worden en kan er begonnen worden met Shuffle. Onder normale omstandigheden zou er niks aangepast moeten worden aan dit document. Hieronder wordt er wel per component toegelicht wat het doet en hoe het geconfigureerd is.

Frontend
De frontend van Shuffle is de webinterface waarmee gebruikers op een visuele manier kunnen werken met het platform. Via deze interface kunnen gebruikers workflows opbouwen, beheren en uitvoeren. De frontend vormt de visuele laag boven op de achterliggende API's die Shuffle aansturen. Hoewel de logica en uitvoering achter de schermen plaatsvinden, maakt de frontend het mogelijk om dit allemaal op een gebruiksvriendelijke en overzichtelijke manier te bedienen.

Op de onderstaande schermafbeelding kunt u de configuratie van de frontend zien in de docker-compose.yml. Hieronder wordt kort toegelicht waarvoor de verschillende opties dienen:

- 'image': De Docker image waarop de frontend gebaseerd is, Shuffle heeft zijn eigen frontend image die hiervoor gebruikt wordt.
- 'container_name': De naam die gegeven wordt aan de Docker container om de container makkelijker te herkennen.
- 'hostname': De hostnaam die gegeven wordt aan de Docker container voor interne communicatie.
- 'ports': De poorten die gebruikt worden voor HTTP en HTTPS. Deze kunnen zelf gekozen worden door het aan te passen in het .env-bestand, maar zijn standaard 80 en 443.
- 'networks': Het interne netwerk waarmee de frontend verbonden is en communiceert met de andere componenten.
- 'environment': Omgevingsvariabele waar de hostnaam van de backend ingegeven wordt. Zo weet de frontend hoe het met de backend kan verbinden en wordt dit toegevoegd aan het .env-bestand.
- 'restart': Docker herstart de container automatisch als het stopt, tenzij dit door de gebruiker gebeurt.
- 'depends_on': Welk component eerst geconfigureerd moet zijn vooraleer de frontend geconfigureerd kan worden.

```
frontend:
  image: ghcr.io/shuffle/shuffle-frontend:latest
  container_name: shuffle-frontend
  hostname: shuffle-frontend
  ports:
    - "${FRONTEND_PORT}:80"
    - "${FRONTEND_PORT_HTTPS}:443"
  networks:
    - shuffle
  environment:
    - BACKEND_HOSTNAME=${BACKEND_HOSTNAME}
  restart: unless-stopped
  depends_on:
    - backend
```

Figuur 28: Schermafbeelding van frontendconfiguratie in docker-compose.yml

Backend

De backend van Shuffle heeft als functie het centrale component te zijn in het platform. Het is opgebouwd als een REST API die inkomende verzoeken van de frontend of externe systemen afhandelt. Het stuurt hierbij de verschillende interne componenten aan zoals de database, workflow engine en Orborus. Wanneer er een HTTP-verzoek binnenkomt, voert de backend de volgende stappen uit:

1) Authenticatie en autorisatie worden eerst gecontroleerd. Er wordt zo bepaald of het verzoek geldig is en of de gebruiker toegang heeft tot de gevraagde actie.
2) Daarna wordt het verzoek verwerkt**.** Afhankelijk van het type verzoek kunnen er bijvoorbeeld workflows worden gestart, logs worden opgehaald, of gegevens worden aangepast.
3) Indien nodig wordt een taak aangemaakt en gestart via Orborus, dat verantwoordelijk is voor het uitvoeren van taken binnen workflows.
4) Tot slot stuurt de backend een HTTP-statuscode en eventueel aanvullende gegevens terug naar de gebruiker om aan te geven wat het resultaat van het verzoek is.

Op de onderstaande schermafbeelding kunt u de configuratie van de backend zien in de docker-compose.yml. Hieronder wordt kort toegelicht waarvoor de verschillende opties dienen:

- ‘image’: De Docker image waarop de backend gebaseerd is, Shuffle heeft zijn eigen backend image die hiervoor gebruikt wordt.
- ‘container_name’: De naam die gegeven wordt aan de Docker container om de container makkelijk te herkennen.
- ‘hostname’: De hostnaam die gegeven wordt aan de Docker container voor interne communicatie. De naam komt uit het .env-bestand.
- ‘ports’: De poort die gebruikt wordt om met de backend te verbinden. Dit kan zelf gekozen worden door het aan te passen in het .env-bestand, maar is standaard 5001.
- ‘networks’: Het interne netwerk waarmee de backend verbonden is en communiceert met de andere componenten.
- ‘volumes’: Maakt mappen van de hostmachine beschikbaar binnen de container. De container kan hierdoor bestanden lezen of opslaan op een plek buiten de container.
- ‘env_file’: Laadt de omgevingsvariabelen, die in het .env-bestand zitten, in.
- ‘environment’: Omgevingsvariabelen waar de paden van de shuffle-apps en shuffle-files ingegeven worden. Zo weet de backend waar deze te vinden zijn.
- ‘restart’: Docker herstart de container automatisch als het stopt, tenzij dit door de gebruiker gebeurt.

```yaml
backend:
  image: ghcr.io/shuffle/shuffle-backend:latest
  container_name: shuffle-backend
  hostname: ${BACKEND_HOSTNAME}
  # Here for debugging:
  ports:
    - "${BACKEND_PORT}:5001"
  networks:
    - shuffle
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
    - ${SHUFFLE_APP_HOTLOAD_LOCATION}:/shuffle-apps:z
    - ${SHUFFLE_FILE_LOCATION}:/shuffle-files:z
  env_file: .env
  environment:
    #- DOCKER_HOST=tcp://docker-socket-proxy:2375
    - SHUFFLE_APP_HOTLOAD_FOLDER=/shuffle-apps
    - SHUFFLE_FILE_LOCATION=/shuffle-files
  restart: unless-stopped
```

Figuur 29: Schermafbeelding van backendconfiguratie in docker-compose.yml

Orborus
Orborus is het uitvoerende component binnen Shuffle en heeft als taak het afhandelen van taken binnen workflows. Het ontvangt instructies van de backend over hoe de workflow uitgevoerd moet worden. Het volgt deze instructies stap voor stap op om het gewenste resultaat te bieden aan de gebruiker.
Orborus draait op een aparte container om schaalbaarheid en betrouwbaarheid te garanderen. Het is hierdoor mogelijk om meerdere workflows tegelijk in parallel uit te voeren, wat extra functionaliteiten biedt voor de gebruiker.

Op de onderstaande schermafbeelding kunt u de configuratie van Orborus zien in 'docker-compose.yml'. Hieronder wordt kort toegelicht waarvoor de verschillende opties dienen:

- 'image': De Docker image waarop Orborus gebaseerd is, Shuffle heeft zijn eigen Orborus image die hiervoor gebruikt wordt.
- 'container_name': De naam die gegeven wordt aan de Docker container om de container makkelijk te herkennen.
- 'hostname': De hostnaam die gegeven wordt aan de Docker container voor interne communicatie..
- 'networks': Het interne netwerk waarmee de backend verbonden is en communiceert met de andere componenten.
- 'volumes': Maakt mappen van de hostmachine beschikbaar binnen de container. De container kan hierdoor bestanden lezen of opslaan op een plek buiten de container.
- 'environment': Hierin staan de omgevingsvariabelen die gebruikt worden in het .env-bestand. Omdat dit teveel opties zijn, worden ze niet in dit document uitgelegd om de structuur te bewaren.
- 'env_file': Laadt de omgevingsvariabelen, die in het .env-bestand zitten, in.
- 'restart': Docker herstart de container automatisch als het stopt, tenzij dit door de gebruiker gebeurt.
- 'security_opt': Bepaalt welke systeemaanroepen uitgevoerd mogen worden. Orborus wordt hier niet in beperkt en mag ze allemaal uitvoeren.

```yaml
orborus:
  image: ghcr.io/shuffle/shuffle-orborus:latest
  container_name: shuffle-orborus
  hostname: shuffle-orborus
  networks:
    - shuffle
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
  environment:
    - SHUFFLE_APP_SDK_TIMEOUT=300
    - SHUFFLE_ORBORUS_EXECUTION_CONCURRENCY=7 # The amount of concurrent executions Orborus can handle.
    #- DOCKER_HOST=tcp://docker-socket-proxy:2375
    - ENVIRONMENT_NAME=Shuffle
    - ORG_ID=Shuffle
    - BASE_URL=http://${OUTER_HOSTNAME}:5001
    - DOCKER_API_VERSION=1.40
    - HTTP_PROXY=${HTTP_PROXY}
    - HTTPS_PROXY=${HTTPS_PROXY}
    - SHUFFLE_PASS_WORKER_PROXY=${SHUFFLE_PASS_WORKER_PROXY}
    - SHUFFLE_PASS_APP_PROXY=${SHUFFLE_PASS_APP_PROXY}
    - SHUFFLE_STATS_DISABLED=true
    - SHUFFLE_LOGS_DISABLED=true
    - SHUFFLE_SWARM_CONFIG=run
    - SHUFFLE_WORKER_IMAGE=ghcr.io/shuffle/shuffle-worker:latest
  env_file: .env
  restart: unless-stopped
  security_opt:
    - seccomp:unconfined
```

Figuur 30: Schermafbeelding van Orborusconfiguratie in docker-compose.yml

Opensearch
Opensearch is de centrale database binnen Shuffle waar gegevens uit workflows opgeslagen, doorzocht en geanalyseerd worden. Deze gegevens kunnen van alles zijn zoals logs van uitgevoerde acties, foutmeldingen en status van workflows. Door het gebruik van Opensearch kunnen gebruikers makkelijk belangrijke data ophalen en terugvinden wat er tijdens de workflows gebeurd is.
Opensearch draait net als de andere componenten in een aparte container. Dit maakt het makkelijker om troubleshooting uit te voeren en geeft extra schaalbaarheid en flexibiliteit voor de gebruiker.

Op de onderstaande schermafbeelding kunt u de configuratie van Opensearch zien in 'docker-compose.yml'. Hieronder licht ik kort toe waarvoor de verschillende opties dienen:

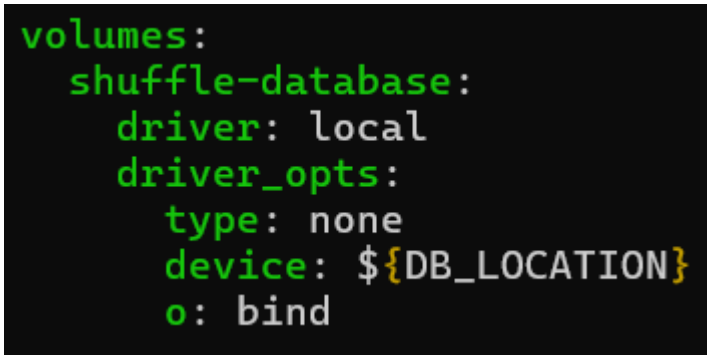
- 'image': De Docker image waarop Opensearch gebaseerd is, Shuffle heeft zijn eigen Opensearch image die hiervoor gebruikt wordt.
- 'hostname': De hostnaam die gegeven wordt aan de Docker container voor interne communicatie.
- 'container_name': De naam die gegeven wordt aan de Docker container om de container makkelijk te herkennen.
- 'environment': Hierin staan de omgevingsvariabelen die gebruikt worden in het .env-bestand. Omdat dit te veel opties zijn, worden ze niet in dit document uitgelegd om de structuur te bewaren.
- 'ulimits:memlock': Stelt in hoeveel RAM-geheugen gelocked mag worden. Dit houdt in dat het niet naar het wisselgeheugen wordt verplaatst. Door dit op -1 te zetten, betekent het dat dit geen limiet heeft. 'soft' geeft de standaardlimiet weer en 'hard' de maximumlimiet.
- 'ulimits:nofile': Stelt in hoeveel open bestanden en sockets Opensearch mag hebben. 'soft' betekent opnieuw de standaardlimiet en 'hard' de maximumlimiet.
- 'volumes': Maakt mappen van de hostmachine beschikbaar binnen de container. De container kan hierdoor bestanden lezen of opslaan op een plek buiten de container.
- 'ports': De poorten die gebruikt worden voor Opensearch.
- 'networks': Het interne netwerk waarmee de backend verbonden is en communiceert met de andere componenten.
- 'restart': Docker herstart de container automatisch als het stopt, tenzij dit door de gebruiker gebeurt.

```
opensearch:
  image: opensearchproject/opensearch:2.19.1
  hostname: shuffle-opensearch
  container_name: shuffle-opensearch
  environment:
    - "OPENSEARCH_JAVA_OPTS=-Xms2048m -Xmx2048m" # minimum and maximum Java heap size, recommend setting both to 50% of system RAM
    - bootstrap.memory_lock=true
    - DISABLE_PERFORMANCE_ANALYZER_AGENT_CLI=true
    - cluster.initial_master_nodes=shuffle-opensearch
    - cluster.routing.allocation.disk.threshold_enabled=false
    - cluster.name=shuffle-cluster
    - node.name=shuffle-opensearch
    - node.store.allow_mmap=false
    - discovery.seed_hosts=shuffle-opensearch
    - OPENSEARCH_INITIAL_ADMIN_PASSWORD=${SHUFFLE_OPENSEARCH_PASSWORD}
  ulimits:
    memlock:
      soft: -1
      hard: -1
    nofile:
      soft: 65536
      hard: 65536
  volumes:
    - shuffle-database:/usr/share/opensearch/data:z
  ports:
    - 9200:9200
  networks:
    - shuffle
  restart: unless-stopped
```

Figuur 31: Schermafbeelding van Opensearchconfiguratie in 'docker-compose.yml'

Volumes
Volumes worden gebruikt om data persistent op te slaan. Dit betekent dat deze data ook als de container stopt, blijft bestaan. Voor gevoelige data die niet verloren mag gaan, is dit de perfecte opslagplaats. Voor Shuffle is er het volume 'Shuffle-database' aangemaakt. Deze kan aangesproken worden in andere delen van het docker-compose bestand om bestanden in op te slaan.
Op onderstaande afbeelding kunt u zien hoe dit geconfigureerd is in 'docker-compose.yml'. Hieronder wordt kort toegelicht waarvoor de verschillende opties dienen:

- 'driver': Hier wordt aangegeven waar de bestanden opgeslagen worden. In dit geval is dit lokaal op het hostsysteem.
- 'driver_opts': Extra opties die toegevoegd kunnen worden aan het volume.
- 'driver_opts.type': Geeft aan welk specifiek type bestandsysteem wordt gebruikt.
- 'driver_opts.device': Geeft aan welke map op het hostsysteem gebruikt moet worden om de bestanden in op te slaan.
- 'driver_opts.o': Geeft aan dat het volume een bind-mount is. Dit wil zeggen dat de map op het hostsysteem gekoppeld wordt aan de containers.



Figuur 32: Schermafbeelding van volumeconfiguratie in 'docker-compose.yml'

Networks
Ten slotte is er nog het netwerk dat geconfigureerd moet worden. Dit zet een virtueel netwerk op dat de verschillende componenten gebruiken om met elkaar te verbinden. Het netwerk krijgt de naam Shuffle en dit wordt ingegeven bij de componenten zodat ze weten met welk netwerk ze verbinden.

Dit is ook het deel dat voor de meeste problemen heeft gezorgd bij het opzetten van Shuffle in het SOC. De installatie van Shuffle wilde niet succesvol lukken omdat er ergens een error zat die niet gevonden werd. Na meerdere dagen afspeuren van alle verschillende redenen, is er uiteindelijk naar de routes gekeken die Docker had opgesteld. Deze routes bepalen hoe netwerkverkeer zijn eindbestemming bereikt door als bestemming een IP-subnet op te geven. Enkele van deze routes gebruikten hetzelfde IP-subnet die ook in het netwerk van het MPI Oosterlo gebruikt werden. Daarom werd telkens de verbinding verbroken met de server, omdat de route naar het VLAN overschreven werd. Door het IP-subnet aan te passen, was het probleem opgelost.

Op onderstaande schermafbeelding kunt u de configuratie zien van het netwerk in 'docker-compose.yml'. Hieronder wordt kort toegelicht waarvoor de verschillende opties dienen:

- 'driver': Het netwerkmodel dat gebruikt wordt in Docker. In dit geval is het een 'bridge' waarbij er een virtueel netwerk wordt opgesteld dat met de buitenwereld kan communiceren via de host.
- 'ipam.config.subnet': Bepaalt het IP-subnet dat gebruikt wordt voor het virtueel netwerk.

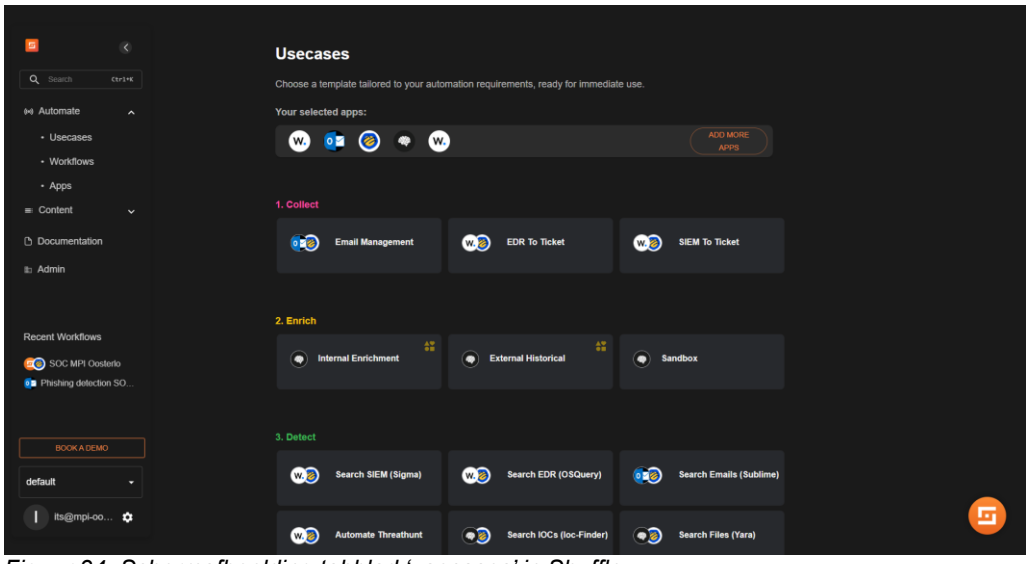
Figuur 33: Schermafbeelding netwerkconfiguratie in 'docker-compose.yml'

DASHBOARD

Nu Shuffle volledig geïnstalleerd is, kan de workflow opgesteld worden. Hiervoor is er enkel een account nodig, wat snel geregistreerd kan worden bij Shuffle zelf. Na een succesvolle aanmelding bij Shuffle komt de gebruiker op het dashboard, dat op onderstaande schermafbeelding zichtbaar is.

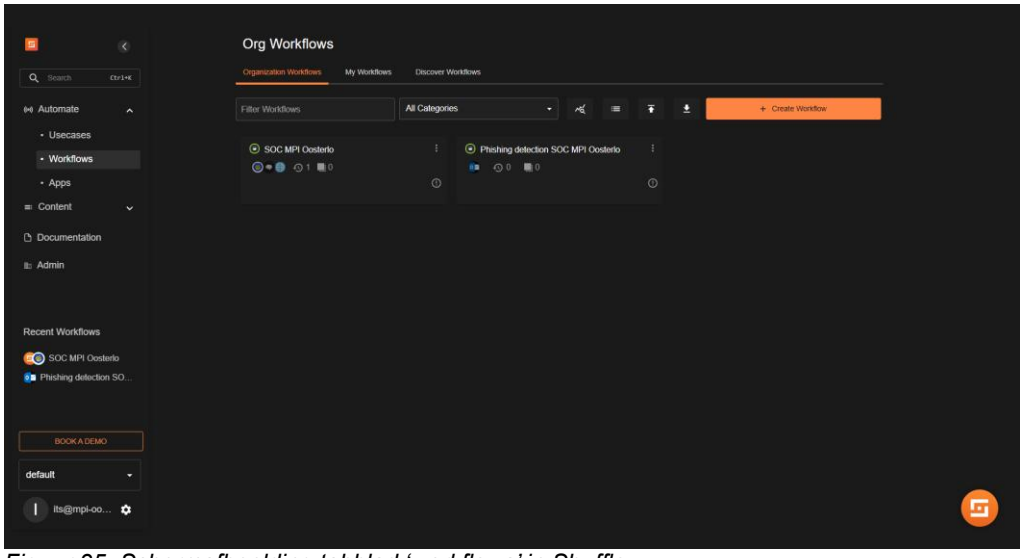
Standaard komt de gebruiker uit op het tabblad ‘usecases’. Dit geeft enkele voorbeelden van mogelijke workflows die opgesteld kunnen worden. Links staat er een uitklapmenu waarin er nog meer opties zijn. De belangrijkste hiervan is het deel ‘Automate’. Dit wordt het meeste in het SOC gebruikt. Er zijn drie tabbladen te vinden, die elk een deel vormen van het automatische proces van Shuffle.

Het eerste tabblad zijn de ‘usecases’. Dit zijn voorbeelden van workflows die al door Shuffle of andere gebruikers zijn opgezet. Het is hier mogelijk te filteren op gebruikte tools en hier een workflow mee op te zetten. Deze workflows zijn onderverdeeld op de taken die ze aanbieden, zoals bijvoorbeeld het verzamelen van data of het detecteren van bedreigingen.



Figuur 34: Schermafbeelding tabblad ‘usecases’ in Shuffle

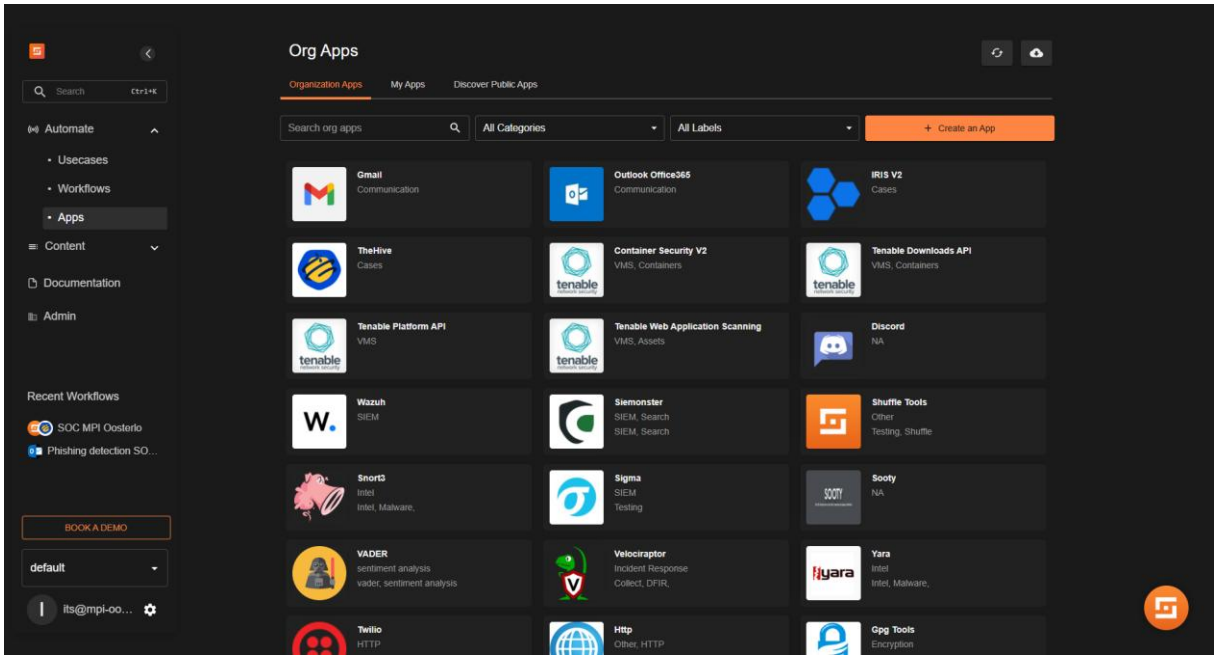
Het tweede zijn de ‘workflows’. Hierin staan de workflows die de gebruiker zelf aanmaakt. Op onderstaande schermafbeelding kunt u de workflows zien die voor het MPI Oosterlo gemaakt zijn. Deze worden in het volgende deel verder besproken.



Figuur 35: Schermafbeelding tabblad ‘workflows’ in Shuffle

Het derde en laatste tabblad is ‘apps’. Shuffle heeft verschillende tools waarmee een automatische workflow opgezet kan worden. Van deze tools wordt een app gemaakt die alle nodige en optionele velden bevat. Zo krijgt de gebruiker een visuele voorstelling van hoe een API-verzoek naar de tool eruit zou zien. Op de schermafbeelding hieronder kunt u de verschillende apps zien die Shuffle standaard aanbiedt. Deze zijn vaak door mensen in de community gemaakt en officieel toegevoegd door Shuffle.

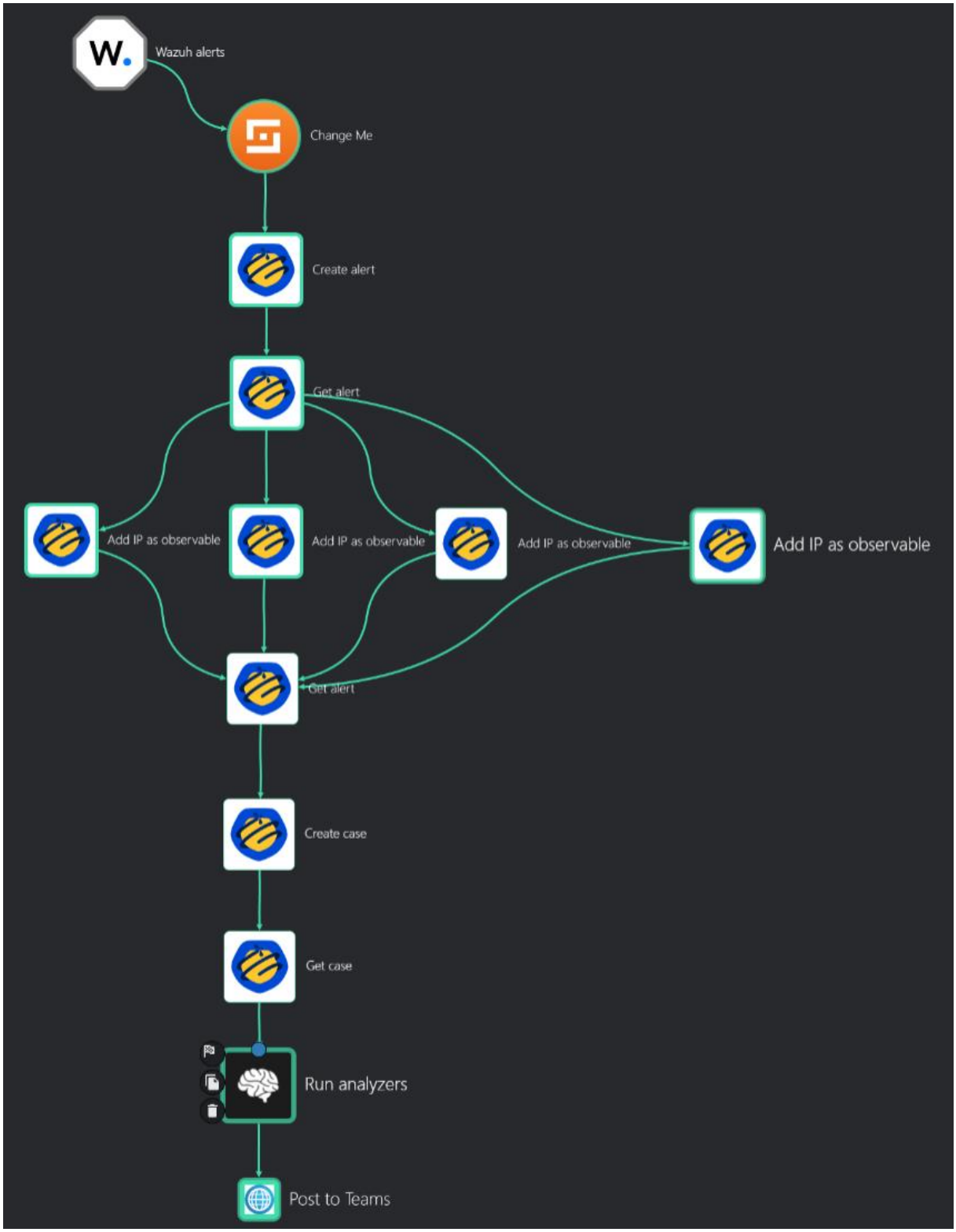
Mocht er toch de nood zijn om bij een applicatie een eigen app te maken, is dit ook mogelijk. Shuffle apps zijn in Python geschreven met OpenAPI or Swagger. De gebruiker kan zelf de code schrijven of de App laten maken door gebruik te maken van de GUI.



Figuur 36: Schermafbeelding tabblad ‘apps’ in Shuffle

WORKFLOW 'SOC MPI OOSTERLO'

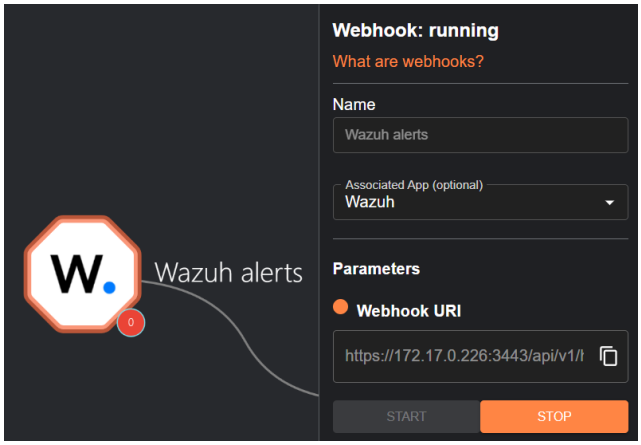
Op onderstaande schermafbeelding kunt u de volledige workflow zien. Dit vormt de centrale component die alerts van Wazuh ophaalt, naar TheHive stuurt om er een case van te maken, de case te analyseren en een melding te maken in Teams. In de volgende delen wordt elk deel van de workflow uitgelegd en welke functie dit heeft in het SOC.



Figuur 37: Schermafbeelding workflow 'SOC MPI Oosterlo'

Wazuh alerts

Om de workflow in gang te zetten, moet er een trigger zijn die de start maakt. Zoals daarstraks al besproken zijn er hiervoor vier opties: een webhook, een schedule, een subflow en user input. Omdat Wazuh op onregelmatige momenten een alert genereerd en deze direct onderzocht moet worden, is de webhook de beste keuze voor de stageopdracht. Deze wordt aangesproken door Wazuh wanneer er een alert aan bepaalde voorwaarden voldoet en zet dan de workflow in gang. De webhook is een URI die door Shuffle gegeneerd wordt om vanuit verschillende tools met Shuffle te verbinden. Op de schermafbeelding hieronder ziet u hoe deze webhook opgezet is in Shuffle.



Figuur 38: Schermafbeelding webhook Wazuh in Shuffle

Om deze integratie te laten werken, zijn er nog twee bestanden nodig die zich bevinden op de Github van Wazuh. Dit zijn 'custom-shuffle' en 'custom-shuffle.py'. Er moet niet veel aan deze bestanden veranderd worden, enkel in 'custom-shuffle.py' is er een kleine aanpassing nodig. Hier wordt 'verify=False' toegevoegd aan de 'send_msg' functie, omdat Shuffle met HTTPS werkt via self-signed certificaten. Deze kunnen niet gecontroleerd worden door Wazuh en zou de integratie blokkeren. Door de controle af te zetten, werkt de integratie wel. Op de schermafbeelding hieronder ziet u de aanpassing in de functie 'send_msg' in 'custom-shuffle.py'. De volledige bestanden zijn te groot om in dit document te laten zien, maar kunt u vinden in Bijlage 3.

```
def send_msg(msg, url):
    debug("# In send msg")
    headers = {'content-type': 'application/json', 'Accept-Charset': 'UTF-8'}
    res = requests.post(url, data=msg, headers=headers, verify=False)
    debug("# After send msg: %s" % res)
```

Figuur 39: Schermafbeelding functie 'send_msg' uit 'custom_shuffle.py'

Nu alles klaar staat om alerts naar Shuffle te sturen, moet enkel 'ossec.conf' nog aangepast worden. Wazuh voorziet hiervoor een blok genaamd 'integration' in het 'ossec.conf'-bestand. Deze kunt u zien op de schermafbeelding hieronder. Het blok 'integration' staat online in de documentatie van Wazuh en kan naar 'ossec.conf' gekopieerd worden. Nu kan de integratie opgezet worden door enkele opties aan te passen. Hieronder wordt kort toegelicht waarvoor de verschillende opties dienen:

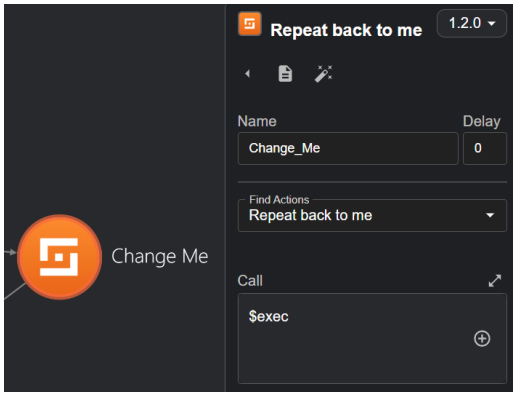
- 'name': De naam die gegeven wordt aan de integratie, moet overeenkomen met de naam van de integratiebestanden.
- 'hook_url': De webhook waarmee de integratie opgezet wordt.
- 'level': Het meldingsniveau van de alerts die doorgestuurd worden met de webhook.
- 'alert_format': In welk formaat de alerts doorgestuurd worden.

```
<integration>
  <name>custom-shuffle</name>
  <hook_url>https://172.17.0.226:3443/api/v1/hooks/webhook_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx </hook_url>
  <level>12</level>
  <alert_format>json</alert_format>
</integration>
```

Figuur 40: Schermafbeelding blok 'integration' uit 'ossec.conf'

Repeat back to me

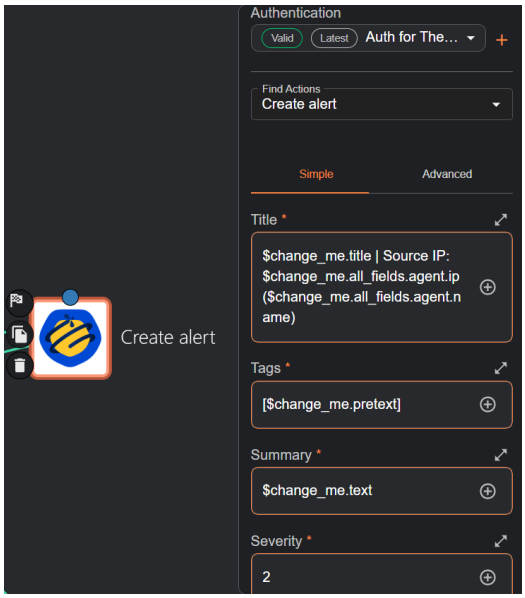
Nu er alerts binnenkomen op Shuffle, kan er met deze gegevens aan de slag gegaan worden. Met Shuffle is het niet mogelijk om gegevens, die uit een webhook komen, later terug aan te spreken. Als er dus iets moet gebeuren met de alerts, zoals bijvoorbeeld een observable opzetten, kan dit niet meer opgeroepen worden. Daarom is de functie 'Repeat back to me' toegevoegd aan de workflow. Dit is een functie die bij in de app Shuffle-tools zit en die standaard toegevoegd wordt aan een nieuwe workflow.
De functie 'Repeat back to me' herhaalt de laatste gegevens die het binnenkrijgt. In dit geval gaat het de data zijn die binnenkomt met de webhook. Deze gegevens zitten in de 'runtime argument' wat aangeroepen wordt met '$exec'. Op de schermafbeelding hieronder kunt u de app zien in Shuffle.



Figuur 41: Schermafbeelding 'Repeat back to me' in Shuffle

Create alert

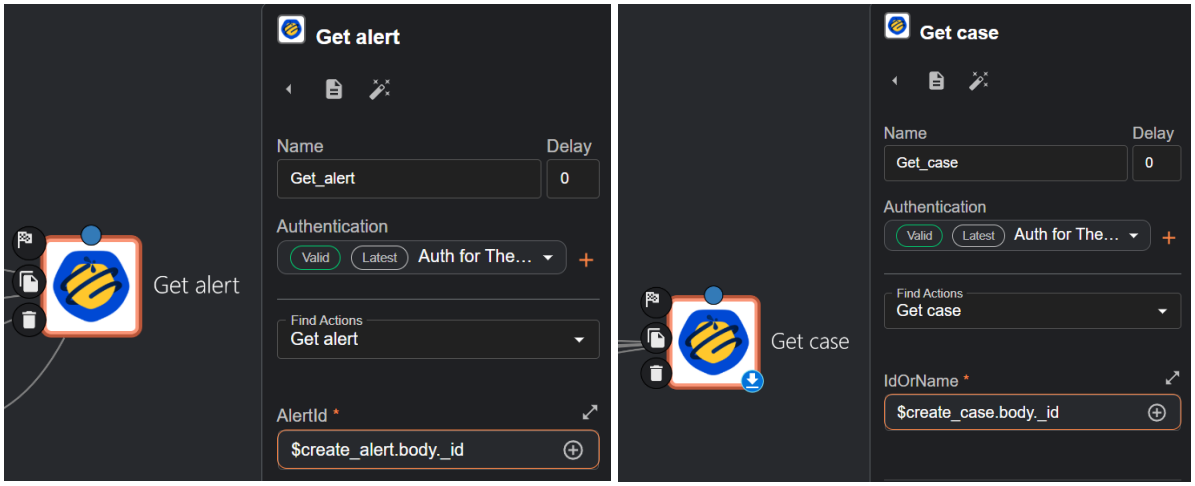
Nu de gegevens van de Wazuh alert opgehaald kunnen worden, kan er een alert gemaakt worden in TheHive. Dit gebeurt via de app 'TheHive', die met een API-call een nieuwe alert kan aanmaken. Hiervoor moet er eerst een verbinding worden opgezet met de server van TheHive. Shuffle heeft hiervoor een veld genaamd 'authentication'. Hierin worden de benodigde zaken getoond voor een connectie op te stellen met de service. Voor TheHive is dit de URL van de server en een API-key. De API-key is afkomstig van een gebruiker in een organisatie in TheHive, dit komt later aan bod. Shuffle slaat deze verbinding op zodat het gebruikt kan worden wanneer de app van TheHive opnieuw gebruikt wordt.
Nu kan de alert voor TheHive gemaakt worden. Hiervoor zijn vier zaken nodig: een titel, een tag, een samenvatting en de ernst. Om deze zaken in te vullen, worden gegevens vanuit de Wazuh alert gebruikt. Zoals u in de schermafbeelding hieronder kunt zien, worden variabelen gebruikt die direct uit de vorige app 'Repeat back to me' komen. Zo worden de gegevens dynamisch aangepast op basis van de binnenkomende alerts. Enkel de ernst staat op een vaste waarde, omdat Shuffle hier syntax-errors op geeft.



Figuur 42: Schermafbeelding 'Create alert' uit Shuffle

Get alert/case

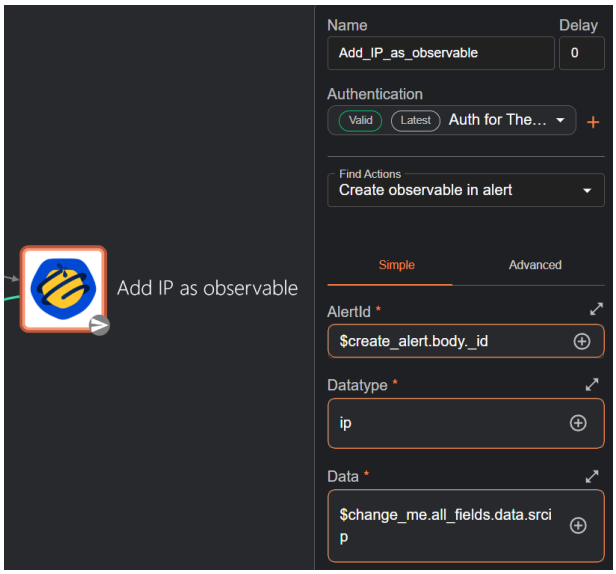
Om troubleshooting in de workflow makkelijker te maken, is het handig om te zien wat de verschillende acties genereren. Daarvoor zijn de ‘Get alert’ of ‘Get case’ functies de perfecte oplossing. Hiermee kan de alert of case die net aangemaakt is, opgehaald en onderzocht worden. Zo kunnen fouten in de workflow snel opgespoord worden en kan erop gereageerd worden.
Op de schermafbeelding hieronder ziet u de ‘Get alert’ functie. Deze werkt door de alert-ID op te vragen. De ‘Get case’ functie hanteert hetzelfde principe en kunt u op de schermafbeelding ernaast zien.



Figuur 43 & 44: Schermafbeeldingen van ‘Get alert’ en ‘Get case’ uit Shuffle

Add IP as observable

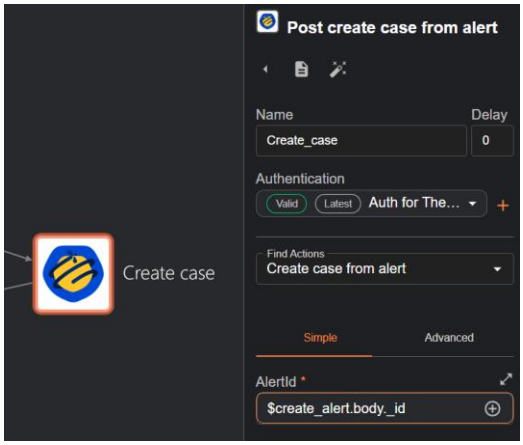
Nu de alerts aangemaakt worden in TheHive, is het tijd om hier data aan toe te voegen die onderzocht kan worden. Hiervoor wordt er gebruik gemaakt van observables. Deze worden in TheHive toegevoegd in een apart tabblad binnen de alert, maar hier kom ik straks op terug.
Om een observable toe te voegen, wordt de funtie ‘Create observable in alert’ gebruikt. Hiervoor zijn er enkele zaken nodig: een alertID, het datatype en de data. Deze worden opnieuw ingevuld met variabelen, zoals met het maken van een alert. Enkel het datatype is een statisch gegeven.
In het SOC wordt er een IP-adres toegevoegd als observable. Omdat hier het probleem opkomt van Wazuh die verschillende indexen hieraan geeft, is er een andere oplossing voor gezocht. Daarom zijn er drie verschillende manieren gebruikt om een IP-adres als observable toe te voegen. Op deze manier is er zekerheid dat het IP-adres toegevoegd wordt.
Op de schermafbeelding hieronder ziet u de ‘Create observable in alert’ functie. Deze werd gebruikt voor alerts uit Linux-endpoints. De andere apps voor Windows kunt u vinden in Bijlage 4.



Figuur 45: Schermafbeelding ‘Create observable in alert’ uit Shuffle

Create case

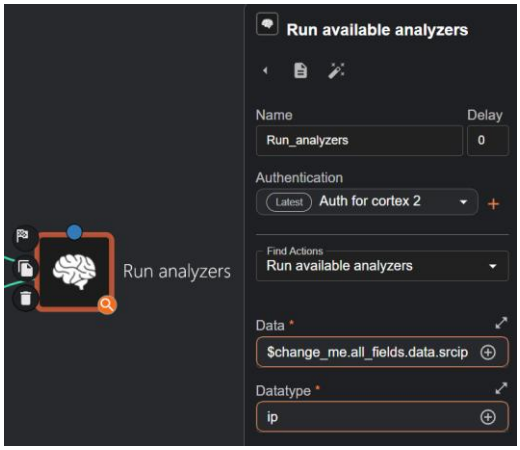
Nu er aan de alert een observable toegevoegd is, kan er een case gemaakt worden van de alert. Hiervoor wordt de functie ‘Create case from alert’ gebruikt. Om dit te laten werken, is enkel het alert-ID nodig. Hiermee kan TheHive zijn eigen alert ophalen en gebruiken om een case aan te maken. Aan deze case worden dan ook de observables uit de vorige stap toegevoegd.
Cases zijn handig omdat ze meer functionaliteiten bieden voor de gebruiker en organisatie. In de uitleg over TheHive wordt er hierop teruggekomen.
Op de schermafbeelding hieronder ziet u de ‘Create case from alert’ functie.



Figuur 46: Schermafbeelding ‘create case from alert’ uit Shuffle

Run analyzers

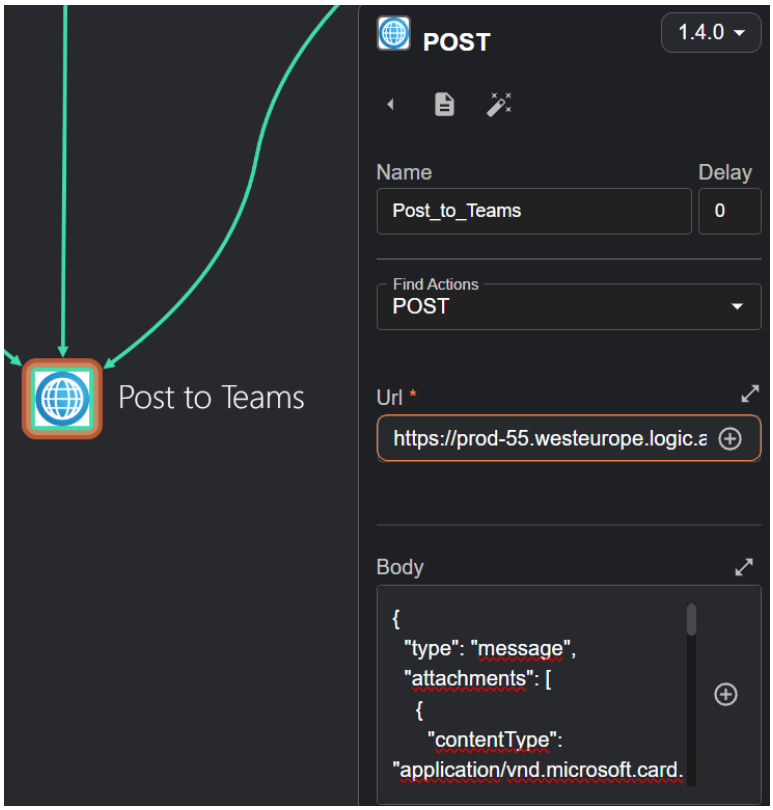
Zodra er een case met observables is aangemaakt, kunnen deze observables geanalyseerd worden. Hiervoor wordt de app van ‘Cortex’ gebruikt met de functie ‘Run available analyzers’. Hiermee gaat Cortex alle mogelijke analyzers oproepen om onderzoek te doen naar het IP-adres.
Om deze analyse mogelijk te maken, moet er eerst een verbinding zijn met Cortex. Hiervoor is, net zoals met TheHive, de URL van de server en de API-key van een gebruiker in de organisatie nodig. Hoe dit gebeurt komt later aan bod in het deel van Cortex.
Om de analyzers te laten werken, moeten ze eerst weten wat er geanalyseerd moet worden. Shuffle heeft hiervoor twee zaken nodig: de data en het datatype. Deze zijn zichtbaar op onderstaande schermafbeelding. Normaal zou hiervoor de observable gebruikt kunnen worden, deze heeft namelijk beide al gedefinieerd, maar in het SOC komen de observables niet bij in de case op Shuffle te staan. Het is dus niet mogelijk om deze op te halen met een variabele. Daarom is hier dezelfde oplossing toegepast die ook gebruikt is bij het toevoegen van de observables in de alert. Er zijn drie apps die de IP-adressen ophalen van alerts uit Linux- en Windowsendpoints en in de data steken. Het datatype staat statisch op IP-adres ingesteld.
De analyzers in Cortex kunnen nu deze informatie gebruiken om hun analyse uit te voeren. De resultaten hiervan zijn in Cortex zichtbaar, maar dit komt later aan bod in het deel van Cortex.



Figuur 47: Schermafbeelding ‘Run available analyzers’ uit Shuffle

Post to Teams

Tenslotte is er nog de melding die aangemaakt wordt op Teams. Een SOC kan natuurlijk niet volledig automatisch zijn als er nergens een melding is van een nieuwe alert. Hiervoor zijn verschillende opties mogelijk zoals Outlook, Teams en Discord. Zoals eerder al vermeld is hier de beslissing genomen om Teams te gebruiken.
Om een integratie met Teams te maken, moet er eerst een connectie zijn. Shuffle biedt zelf geen apps aan om die connectie te maken, maar gelukkig Teams wel. Hier kan er een webhook opgesteld worden waarmee Shuffle kan verbinden. Deze wordt aangemaakt in Teams zelf en is gelinkt aan een chat, maar dit komt later aan bod in het deel van Teams.
Om de webhook aan te kunnen spreken, wordt er gebruik gemaakt van een POST-aanvraag. Dit is een aanvraag die dient om nieuwe informatie, of in dit geval een nieuw bericht, op te sturen naar een URL. Shuffle vraagt om een URL (de webhook) en een body toe te voegen. In de body wordt het bericht geplaatst dat naar Teams verstuurd moet worden. Dit moet in de juiste structuur zijn dat bepaald wordt door Microsoft en te vinden is in het Power Automate Portaal.
Op onderstaande schermafbeelding kunt u zien hoe de app er in Shuffle uitziet. Het bericht bevat de titel van de alert, het IP-adres dat de alert gegenereerd heeft en de hostname hiervan. De volledige body kunt u vinden in Bijlage 5.



Figuur 45: Schermafbeelding 'Post to teams' uit Shuffle

3.2.2. TheHive

De workflow van Shuffle kan niet werken zonder dat er een instantie van TheHive is opgezet. In dit deel wordt de installatie van TheHive uitgelegd en hoe de webinterface werkt.

SETUP

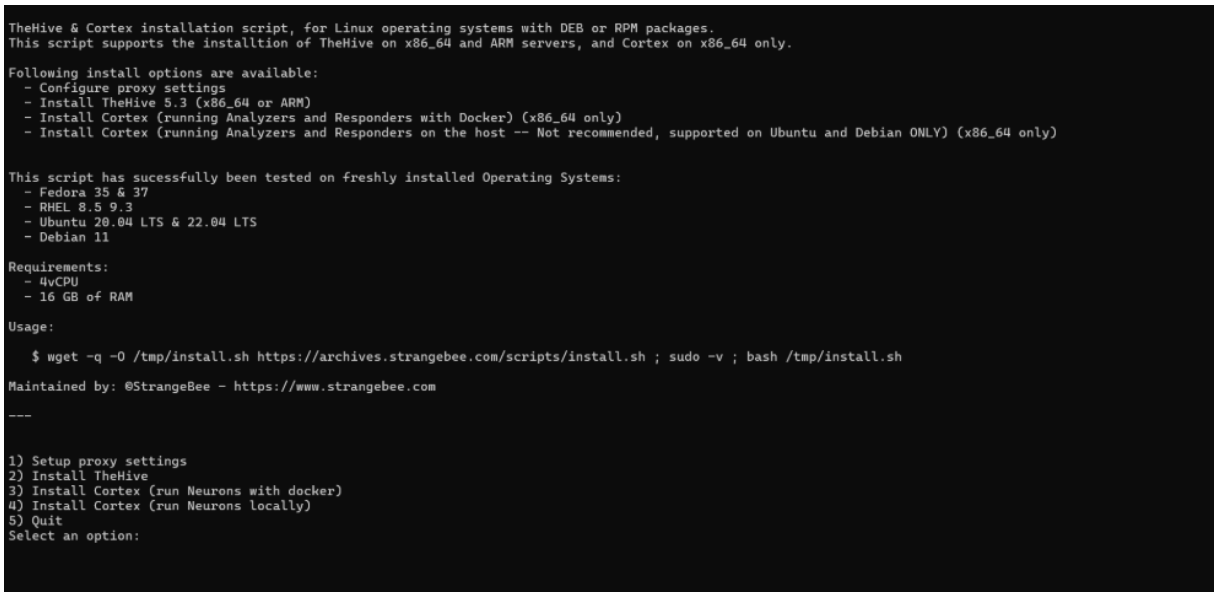
TheHive bestaat uit vijf hoofdcomponenten die nodig zijn om een goede instantie op te kunnen zetten. Hieronder wordt kort toegelicht wat deze componenten doen:

- Java Virtual Machine: Zorgt ervoor dat programma's die geschreven zijn in Java, uitgevoerd kunnen worden op het systeem.
- Apache Cassandra: Een NoSQL-database die gebruikt wordt om cases, alerts en observables op te slaan.
- Elasticsearch: Wordt gebruikt voor de database te doorzoeken en te indexeren.
- File storage: Zorgt ervoor dat bestanden opgeslagen kunnen worden op het lokale bestandssysteem of op een gedeeld opslagvolume.
- TheHive: De hoofdapplicatie die via een webinterface de mogelijkheid biedt om cases en alerts te beheren en onderzoeken. Het staat toe om vlot samen te werken met andere leden van de organisatie.

a) Installatie

De componenten kunnen afzonderlijk geïnstalleerd worden, of via een installatiescript. Bij meerdere pogingen tot handmatige installatie traden er telkens fouten op. Daarom is ervoor gekozen om het installatiescript te gebruiken, wat wel succesvol bleek. Sindsdien is deze methode in gebruik gebleven en is de handmatige installatie niet meer getest.

Het installatiescript kan van de site van TheHive gehaald worden. Het voorziet in de documentatie een commando waarmee dit opgehaald kan worden. Het enige wat dan nog rest is om het script uit te voeren. Het script geeft vijf mogelijkheden die uitgevoerd kunnen worden. Deze kunt u zien op onderstaande schermafbeelding. Er wordt gekozen voor optie '2' om de installatie van TheHive te starten.
Na de installatie is de webinterface van TheHive beschikbaar op http://IP-adres:9000.

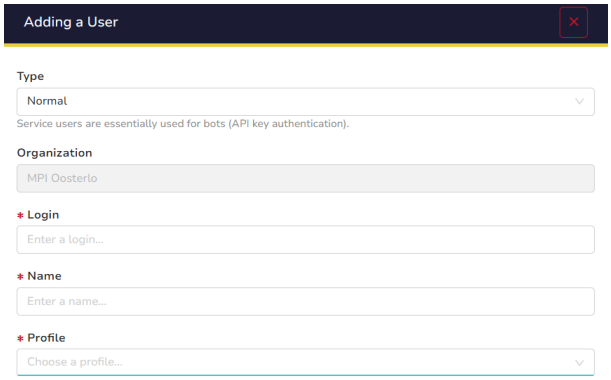


Figuur 46: Schermafbeelding installatiescript TheHive

b) Configuratie

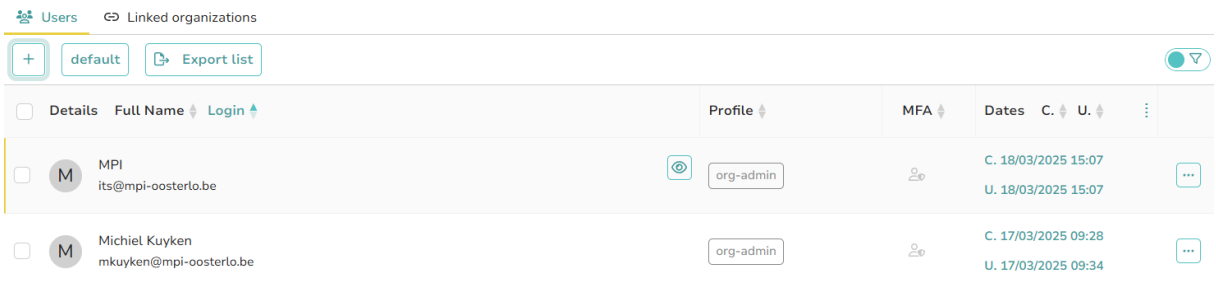
Zoals eerder al vermeld, werkt TheHive met organisaties en gebruikers binnen deze organisaties. Dit wordt gebruikt zodat grote bedrijven verschillende organisaties kunnen opzetten per team. Het MPI Oosterlo heeft een klein IT-team, dus volstaat het om slechts één organisatie hiervoor aan te maken. Dit kan in het adminportaal, waar standaard toegang tot is met gebruikersnaam en wachtwoord admin. Het wordt geadviseerd om dit wachtwoord meteen aan te passen na de eerste inlog. Dit is dan ook direct gedaan in het SOC.

Nu er toegang is tot het adminportaal, kan er een nieuwe organisatie gemaakt worden. Deze krijgt de naam MPI Oosterlo. In de organisatie kunnen gebruikers toegevoegd worden door op de plus te klikken onder ‘users’. Er opent dan het menu dat u op onderstaande schermafbeelding kan zien.
Er zijn verschillende mogelijkheden voor de nieuwe gebruiker. Eerst moet het type bepaald worden, dit kan ‘normal’ of ‘service’ zijn. Een ‘normal’-gebruiker is een standaardgebruiker die voor leden van het team gebruikt wordt. Een ‘serivce’-gebruiker is voor automatisatie te bevorderen. Deze gebruikers worden meestal enkel gebruikt voor API-keys en verbindingen op te zetten met andere services. De gebruikers in de organisatie MPI Oosterlo zijn allemaal type ‘normal’. Dit is omdat zij ook een API-key hebben en een service dus niet van toepassing is.
De nieuwe gebruiker moet een e-mailadres en naam krijgen. In het SOC is het e-mailadres van de IT-dienst gebruikt en heeft het de naam ‘MPI’ gekregen.
Als laatste moet er nog het profiel bepaald worden. Zoals eerder al vermeld zijn er drie verschillende profielen met elk hun eigen rechten. Het account ‘MPI’ krijgt org-admin als profiel, zodat het toegang heeft tot alles. Eén account is in principe voldoende voor het MPI Oosterlo VZW, maar hier kunnen ze in de toekomst extra accounts toevoegen moest dit nodig zijn.



Figuur 47: Schermafbeelding van gebruiker toevoegen aan organisatie

Nu het account aangemaakt is, komt dit bij in de organisatie te staan. Er is nu de mogelijkheid om een wachtwoord toe te voegen en de API-key op te halen. Deze API-key wordt gebruikt voor authenticatie in tools zoals Shuffle, zoals eerder al vermeld in het deel over Shuffle. Dit kan door de gebruiker bekeken worden met het oog dat langs het profiel komt te staan. Op onderstaande schermafbeelding kunt u zien hoe de lijst van gebruikers eruitziet.



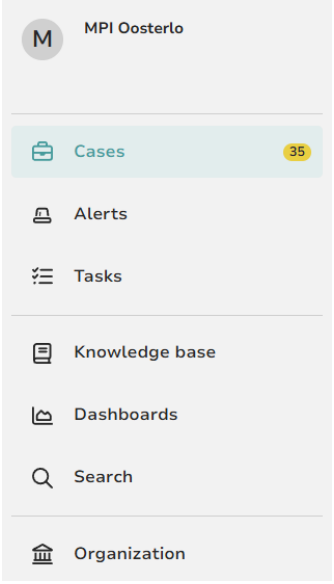
Figuur 48: Schermafbeelding lijst van gebruikers binnen de organisatie

Het laatste dat nog in orde moet worden gebracht, is de licentie van TheHive. Zoals eerder vermeld heeft TheHive een gratis community-versie die gebruikt wordt in het SOC. Dit biedt genoeg functionaliteiten voor de stageopdracht. Er is hiervoor wel een account nodig op het licenseportaal van Strangebee, de overkoepelende organisatie van TheHive en Cortex. Na de eerste aanvraag moet dit elk jaar vervangen worden. Een gedetailleerde handleiding hiervan is te vinden in de documentatie van TheHive.

WEBINTERFACE

Nu TheHive volledig geïnstalleerd is, kan er onderzoek gedaan worden naar de alerts. Hiervoor is er enkel een aanmelding nodig met één van de accounts die binnen de organisatie aangemaakt is.

Standaard komt de gebruiker uit op het tabblad ‘Cases’. Dit toont alle cases die nog openstaan. Links staat er een uitklapmenu waarin er nog meer tabbladen beschikbaar zijn. De belangrijkste hiervan zijn ‘Alerts’, ‘Tasks’ en ‘Organization’, maar de overige worden ook in dit document besproken.
Op onderstaande schermafbeelding ziet u het uitklapmenu met zijn verschillende opties.



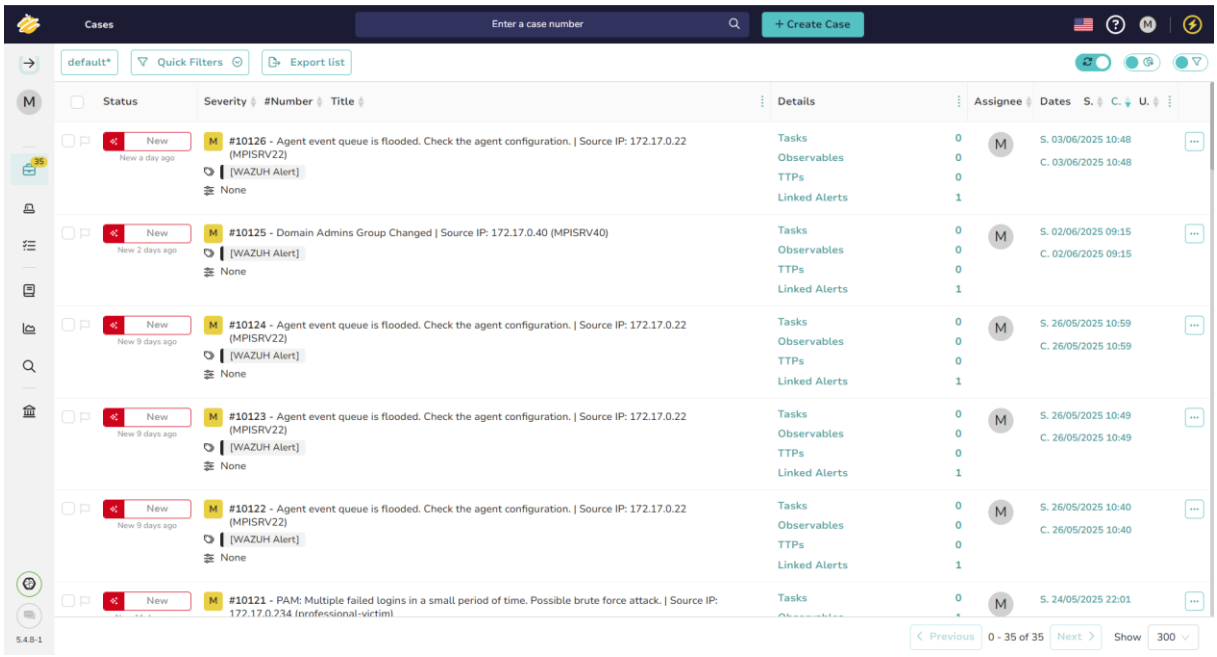
Figuur 49: Schermafbeelding uitklapmenu TheHive

a) Cases

Het eerste tabblad is het tabblad dat standaard geopend wordt nadat er aangemeld is. Dit is het tabblad met alle cases die nog in behandeling zijn. De cases worden automatisch aangemaakt, zoals besproken is in het gedeelte over Shuffle. Bij elke case staat informatie zoals: ID, gebruiker die de case heeft aangemaakt, de datum van aanmaak en de alert(s) die eraan gelinkt zijn. Als een case afgehandeld is, kan deze ook gesloten worden door leden van de organisatie.

Als een case opengeklapt wordt, wordt er meer informatie getoond. Dit gebeurt door een menubalk waarmee andere tabbladen geopend kunnen worden. Hieronder worden de drie belangrijkste tabbladen besproken.
Het eerste tabblad is ‘General’, dit bevat alle algemene informatie over de case. Hier staat de titel, de tag en een beschrijving van de case. Hieruit kan er al veel informatie gehaald worden die nuttig is om analyse op uit te voeren. Dit tabblad kunt u zien in Bijlage 6.
Het tweede tabblad is ‘Tasks’. Hierin kunnen gebruikers nieuwe taken toevoegen voor de analyse van de case. Het bevordert op deze manier het samenwerken binnen teams en geeft een duidelijk overzicht van wat er moet gebeuren of al gebeurd is. Een teamlead kan ook kiezen wie voor welke taak verantwoordelijk is. Dit tabblad kunt u zien in Bijlage 7.
Het laatste tabblad dat in dit document besproken wordt, is ‘Observables’. Hierin komen de observables te staan die met Shuffle aan de case zijn toegevoegd. In dit SOC staat hier het IP-adres van de mogelijke aanvaller. Dit kan dan onderzocht worden door analyzers uit Cortex, die door een API-key verbonden zijn met TheHive. Hier wordt verder op in gegaan in het gedeelte over Cortex.
Dit tabblad kunt u zien in Bijlage 8.

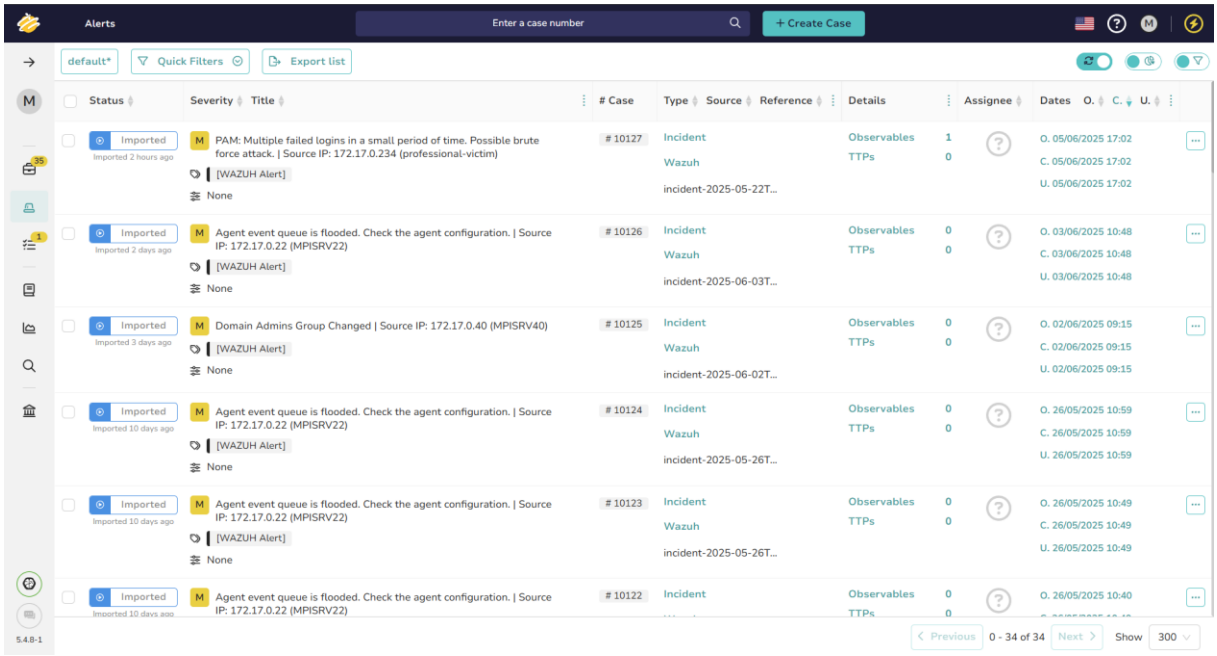
Op onderstaande schermafbeelding kunt u het algemene tabblad ‘Cases’ zien.



Figuur 50: Schermafbeelding tabblad ‘Cases’

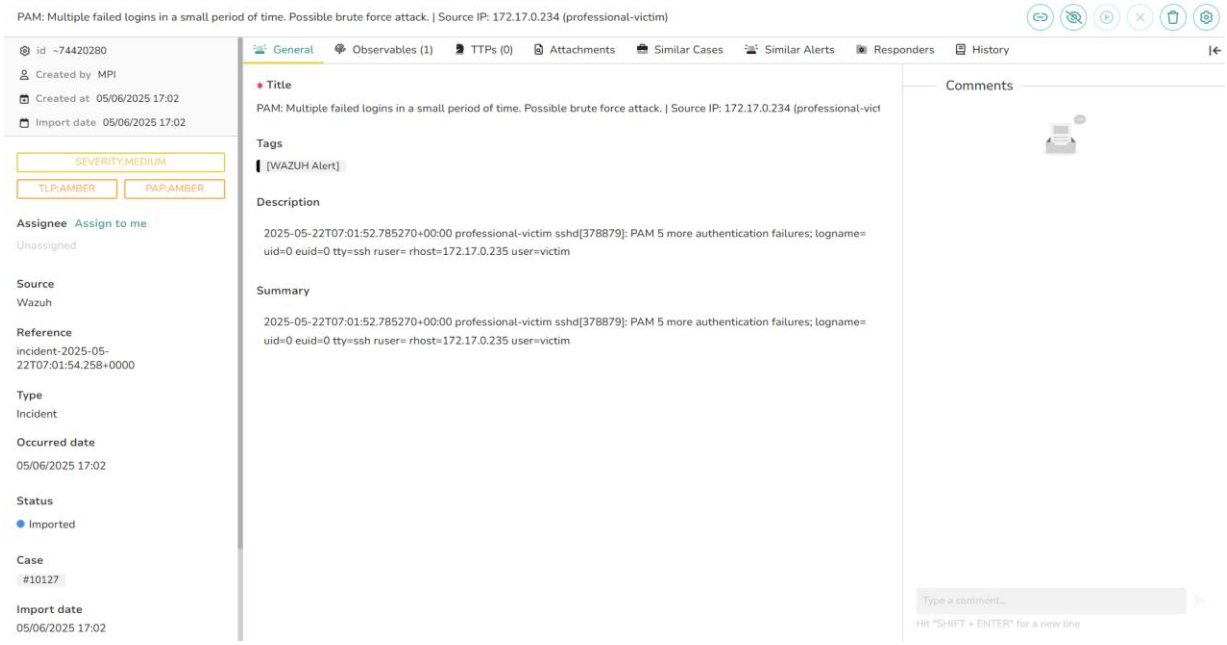
b) Alerts

Het tabblad ‘Alerts’ heeft grotendeels dezelfde functies als het tabblad ‘Cases’, maar dan met alerts. Het toont alle alerts die binnengekomen zijn op TheHive. Er wordt per alert ook dezelfde informatie getoond als bij een case, zoals het ID, de gebruiker die de alert heeft aangemaakt en de datum van aanmaak.



Figuur 51: Schermafbeelding tabblad ‘Alerts’

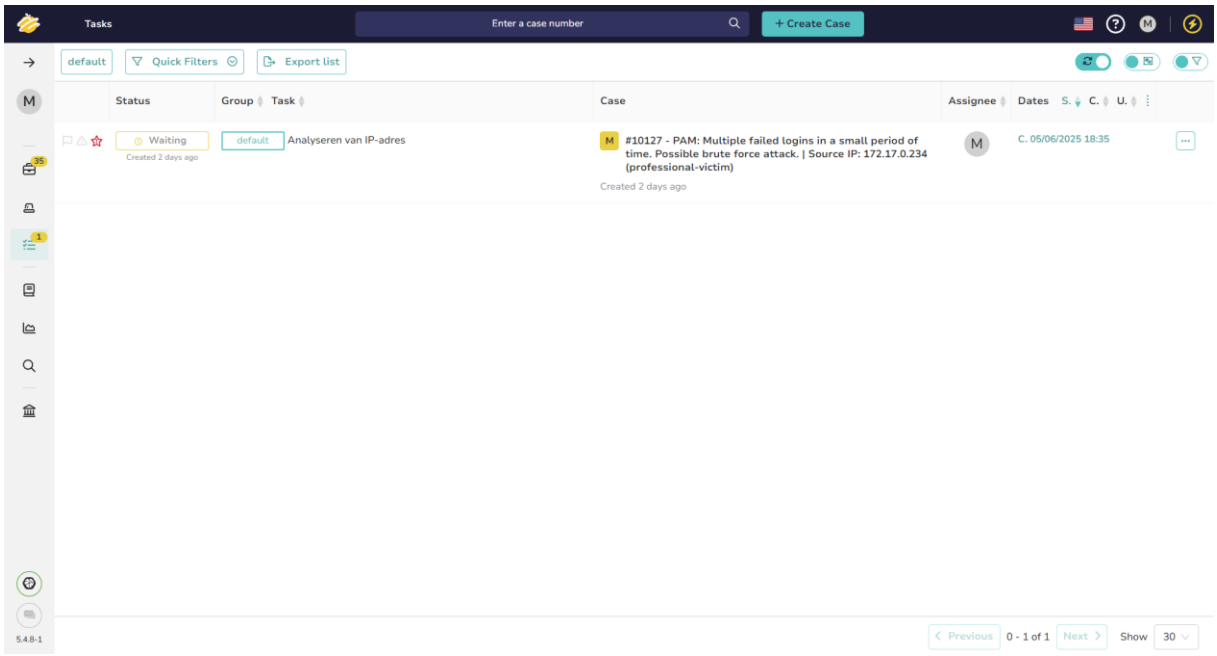
Als een gebruiker de alert openklapt, krijgt het ook grotendeels dezelfde informatie als bij een case te zien. Er wordt algemene informatie gegeven over de alert in het tabblad ‘General’ en de observables zijn zichtbaar in het tabblad ‘Observables’. ‘Tasks’ zijn dan weer niet zichtbaar, omdat het makkelijker is om een case te maken voor meerdere alerts en deze de analysetaken te geven.
Op het tabblad ‘General’ is er een beschrijving en samenvatting van de alert beschikbaar. In het SOC is dit twee keer dezelfde informatie, omdat de case de beschrijving van de alert gebruikt om zijn eigen beschrijving aan te maken. Daarom is de beschrijving hetzelfde als de samenvatting, om ook een goede beschrijving in de case te hebben.



Figuur 52: Schermafbeelding van alert

c) Tasks

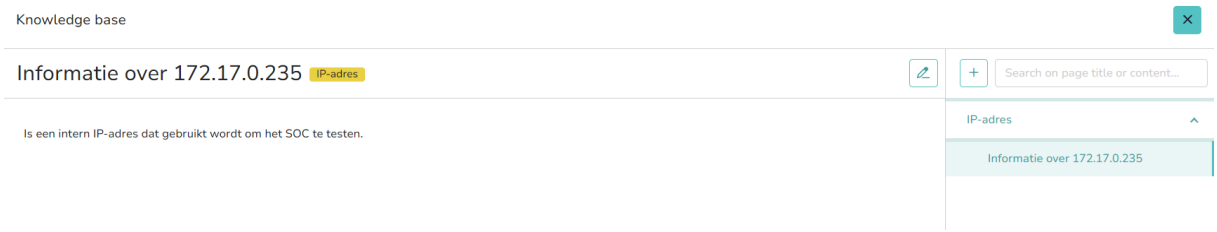
Om op een centraal punt de taken te kunnen zien, voorziet TheHive hiervoor een apart tabblad. Op het tabblad kunnen de gebruikers de status, de case waaraan de taak gelinkt is en de verantwoordelijke zien. Het zorgt ervoor dat leden gemakkelijk van elkaar kunnen zien wie bezig is met welke taak. Zo kunnen ze niet alleen voor zichzelf een duidelijk overzicht krijgen, maar ook voor de leidinggevende die dit kan bijhouden.



Figuur 53: Schermafbeelding tabblad 'Tasks'

d) Knowledge base

Om informatie onderling te kunnen delen, voorziet TheHive een tabblad 'Knowledge base'. Hier kunnen leden van de organisatie informatie toevoegen over observables of alerts. Dit gebeurt aan de hand van pagina's waaraan het lid een titel en beschrijving moet toevoegen. Om de verschillende informatiepagina's te ordenen per onderwerp, wordt er ook een categorie toegevoegd. Zo is de data makkelijk te doorzoeken.



Figuur 54: Schermafbeelding tabblad 'Knowledge base'

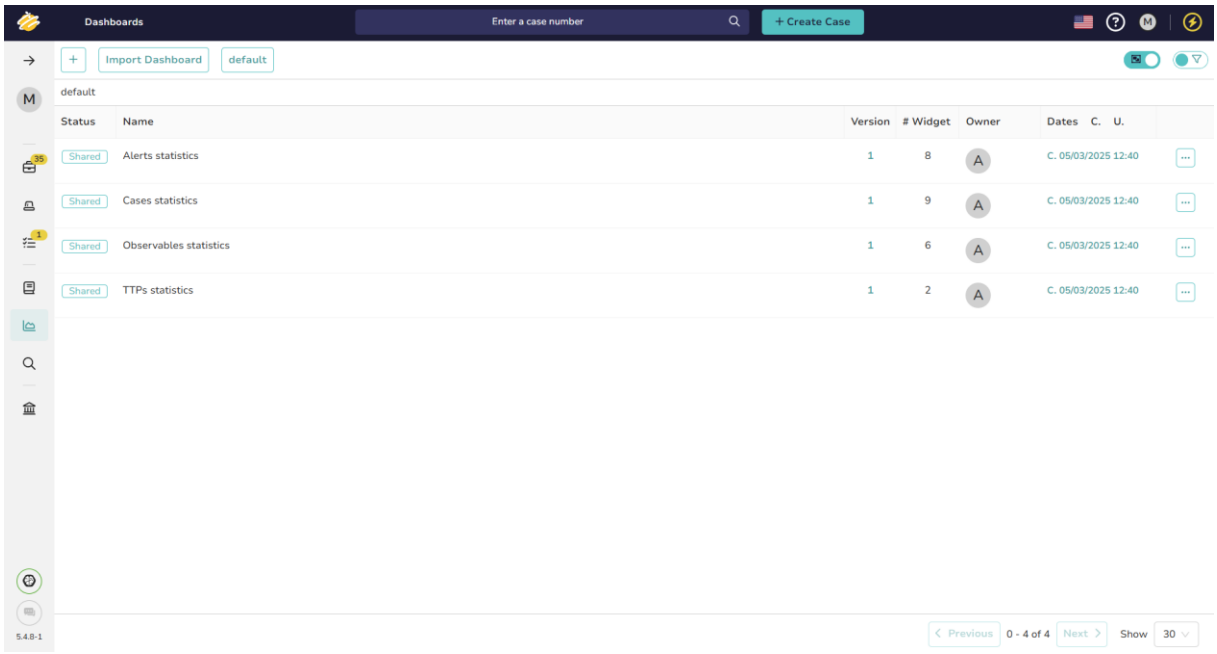
e) Dashboards

Om een overzicht te krijgen van hoeveel alerts en cases er gemaakt zijn, voorziet TheHive een tabblad met verschillende dashboards. Hierop krijgt de gebruiker een overzicht van de verschillende soorten informatie die nuttig kunnen zijn voor een organisatie. In totaal biedt TheHive standaard vier dashboards aan, die u op onderstaande schermafbeelding kunt zien, maar er kunnen er extra aangemaakt worden door de gebruikers.
Hieronder worden drie van de vier dashboards overlopen. Het dashboard over TTP's is voor dit SOC niet relevant, omdat hier nog niets mee gedaan is. Om het document overzichtelijk te houden, kunt u de schermafbeeldingen van de dashboards terugvinden in de bijlagen.

Het eerste dashboard toont de statistieken van de alerts. Belangrijke informatie dat hier getoond wordt zijn de hoeveelheid alerts die per maand aangemaakt worden, de hoeveelheid alerts die er per ernstgraad zijn en de hoeveelheid alerts er van een bepaalde bron komen. Hieruit kunnen teamleads en analisten afleiden wat hun kwetsbaarste maanden zijn en waaraan dit kan liggen.
De schermafbeeldingen van dit dashboard kunt u vinden in Bijlage 9.

Het tweede dashboard toont dezelfde informatie, maar dan voor cases. Eén van de grote verschillen met het dashboard voor alerts, is dat bij de cases er ook de hoeveelheid cases per gebruiker staat. Met de informatie op het dashboard kunnen organisaties opnieuw bekijken wat hun grootste kwetsbaarheden zijn.
De schermafbeelding van dit dashboard kunt u vinden in Bijlage 10.

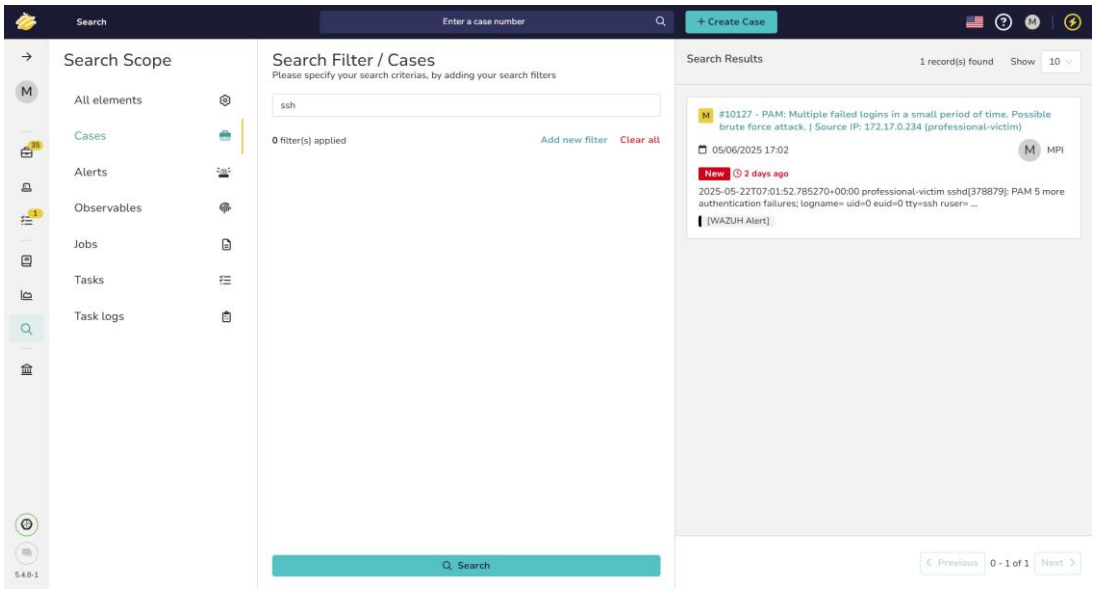
Het laatste dashboard dat besproken wordt, is dat van de observables. Hier wordt net als op de andere dashboards de hoeveelheid observables per maand getoond en per type. Momenteel zijn dit enkel nog IP-addressen, maar later als er extra observables toegevoegd worden, kan hier bekeken worden hoeveel van elk type. De hoeveelheid observables per kleur van het TLP wordt ook getoond, maar deze staat ook vast op 'amber'.
De schermafbeelding van dit dashboard kunt u vinden in Bijlage 11.



Figuur 55: Schermafbeelding tabblad 'Dashboards'

f) Search

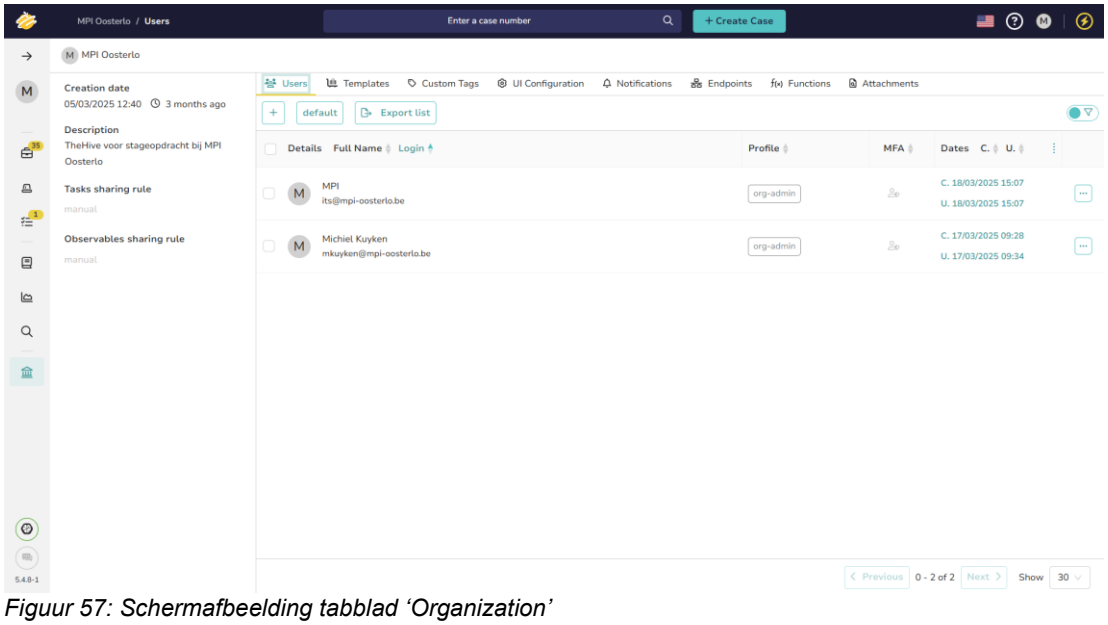
Om het zoeken naar cases en alerts makkelijker te maken, is er een ‘Search’-tabblad voorzien door TheHive. Op dit tabblad kan er gefilterd worden op de verschillende elementen en kan er gezocht worden naar een woord dat erin voor moet komen. In het voorbeeld dat u hieronder kunt zien, is er gezocht naar alle cases die het woord ssh bevatten. Zo kan er snel de juiste case opgehaald worden zonder te moeten scrollen door alle cases in het tabblad ‘Cases’.



Figuur 56: Schermafbeelding tabblad ‘Search’

g) Organization

Ten slotte blijft het tabblad ‘Organization’ over. Hierin staat alle informatie over de organisatie waarin de gebruiker zich bevindt. Dit kan gaan over de dag dat de organisatie is aangemaakt en de leden tot de zelfgemaakte templates en tags en de UI configuratie die aangepast kan worden. Het is ook hier dat de API-key voor Cortex toegevoegd wordt om analyses uit te kunnen voeren binnen cases.
Het is vooral voor gebruikers met de rol org-admin een handige tool om snelle aanpassingen te maken, zonder in te moeten loggen op het admin-account.
Op de schermafbeelding hieronder kunt u het standaardscherm zien. In dit document worden de verschillende tabbladen en opties hierin niet uitgelegd, omdat het anders te onoverzichtelijk wordt.



Figuur 57: Schermafbeelding tabblad ‘Organization’

3.2.3. Cortex

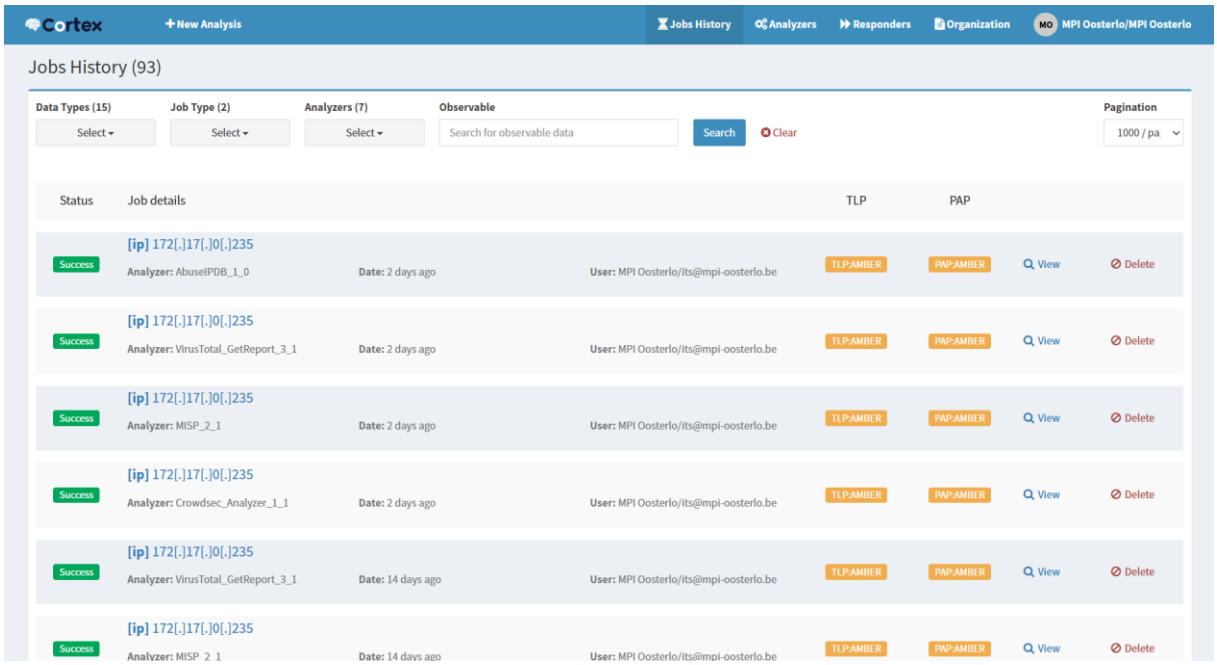
Nu de alerts in TheHive aangemaakt worden, moeten ze nog geanalyseerd kunnen worden. Hiervoor wordt er gebruik gemaakt van Cortex. Cortex voorziet verschillende analyzers, waarmee de alerts en observables geanalyseerd kunnen worden. De installatie van Cortex en hoe de analyzers toegevoegd worden, worden in dit deel besproken. De realisatie van Threat Intelligence komt hier ook aan bod, omdat dit de analyzers vormen die door Cortex gebruikt worden.

SETUP

De installatie van Cortex, komt grotendeel overeen met de installatie van TheHive. Dit komt omdat ze allebei van dezelfde maker zijn, StrangeBee. Voor de installatie en configuratie zijn dezelfde stappen overlopen als in de installatie van TheHive. Het begint met hetzelfde installatiescript, maar nu wordt optie drie gekozen. Dit installeert de nodige componenten in Docker containers.
De componenten die Cortex nodig heeft, zijn ook ongeveer hetzelfde als bij TheHive. Hieronder worden ze kort toegelicht:

- Java Virtual Machine: Zorgt ervoor dat programma's die geschreven zijn in Java, uitgevoerd kunnen worden op het systeem.
- Elasticsearch: Wordt gebruikt om de database te doorzoeken en te indexeren.
- Docker: Docker wordt in Cortex gebruikt om analyzers en responders uit te voeren. Elke analyzer en responder heeft zijn eigen Docker Image, wat het gebruik hiervan makkelijker maakt en met minder kans op fouten.
- Cortex: De hoofdapplicatie die via een webinterface de mogelijkheid biedt om analyzers en responders te beheren en gebruiken.

Nu Cortex succesvol is geïnstalleerd, kan de configuratie beginnen. Cortex en TheHive lijken hierin ook weer hard op elkaar. Voor beide moet een organisatie aangemaakt worden via het admin-account en hier kunnen leden aan toegevoegd worden. Dit gebeurt op de webinterface die beschikbaar is op http://IP-adres:9001 Omdat dit al uitgebreid besproken is in het deel over TheHive, gaat er hier niet verder op ingegaan worden. Na het inloggen met één van de leden van de organisatie, wordt de onderstaande webinterface getoond.



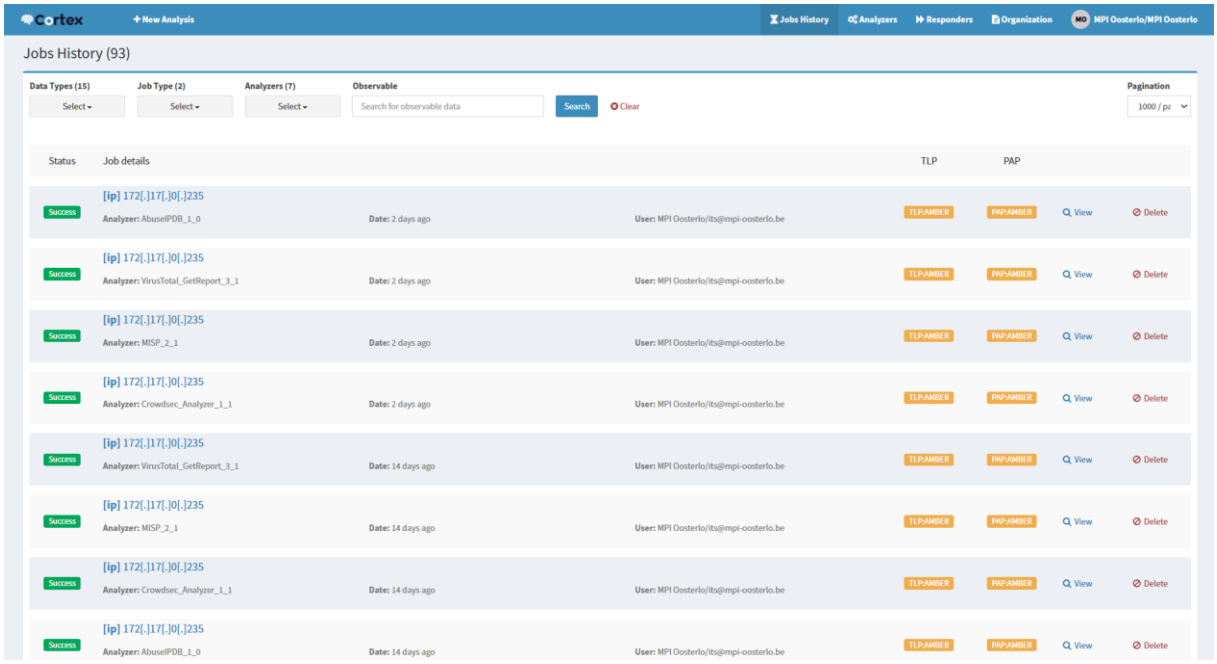
Figuur 58: Schermafbeelding webinterface Cortex

WEBINTERFACE

Op de webinterface zijn er net zoals bij TheHive verschillende tabbladen zichtbaar. Op deze tabbladen staat er informatie over Cortex zelf en over de taken die het uitgevoerd heeft. Van deze tabbladen worden er in dit document drie van de vier besproken. Het tabblad ‘Responders’ wordt niet besproken omdat dit niet in het SOC gebruikt is. De overige tabbladen worden in de volgende punten uitgelegd.

a) Jobs History

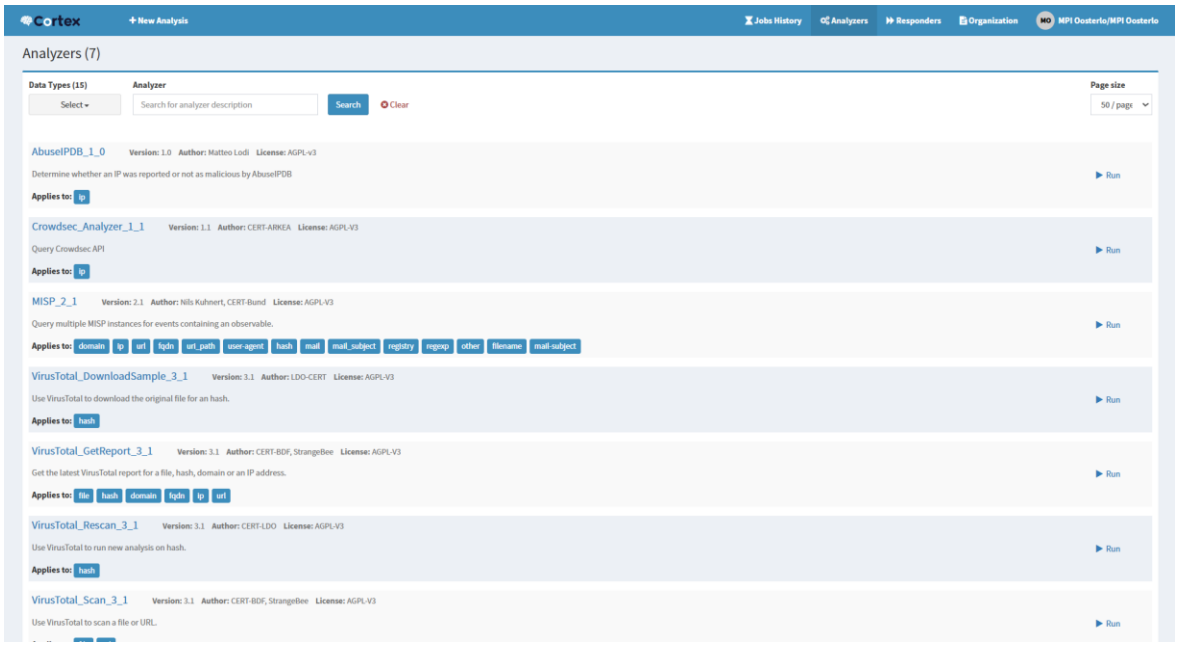
Het eerste tabblad toont alle taken die Cortex heeft uitgevoerd. Binnen het SOC werken Cortex en TheHive nauw samen. Via Shuffle worden de observables die aan een case in TheHive gekoppeld zijn, automatisch geanalyseerd door analyzers in Cortex. Op die manier kan snel worden bepaald of een observable kwaadaardig is of niet.
De resultaten van die analyses zijn echter niet direct zichtbaar in TheHive. Om dit deels op te lossen, wordt in het SOC gebruikgemaakt van een API-sleutel die Cortex toevoegt aan TheHive. Hierdoor is het mogelijk om vanuit een case in TheHive een analyse te starten. De resultaten van die analyses verschijnen echter niet in dit tabblad, waardoor de zichtbaarheid alsnog beperkt blijft.
Het tabblad ‘Job History’ wordt vooral gebruikt om patronen te kunnen herkennen, zoals veel terugkerende IP-adressen of domeinnamen. Ook kan er handmatig een analyse gestart worden en hoeft dit dus niet altijd via een case te gebeuren.
Op onderstaande schermafbeelding kunt u het tabblad zien.



Figuur 59: Schermafbeelding tabblad ‘Jobs History’

b) Analyzers

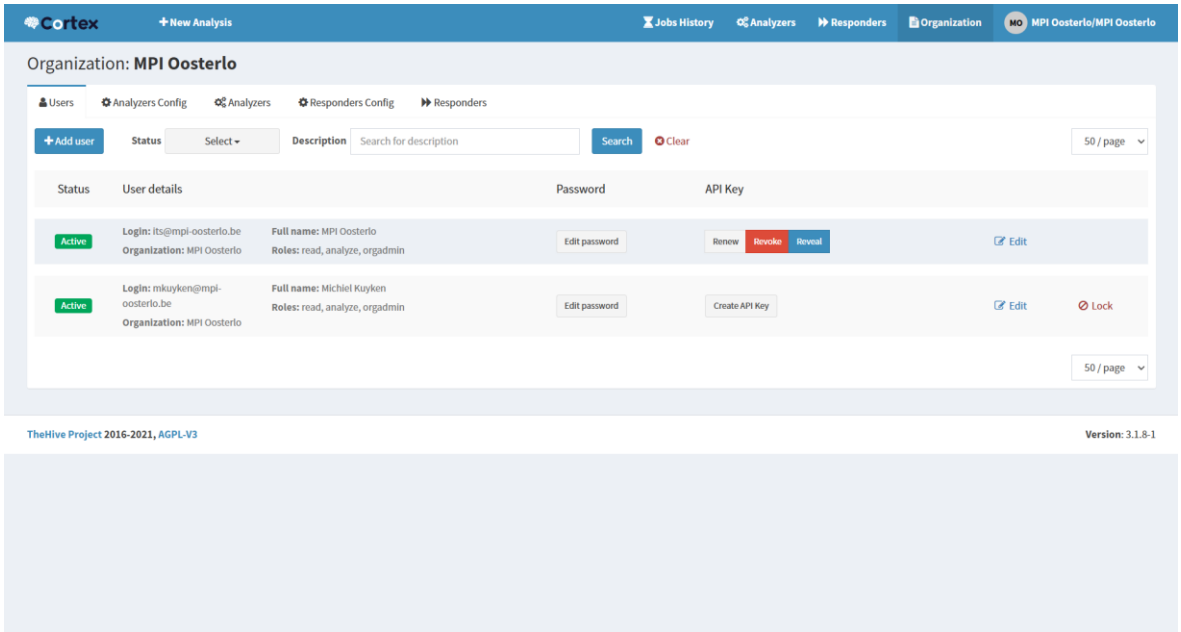
In het tabblad ‘Analyzers’ staan alle analyzers die geconfigureerd zijn. Een organisatie kan zo bekijken welke analyzer goed is om een bepaald soort analyse uit te voeren. Ze kunnen hier dan ook snel merken of er nog een tool mist om een specifieke analyse uit te voeren en hiernaar opzoek gaan.
In het SOC zijn dit de tools die gebruikt worden als analyzers: AbuseIPDB, Crowdsec, VirusTotal en MISP. Hoe deze worden toegevoegd, wordt verder besproken in het tabblad ‘Organization’.



Figuur 60: Schermafbeelding tabblad ‘Analyzers’

c) Organization

Het laatste tabblad is ‘Organization’. Net als in TheHive dient dit om aanpassingen te maken in de configuratie van de organisatie. Dit kan gaan over het toevoegen van nieuwe gebruikers tot het aanmaken van API-sleutels. Een ander belangrijke configuratie die via dit tabblad gebeurt, is de configuratie van analyzers en reponders. In dit SOC is er enkel gebruik gemaakt van analyzers en wordt enkel dit besproken. Het gebruik van responders is door het MPI Oosterlo VZW als out-of-scope bepaald, omdat ze dit liever zelf in handen hebben.

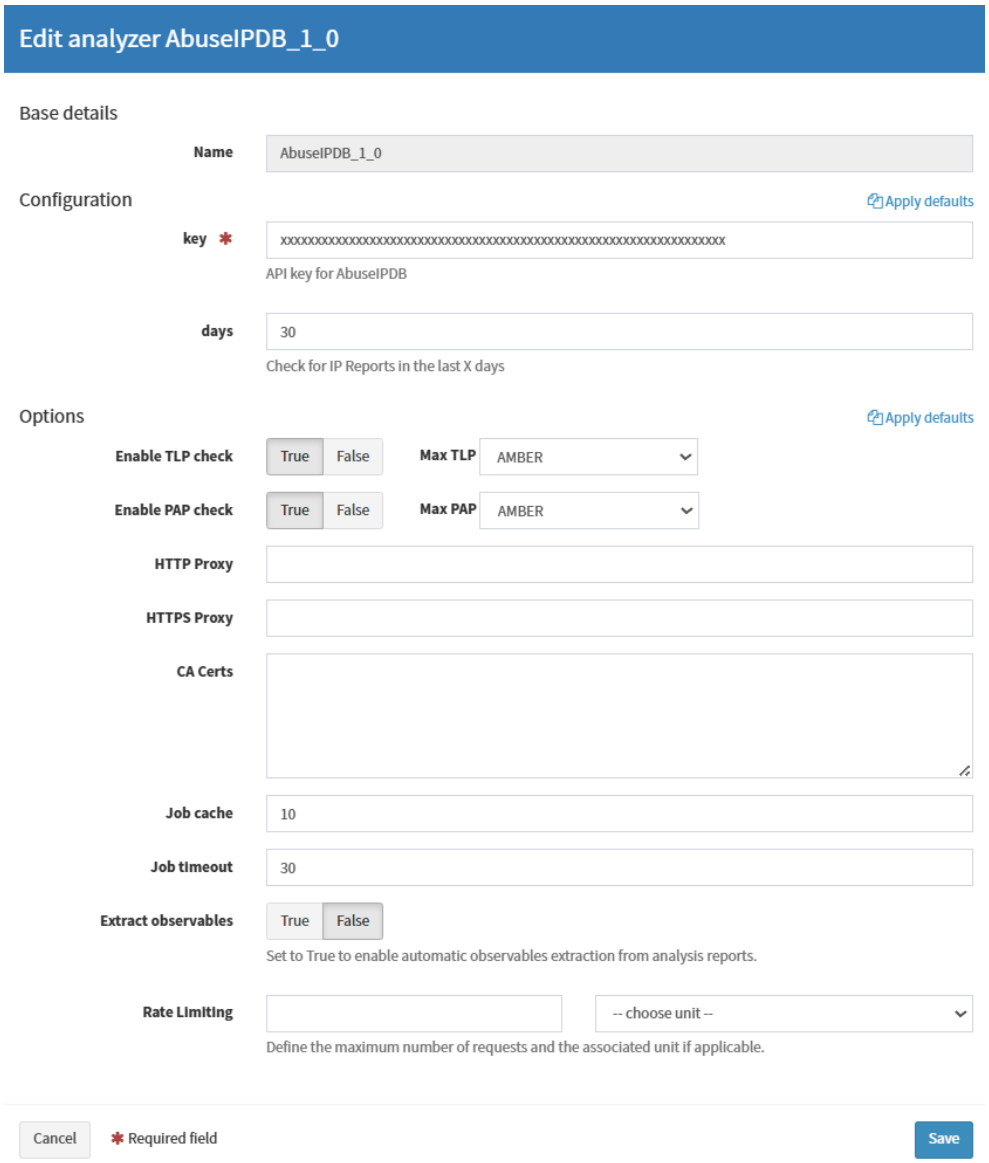


Figuur 61: Schermafbeelding tabblad ‘Organization’

AbuseIPDB

De eerste analyzer die besproken wordt, is AbuseIPDB. Dit is een analyzer die enkel IP's onderzoekt en daarna een rapport geeft of het kwaadaardig is of niet. In dit rapport staat ook welke landen het IP als kwaadaardig beschouwen en voor welke reden.

Om AbuseIPDB toe te voegen als analyzer, is er slecht één vereiste nodig. Dit is een API-sleutel van een geldig AbuseIPDB account. Dit account kan gemakkelijk aangemaakt worden op hun site en daarna kan de API-sleutel geregistreerd worden. Na deze sleutel toegevoegd te hebben, kan de rest op de standaardwaarden blijven staan. Deze beïnvloeden de algemene werking van de analyzer niet en dienen enkel voor de optimalisering.
Op de onderstaande schermafbeelding kunt u de configuratie van AbuseIPDB zien in Cortex.
Een voorbeeld van een rapport van AbuseIPDB kunt u vinden in Bijlage 12.

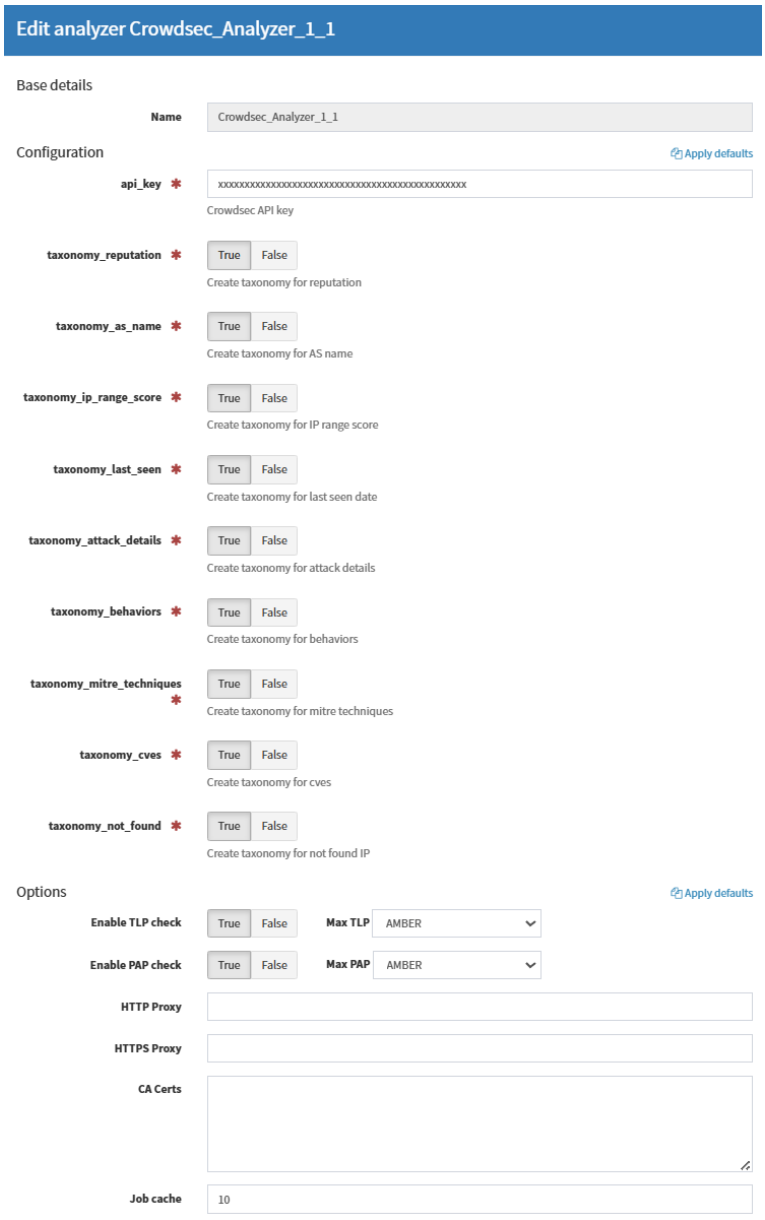


Figuur 62: Schermafbeelding configuratie AbuseIPDB

Crowdsec

De tweede analyzer die besproken wordt, is Crowdsec. Dit is een tool waarvan er gevraagd werd of er een integratie mogelijk was. Deze integratie is er uiteindelijk gekomen met hun CTI-tool die onderzoek kan doen naar IP-adressen. Hierover geeft het dan informatie of het kwaadaardig is en vanwaar het afkomstig is. Het toont ook welke aanvallen gelinkt zijn aan het IP-adres.

Om Crowdsec toe te voegen aan de lijst van analyzers, zijn er enkele verplichte velden. De eerste is een API-sleutel van een geldig Crowdsec-account. Dit is makkelijk te verkrijgen door een account aan te maken op Crowdsec en de API-sleutel te laten genereren.
Onder de API-sleutel staan veel verschillende opties om aan of uit te zetten. Deze komen echter allemaal op hetzelfde neer. Er wordt gevraagd of de optie toegevoegd moet worden aan de output van Crowdsec. Dit houdt in dat als 'taxonomy_reputation' aanstaat, dat de reputatie van het IP-adres wordt toegevoegd aan het rapport. Het is best om dit allemaal aan te zetten, omdat dit meer informatie geeft over het IP-adres en waarom het wel of niet kwaadaardig is.
Op de onderstaande schermafbeelding kunt u de configuratie van Crowdsec zien in Cortex.
Een voorbeeld van een rapport van Crowdsec is zichtbaar in Bijlage 13.

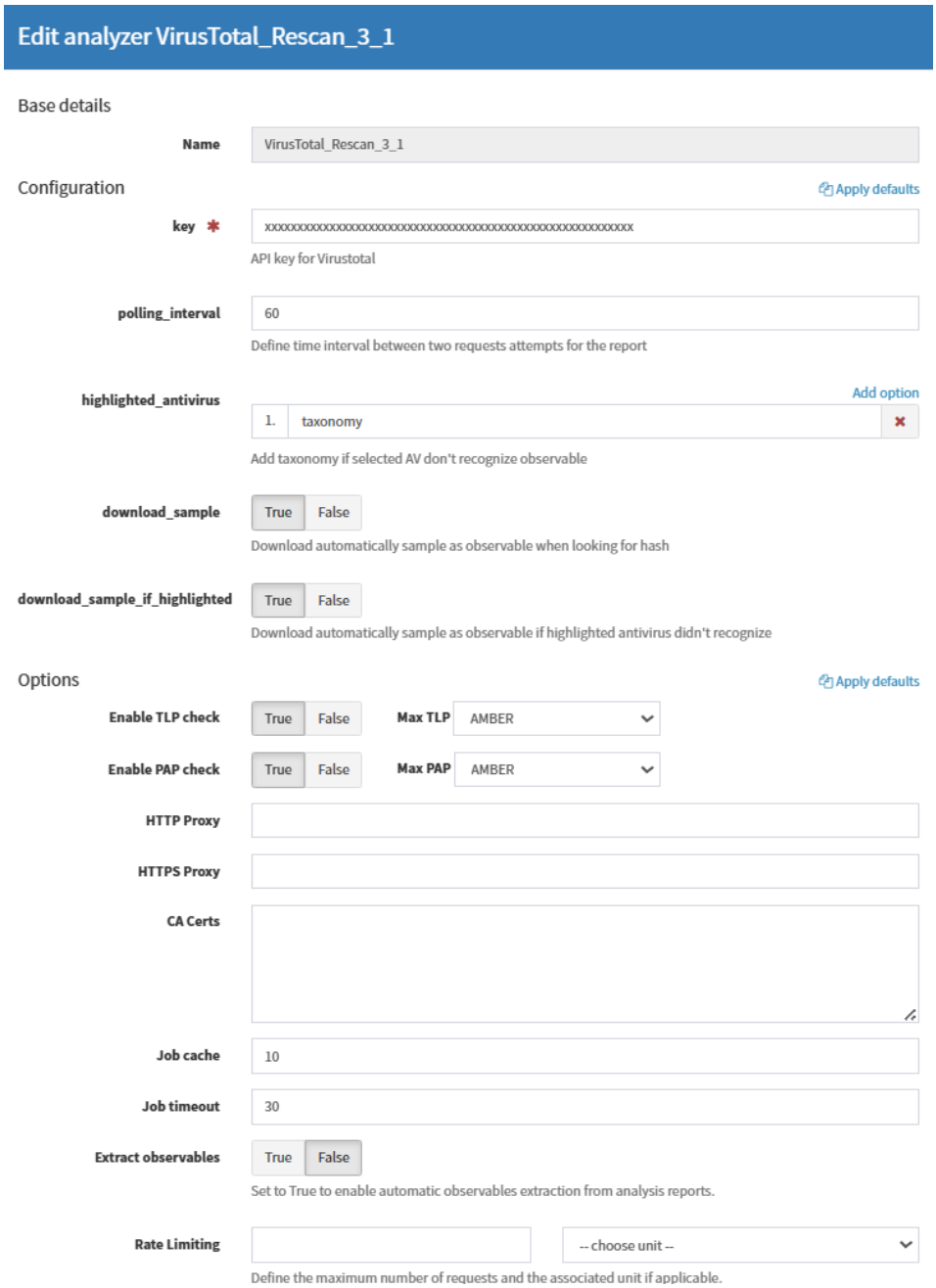


Figuur 63: Schermafbeelding configuratie Crowdsec

VirusTotal

De derde analyzer is VirusTotal, een van de bekendste online analyseplatformen. In tegenstellig tot AbuseIPDB en Crowdsec kan VirusTotal veel meer analyseren dan enkel IP-adressen. Zoals eerder in de analyse al vermeldt, biedt het mogelijkheden aan om IP-adressen, domeinnamen, bestanden en URL's te controleren. VirusTotal kan daarom niet ontbreken in het SOC, omdat het zowel open-source is en uitgebreide analyses kan doen.

Om VirusTotal toe te voegen als analyzer, is er enkel een API-sleutel nodig. Deze kan opnieuw verkregen worden door een account aan te maken en de API-sleutel hiervan te kopiëren. De overige opties zijn opnieuw niet verplicht en laten we op hun standaardwaarde staan.
Op onderstaande schermafbeelding kunt u de configuratie van VirusTotal zien in Cortex.
Een rapport van VirusTotal kunt u vinden in Bijlage 14.



Figuur 64: Schermafbeelding configuratie VirusTotal

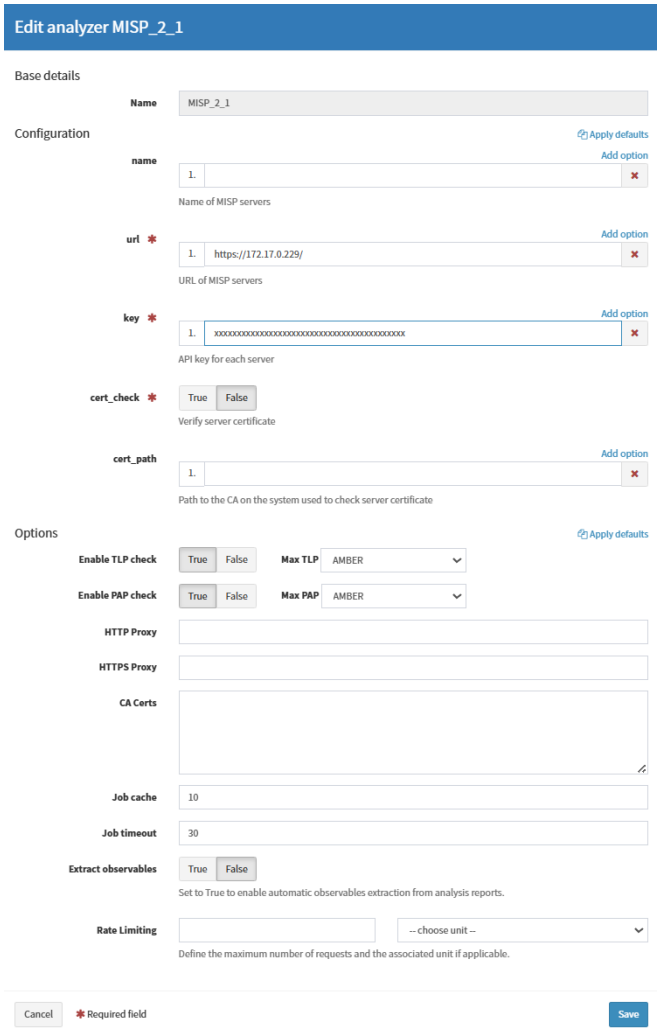
MISP

De laatste analyzer die besproken wordt in dit document is MISP. MISP is een threat intelligence platform dat dient om informatie tussen organisaties te kunnen delen. Veel grote bedrijven bieden hier ook hun gevonden informatie aan, zodat anderen zich beter hiertegen kunnen beveiligen. MISP kan gebruikt worden om zowel nieuwe informatie toe te voegen als informatie op te halen. Deze informatie kan gebruikt worden bij tal van observables, zoals IP-adressen maar ook de verzender van een mail.

Het ophalen van informatie is de reden waarom het als analyzer gebruikt kan worden in Cortex. Door te vergelijken met andere informatie die al geanalyseerd is of vaak terugkomt, kan er bepaald worden of iets kwaadaardig is.

Om MISP toe te voegen, zijn er meer stappen vereist dan bij de vorige analyzers. MISP heeft namelijk zijn eigen VM nodig en niet gewoon een account. Na de installatie van MISP is er een webinterface beschikbaar. Cortex heeft de URL van deze webinterface nodig om een verbinding op te kunnen stellen.
De tweede vereiste is een API-sleutel, die verkrijgbaar is via de webinterface.
De laatste vereiste is de keuze of certificaten gecontroleerd moeten worden. Omdat de MISP op een lokale VM draait, staat dit uit om conflicten te voorkomen.

Uiteindelijk wordt MISP niet gebruikt in het SOC. Tijdens het opzetten hiervan konden er geen informatielijsten opgehaald worden van andere organisaties. Na verschillende pogingen gedaan te hebben en door een gebrek aan tijd, is MISP gedeeltelijk geschrapt uit het SOC. De analyzer is nog steeds actief, maar kan geen analyse uitvoeren omdat er geen informatie aanwezig is.
Op onderstaande schermafbeelding kunt u de configuratie van MISP in Cortex zien.



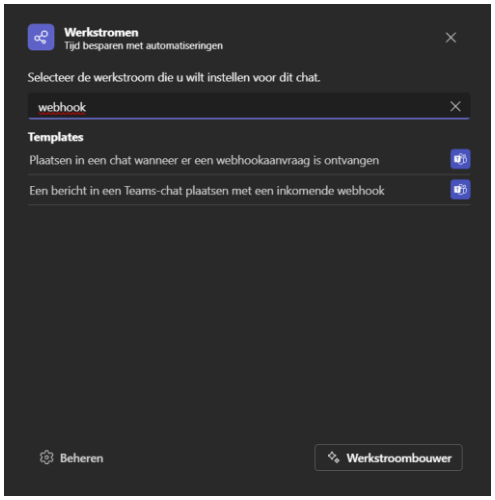
Figuur 65: Schermafbeelding configuratie MISP

3.2.4. Teams

Het laatste deel van de realisatie, is het maken van een melding in Teams. Voor Teams is er geen installatie nodig. Het staat standaard op alle laptops binnen de organisatie geïnstalleerd. Het enige dat wel nodig was, was een aparte chat aanmaken waarin de meldingen van het SOC binnenkomen. Om deze meldingen hier te krijgen, werd er gebruik gemaakt van een werkstroom. Hieronder wordt uitgelegd hoe deze werkstroom opgezet is.

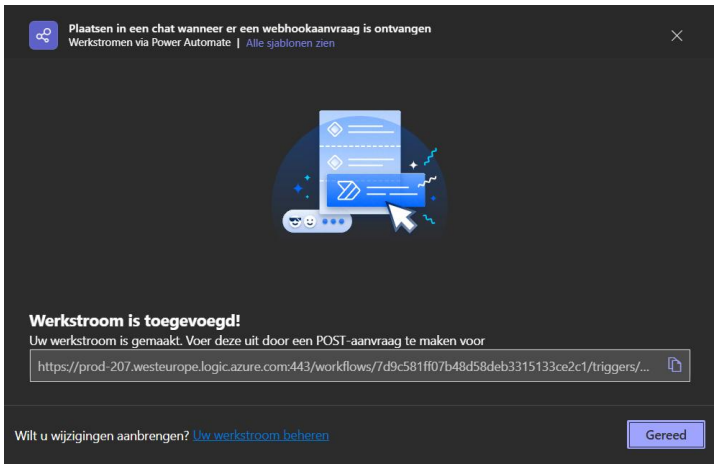
WERKSTROOM

Om een werkstroom op te zetten in Teams, moet er eerst bepaald worden aan welke chat het gekoppeld moet worden. Hiervoor is er in Teams een chat opgezet genaamd 'SOC Alerts' met daarin de werknemers van de helpdesk. Zij zullen na de stage het project verder onderhouden en moeten reageren op alerts die binnenkomen. In deze chat is het mogelijk om een werkstroom op te zetten die een eigen webhook heeft. Als deze webhook aangesproken wordt, dan kan er een bericht in de chat geplaatst worden.
Op onderstaande schermafbeelding ziet u de keuzes binnen het opstellen van een werkstroom met webhook. Er wordt gekozen voor de bovenste optie.



Figuur 66: Schermafbeelding werkstroom opzetten in Teams met webhook

De werkstroom kan nu worden opgezet. Er moet hiervoor eerst een naam gekozen worden en een verbinding gemaakt worden met het account dat de werkstroom opzet. Zonder een geldig account kan de webhook en werkstroom niet gebruikt worden. Ten slotte moet er enkel nog gekozen worden aan welke chat de werkstroom gelinkt moet zijn. Er wordt dan een webhook-URI getoond. Deze webhook werkt via Power Automate en kan in Shuffle gebruik worden om een melding te sturen.
Op onderstaande schermafbeelding kunt u een voorbeeld van een succesvolle webhook zien.

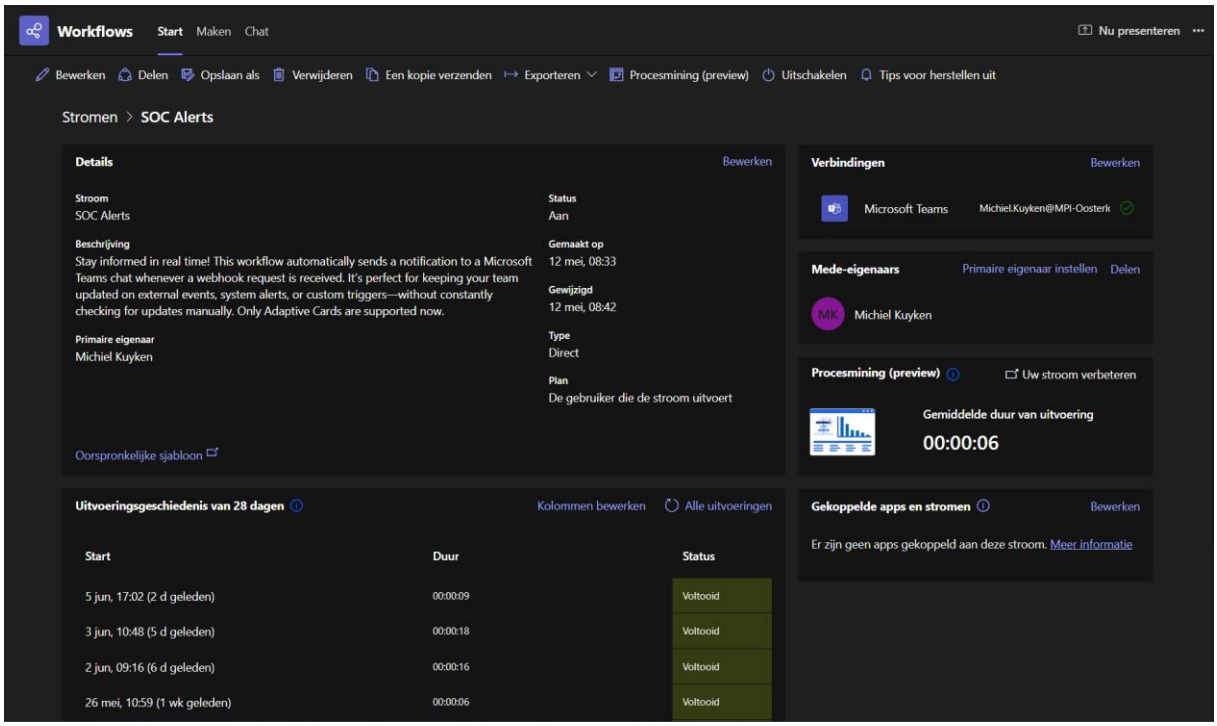


Figuur 67: Schermafbeelding succesvolle opzet webhook

Om beter inzicht te krijgen in de werkstromen en deze te beheren, biedt Microsoft Teams een apart tabblad genaamd ‘Workflows’. Hier worden alle werkstromen weergegeven die een gebruiker zelf heeft aangemaakt, of waaraan hij of zij is toegevoegd via een chat. De zojuist aangemaakte werkstroom is hier terug te vinden, inclusief informatie over de werking en uitgevoerde acties. Vanuit dit overzicht kunnen werkstromen ook beheerd worden, zoals het tijdelijk uitschakelen of volledig verwijderen ervan.

Binnen dit dashboard zijn er twee belangrijke tabbladen. Het eerste is ‘Details’. Dit bevat algemene informatie over de werkstroom, zoals wie deze heeft aangemaakt en wanneer. Dit maakt het eenvoudig om verantwoordelijkheden op te volgen en overzicht te bewaren.

Het tweede tabblad is ‘Uitvoeringsgeschiedenis’. Hier worden alle acties weergegeven die de werkstroom heeft uitgevoerd binnen een bepaalde periode. Tijdens het opzetten van het SOC bleek dit tabblad zeer nuttig bij het oplossen van problemen. Zo werd hier ontdekt waarom meldingen niet in de Teamschat verschenen: de oorzaak lag bij een verkeerd opgestelde JSON-code die niet aan de vereiste structuur voldeed. Dit tabblad is dus een handig hulpmiddel voor foutopsporing en het controleren van verbindingen. Op onderstaande schermafbeelding kunt u de informatie van de werkstroom in het SOC bekijken.



Figuur 68: Schermafbeelding van werkstroom in Teams

4. Besluit

Met de realisatie van dit SOC bij het MPI Oosterlo VZW werd er een belangrijke stap gezet in de richting van een meer beveiligde en alerte omgeving. De opzet van het SOC zorgt voor een gecentraliseerd en overzichtelijk systeem dat, op een efficiënte manier, automatisch incidenten verwerkt. De tools die gebruikt zijn, zorgen voor een volledig open-source en krachtige oplossing.

Tijdens de stage werden de noden van de organisatie steeds duidelijker en werden grenzen afgebakend. In het projectplan werd een groot SOC beschreven met alle mogelijkheden die tijdens de lessen gezien waren. Al deze mogelijkheden bleken te uitgebreid om op drie maanden te realiseren. Daarom werd er een aangepast plan opgesteld wat enerzijds haalbaar was en anderzijds aan alle basisnoden voldoet: een SOC dat alerts automatisch kan genereren, analyseren en afhandelen. Deze alerts moeten zichtbaar zijn op een centraal dashboard en de helpdesk krijgt een melding als er een nieuwe alert is. Al deze doelen zitten in het SOC verwerkt. Door gebruik te maken van Wazuh worden er alerts gegenereerd en getoond op het Wazuh Dashboard. Deze alerts komen in TheHive waar er analyse gedaan kan worden met Cortex. Hierna komt er een melding in Teams. Shuffle coördineert alle tools om deze fases automatisch te laten verlopen. Een automatische afhandeling van de alert werd door het MPI Oosterlo VZW als out-of-scope bepaald. Hier houden ze graag zelf de controle over.

Natuurlijk staat de wereld van cybersecurity nooit stil en kan dit SOC aangepast en geüpdatet blijven worden. Eén van de grootste problemen waar ik mee kampte tijdens de stage, was het maken van een uniform logformaat. Door hier gebruik van te maken, wordt het toevoegen van observables makkelijker en kunnen er betere analyses gedaan worden.

In Cortex is er de mogelijkheid om nog meer analyzers toe te voegen. Er kunnen nooit genoeg threat intelligence-tools aanwezig zijn, zodat er zekerheid is dat er geen valse positieven zijn. MISP is een threat intelligenceplatform dat niet afgeraakt is tijdens de stageperiode, maar dat zeker nuttig kan zijn.

In het SOC worden alle servers gemonitord, maar de apparaten van werknemers nog niet. Omdat deze ook een bedreiging kunnen vormen, is het een goed idee om deze op termijn ook toe te voegen aan het SOC. Zo zijn er minder plaatsen voor aanvallers om binnen te kunnen dringen.

Tenslotte is het vooral aangeraden om het SOC verder te optimaliseren. Er komen nog veel alerts binnen die geen echte bedreiging vormen, maar die de bedreigingen verstoppen tussen alle meldingen. Door de alerts die binnenkomen te onderzoeken, kan het SOC verder geoptimaliseerd worden zodat enkel belangrijke alerts binnenkomen. Dit is iets wat op lange termijn moet blijven gebeuren en waar ze veel baat bij gaan hebben.

Als ikzelf terugkijk naar mijn stage en naar het SOC, ben ik tevreden over wat ik heb neergezet. De basisdoelen zijn bereikt en ik heb bij kunnen dragen aan de veiligheid van de organisatie. Tijdens de stage heb ik veel bijgeleerd, niet alleen dankzij mijn stageopdracht, maar ook door mee te draaien op de helpdesk. Zowel mijn hard skills als soft skills heb ik hier kunnen bijspijkeren en het is een onvergetelijke leerervaring geweest.

LITERATUURLIJST

Crowdsec. (z.d.). *Documentatie Crowdsec*. Opgeroepen op Mei 2025, van Crowdsec: https://doc.crowdsec.net/u/getting_started/intro

Docker. (z.d.). *Install Docker Engine on Ubuntu*. Opgeroepen op Mei 2025, van Dockerdocs: https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository

Elastic. (z.d.). *Documentatie Filebeat*. Opgeroepen op Mei 2025, van Elastic: https://www.elastic.co/docs/reference/beats/filebeat/

Fluentd. (z.d.). *Documentatie Fluentd*. Opgeroepen op Mei 2025, van Fluentd: https://docs.fluentd.org/

Graylog. (z.d.). *Documentatie Graylog*. Opgeroepen op Mei 2025, van Graylog: https://go2docs.graylog.org/current/what_is_graylog/what_is_graylog.htm?tocpath=What%20Is%20Graylog%253F%7C_____0

Java. (z.d.). *Wat is Java-technologie en waarom heb ik dit nodig?* Opgeroepen op Mei 2025, van Java: https://www.java.com/nl/download/help/whatis_java.html

javiersanchz. (2025, Januari 27). *Wazuh Integrations*. Opgeroepen op Mei 2025, van Github: https://github.com/wazuh/wazuh/tree/main/integrations

Karkoub, E. (2024, Maart 15). *Integrating Wazuh with Fluentd for unified logging*. Opgeroepen op Mei 2025, van Wazuh: https://wazuh.com/blog/forward-alerts-with-fluentd/

Landeghem, G. V., & Kenens, L. (2023, December 18). *Security Platforms*. Opgeroepen op Mei 2025, van https://docs.google.com/document/d/1_CuyULnFh-38qxlOvwqqwnikbM88Ve9ETBCmhp88MXQ/edit?pli=1&tab=t.0

Mehta, M. (2020, December 16). *Public Key vs Private Key: How Do They Work?* Opgeroepen op Mei 2025, van Infosec Insights: https://sectigostore.com/blog/public-key-vs-private-key-how-do-they-work/

Microsoft. (z.d.). *Browse and add workflows in Microsoft Teams*. Opgeroepen op Mei 2025, van Microsoft: https://support.microsoft.com/en-us/office/browse-and-add-workflows-in-microsoft-teams-4998095c-8b72-4b0e-984c-f2ad39e6ba9a

MISP. (z.d.). *MISP Documentation and Support*. Opgeroepen op Mei 2025, van MISP: https://www.misp-project.org/documentation/

Shuffle. (z.d.). *Documentatie Shuffle*. Opgeroepen op Mei 2025, van Shuffle: https://shuffler.io/docs/about

StrangeBee. (z.d.). *Documentatie Cortex*. Opgeroepen op Mei 2025, van StrangeBee: https://docs.strangebee.com/cortex/

StrangeBee. (z.d.). *TheHive Documentation*. Opgeroepen op Mei 2025, van StrangeBee: https://docs.strangebee.com/thehive/overview/

Van Landeghem, G., & Kenens, L. (2019, Oktober 25). *Threat Intelligence*. Opgeroepen op Mei 2025, van https://docs.google.com/document/d/1XPqotoreX9tJQJcJsEOGVfpB8O3w5zBt71-oxrYhUtA/edit?tab=t.0

VirusTotal. (z.d.). *VirusTotal Documentation Hub*. Opgeroepen op Mei 2025, van VTDoc: https://docs.virustotal.com/

Wazuh. (z.d.). *Documentatie Wazuh*. Opgeroepen op Mei 2025, van Wazuh: https://documentation.wazuh.com/current/getting-started/index.html

xalgord. (2025, Januari 23). *Shuffle Installation*. Opgeroepen op Mei 2025, van Github: https://github.com/shuffle/shuffle/blob/main/.github/install-guide.md

Bijlagen

De bijlagen bevinden zich in het document ‘Bijlagen.pdf’ op mijn portfolio.